



Department of Informatics
Technical University of Munich



TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

BACHELOR'S THESIS IN INFORMATION SYSTEMS

Exchange Platform for eLearning Courses

Valentin Aurel Antoni

TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Bachelor's Thesis in Information Systems

Exchange Platform for eLearning Courses

Austauschplattform für eLearning Kurse

Author:	Valentin Aurel Antoni
Supervisor:	Prof. Dr.-Ing. Georg Carle
Advisor:	Dr. Marc-Oliver Pahl
Date:	November 12, 2018

I confirm that this Bachelor's Thesis is my own work and I have documented all sources and material used.

Garching, November 12, 2018

Location, Date

Signature

ABSTRACT

eLearning became an important part of media didactics over the last few years. Nowadays eLearning is a crucial learning method in most educational systems. It's no surprise that eLearning plays a significant role in universities as extension to the classical auditorium as it provides a lot of benefits like asynchronous communication and learning at your own pace. Almost every university and school provides eLearning courses for their students. These courses are often developed by the institutions themselves and do not get exchanged among different institutes. This is largely because there are almost no appropriate ways to do so. This often leads to poor content quality and no possibility of relevant student feedback. To address this problem, this paper develops an exchange platform for eLearning courses. The exchange platform is designed as a learning content management system to support the exchange of learning content between universities. Any content uploaded to the system gets versioned, a functionality no similar platform fulfills. Additionally, student feedback is collected to ensure course quality. The outcome of this thesis is a scalable exchange platform supporting the current workflows in eLearning.

ZUSAMMENFASSUNG

eLearning hat sich in den letzten Jahren zu einem wichtigen Bestandteil der Mediendidaktik entwickelt. Heutzutage ist eLearning eine wichtige Lernmethode in den meisten Bildungssystemen. Es ist nicht verwunderlich, dass eLearning an Universitäten als Erweiterung der klassischen Vorlesung eine bedeutende Rolle spielt, da es viele Vorteile wie asynchrone Kommunikation und Lernen in eigenem Tempo bietet. Fast jede Universität und Schule bietet eLearning-Kurse für ihre Schüler an. Diese Kurse werden oft von den Institutionen selbst entwickelt und werden nicht zwischen verschiedenen Instituten ausgetauscht. Dies liegt vor allem daran, dass es fast keine geeigneten Möglichkeiten gibt, dies zu tun. Folglich ergibt sich oft schlechte Qualität der Inhalte und wenig Möglichkeiten für Feedback der Studierenden. Um dieses Problem zu lösen, entwickelt dieses Papier eine Austauschplattform für eLearning-Kurse. Die Austauschplattform ist als Learning Content Management System konzipiert, um den Austausch von Lerninhalten zwischen Hochschulen zu unterstützen. Alle Inhalte, die in das System hochgeladen werden, werden versioniert. Diese Funktionalität wird von keiner ähnlichen Plattform erfüllt. Zusätzlich wird das Feedback der Studierenden gesammelt, um die Qualität der Kurse zu gewährleisten. Das Ergebnis dieser Arbeit ist eine skalierbare Austauschplattform, die die aktuellen Arbeitsabläufe im eLearning unterstützt.

CONTENTS

1	Introduction	1
1.1	Goals of the Thesis	1
1.2	Methodology	2
1.3	Outline	3
2	Analysis	5
2.1	Terminological Classification	5
2.2	Learning Content Management Systems	6
2.3	iLab Courses	6
2.3.1	Student Workflow	7
2.3.2	Technical Workflow	8
2.3.3	Roles	8
2.4	Existing Prototype	9
2.4.1	Requirements	9
2.4.2	Implementation	12
2.4.3	Prototype Improvements	14
2.5	Requirements for the Exchange Platform	17
3	Related Work	21
3.1	Moodle	21
3.2	Blackboard	23
3.3	AContent	25
3.4	Sakai	26
3.5	Comparison	26
4	Design	29
4.1	Redevelopment of the Prototype	29
4.1.1	Backend	31
4.1.2	Frontend	31

4.1.3	Prototype Requirements	32
4.2	Import and Export of Labs	33
4.2.1	Authentication	33
4.2.2	Scripts	34
4.3	Student Feedback Survey	38
4.3.1	Displaying Gathered Feedback	38
4.3.2	Integration in the Labsystem	38
4.4	Enhanced Browsing Experience	39
4.5	Analyzing and Presenting Metadata	40
4.6	Branching of a Lab	41
5	Implementation	43
5.1	Setup	43
5.2	Comparing Prototype and New Implementation	46
6	Evaluation	53
6.1	Old Requirements	53
6.1.1	Comparing to Related Work	53
6.1.2	User Evaluation Survey	55
6.2	New Requirements	56
6.2.1	Comparing to Related Work	57
6.2.2	Workflow Walkthroughs	60
7	Conclusion	65
8	Future Work	67
A	Source Code	69
B	User Walkthroughs	71
	Bibliography	81

LIST OF FIGURES

1.1	Separate File Servers (left) vs Exchange Platform (right)	2
2.1	Six dimensions of eLearning [34]	15
2.2	Research Questions	19
3.1	Authors Personal Moodle Page for courses in 2015/2016	23
3.2	Blackboard LMS Landing Page	24
4.1	Export Script	35
4.2	Import Script	36
4.3	Update Script	37
5.1	System Components and their interaction	44
5.2	Sequence diagram of setting up lab for survey [30]	45
5.3	Catalog of the prototype	46
5.4	Details of a lab	47
5.5	Student Feedback of a lab	48
5.6	Landing Page in the new Implementation	49
5.7	Catalog showing all labs	50
5.8	Lab details	51
6.1	Workflow Comparison of exchanging a lab	57

LIST OF TABLES

3.1	Summary of related work in eLearning	28
4.1	Pros and Cons of Continuing work on the Prototype	30
4.2	Dataflow and needed Input for Scripts	37
6.1	Julius Polars Feedback Survey Questions on Student Feedback	55

CHAPTER 1

INTRODUCTION

In addition to the classical auditorium, eLearning is often used as a teaching method. Independency of location, the acquisition of learning content at your own speed and the possibility for asynchronous communication are just some of the advantages [11]. To implement this type of learning an eLearning platform is necessary. For this purpose, most institutions use their own systems and even use different systems for different courses.

For example at Technical University Munich a course is held called iLab, which originated at the University of Tübingen. The course uses the so called Labsystem as their main eLearning platform. Even though this platform provides an Import and Export function there is no easy way of exchanging courses. If one were to exchange content there are several steps to be done. Export the course, zip it, send it via email, unzip it, import it in the other labsystem. This can result in even more effort if the server is only reachable via ssh. Consequently we ask ourselves: *how can we improve the student and instructor experience in eLearning, by supporting the exchange of content?*

1.1 GOALS OF THE THESIS

The goal of this thesis is to develop a Learning Content Management System for the iLab courses. In other words an exchange platform for the eLearning platform of iLab, further described in section 2.3. The system is to exceed the prototype built by Julius Polar in his master thesis *iLab Exchange Platform* from 2017. In the future this platform will be used to improve the workflow of the iLab courses for instructors. The other main objective is to support exchange between universities and therefore promote interchange

of ideas. This platform will replace the current file server setup of each university for the respective courses with one collaborative platform as seen in Figure 1.1.

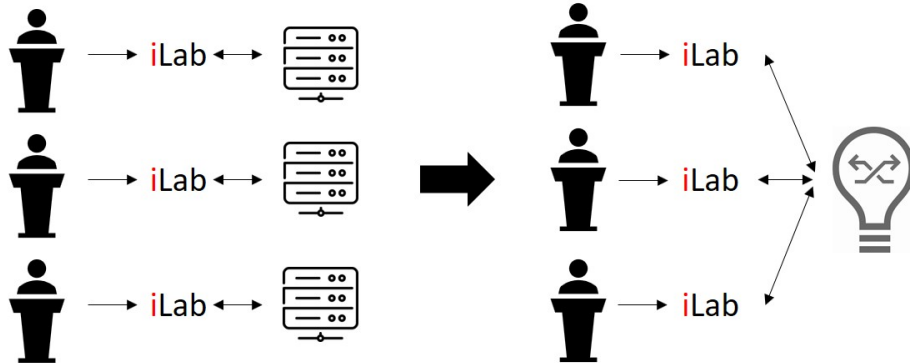


FIGURE 1.1: Separate File Servers (left) vs Exchange Platform (right)

1.2 METHODOLOGY

The development process is done in several steps in this thesis. First we search for a definition for the exchange platform. Taking this definition into account, we present the iLab roles and workflows. Afterwards the existing prototype is critically discussed and improvements found. Outcome of the analysis are the requirements for the exchange platform.

Following the analysis, related work in this topic is reviewed. It is checked whether an existing system can be used, and if not whether it is possible to adopt system features from similar implementations.

The requirements found in the analysis and the ideas from related work lead to a design of the system. The design is also built using the ideas of the existing prototype implementation.

After a suitable design is found, the requirements are implemented. Since parts of the prototype are being newly implemented, see chapter 4, a new implementation of these parts based on the old requirements is performed first.

At last the implementation is evaluated against the requirements found in the analysis. If requirements have not yet been implemented, an explanation shall be given accordingly.

1.3 OUTLINE

Necessary background for this thesis is presented in chapter 2. Here we find an appropriate definition for the platform. Afterwards we define some important terms that might otherwise lead to confusion. Thereafter the iLab course and the current prototype are presented. The chapter ends with a presentation of the requirements necessary for the exchange platform.

In chapter 3 we take a look at popular related works for platforms in eLearning. We review Moodle, Blackboard, AContent and Sakai and give a brief overview in section 3.5 at the end of the chapter.

In the next chapter, chapter 4, we describe how we ended up with our implementation and why we chose to do so. First we discuss a possible redevelopment of the Prototype and how to implement the requirements found by Julius Polar. Thereafter we investigate the design of each requirement found before.

In chapter 5 we present the components and their interaction with each other and we compare the old and new implementation.

With the conclusion in chapter 7 we summarize the thesis and the achieved results.

After the conclusion, chapter 8 is about anything left to implement in the future.

CHAPTER 2

ANALYSIS

This first chapter describes the current state of the exchange platform and all other relevant systems. This way relevant requirements for the later implementation are found. First, we find a definition for a system that improves the current workflows. As the exchange platform shall be used for the iLab course, the iLab and its workflows are presented. Next the existing prototype is discussed to show the current state of development. After its flaws are analyzed room for improvement is found. Finally, the resulting requirements are presented. The requirements are the outcome of this chapter. Requirements found here are emphasised, after the according paragraph, like this: <R00 **Example Requirement**> for providing an exemplary requirement

2.1 TERMINOLOGICAL CLASSIFICATION

This section provides some definitions for frequently used terms.

A **course** is a composition of different modules on a superordinate topic. It typically lasts one semester and is graded. One example is the iLab course in which you learn the basics of the Internet.

A **module** or **lab** is a part of a course. It revolves around a more specific topic, like the Tor-Network. In the case of the iLab course, a period of 2 weeks is expected for each lab.

The **iLab** is an overarching term for eLearning courses discussed in section 2.3. iLab at TUM is including the courses iLab and the more sophisticated iLab², as well as the new iLab X.

The **Labsystem** is the eLearning platform used for the ilab. It is available as open

source on GitHub at <https://github.com/m-o-p/labsystem> [27]. It is further described in section 2.3.

The **prototype** is the prototype of the exchange platform on which the implementation of this work is mainly based. It is analysed in section 2.4.

2.2 LEARNING CONTENT MANAGEMENT SYSTEMS

First, we search for an appropriate definition for a system that improves current workflows in eLearning content exchange. This definition is especially relevant for chapter 3 to find existing software fulfilling similar purpose and to define the research questions in section 2.5.

The Labsystem is the eLearning platform used in the iLab courses. Clark et al. describe eLearning as "instruction delivered on a digital device that is intended to support learning" [1]. This definition describes an eLearning platform, but we aim to develop a system to support content exchange between many systems. Consequently it is not sufficient for the exchange platform developed in this thesis. The definition is just about the learning part, while the developed software is much more about a system to monitor and share content.

A more fitting definition would be of Learning Management Systems (LMS), which Gautreau et al. define as "a selfcontained webpage with embedded instructional tools that permit faculty to organize academic content and engage students in their learning" [12]. This definition still can be expanded to emphasize more on the content exchange. The therefore even more comprehensive definition is about Learning Content Management Systems (LCMS). Ismail et al. extend the definition of LMS to LCMS with "the primary role of an LCMS is to provide a collaborative authoring environment for creating and maintaining learning content. Within the LCMS, workflow processes can be activated to coordinate collaborative authoring of learning content" [19]. The extension of this definition to the previous ones is mainly collaborative authoring and a better description of content management. The tool developed in this thesis shall be used as a LCMS.

2.3 ILAB COURSES

The labsystem is a web-based eLearning system developed in 2003 by Dr. Marc-Oliver Pahl. The concept is in steady development and got awarded with the Ernst Otto Fischer-Lehrpreis of Technische Universität München in 2013 for its innovative learning

approach. Further information can be found here: <http://labsystem.m-o-p.de>. In this section the iLab workflows and roles are presented for readers not familiar with the concept and to help identify requirements.

2.3.1 STUDENT WORKFLOW

First we take a look at the learning side of the iLab workflow, then we analyze the technical side, which is almost only used by instructors. Each topic covered in courses using the Labsystem has the same structure. For example in the Onion Routing lab, the lecture will give the students a basic idea of the topic. This knowledge then gets deepened in the prelab. Students apply this knowledge in the lab, where they build different networks and answer questions after experimenting with their setup. For example building their own Tor Network and search for anonymity advantages and flaws of the system.

The following description of the workflow with the Labsystem is based upon the iLab but could also be used for other courses with similar workflow. The approach is especially well fit for students normally having trouble with preparing and doing follow ups to lectures, as the workflow includes repetitions [36]. Courses using the Labsystem divide every topic in four parts. The first part is a lecture where attendance is mandatory. If one was not able to attend or did not understand a certain aspect of the topic the slides as well as a recording of the lecture gets uploaded into the labsystem. After the lecture the students have two weeks time for the following tasks.

A prelab which is done by each participant individually. It starts with reading information given in the Labsystem and answering specific questions to each subtopic. The questions are often more specific than the given texts, so the students have to read further documentations, for example RFCs.

The main lab is done in teams of two. To make the lab available in the Labsystem both members have to answer all prelab questions first. The lab is done in a room provided by the instructors containing all the necessary equipment for each lab. Necessary information as well as a setup can be found in the Labsystem. After following given steps they have to describe the outcomes to every experiment. Hereafter the instructors grade the answers.

By the time half of the semester is over each participant has to pass an oral attestation covering all previous topics. At the end of the semester another oral exam takes place with the topics covered since.

2.3.2 TECHNICAL WORKFLOW

The following workflows do not only apply to iLab, but to many eLearning systems. At the beginning of each semester an instructor in charge of an eLearning platform must assemble planned courses for the semester. Often these courses are either completely new, or imported from a fileserver. During the semester, courses often get updated and rectified. At the end of the semester the updated courses are either discarded, or exported back to the fileserver where they will be imported back at the start of a new semester. If they are to be reused, but are not exported, the enhancements made to the old version are lost. Additionally any comments and feedback from students are lost either way.

The main issue why the export does not always take place is that it can take a lot of effort. The Labsystem, the eLearning platform of iLab, for example implements an import and export function. This function creates a folder, which is copied to a fileserver via ssh. The exported content often differs from older content, but there is no versioning to show differences. If one were to ask for a lab to use on their own system it would have to be exported, copied via ssh, packed into a zip file and send via email. The receiver would then have to unpack the content from zip file, copy it to the server running the platform and finally import it to the other system.

The workflows above show some major issues. There is no versioning given and the exchange of content is effortful. Additionally if there is any feedback gathered, it is not saved with the learning content, resulting in the loss of feedback. A similar analysis resulted in the requirements for the prototype described in section 2.4.

To summarize, the depicted workflow above is already in place and working well, but lacking the possibility of versioning and easy handling. All data is copied in and out at the beginning of each semester manually. This brings the main requirement, namely easy import and export. The labfiles should therefore be transferred between labssystem and exchange platform with just a few clicks. Thus the platform needs to support all items above, e.g.: text, pictures, multiple choice questions etc.

2.3.3 ROLES

The workflow described above leads to a certain set of roles. Since it is a course at a university there has to be a student as well as an instructor.

The *student* is only allowed to look at the labs and answer the given questions. They are neither allowed to see the other teams solutions nor the sample solutions. The other main role is the *instructor*. As there is more than one instructor this can be divided in two subclasses. One being the author to import/export the course and create content.

The other role is as a corrector. They can view all groups work and correct it in the Labsystem, but they don't need access to the exchange platform.

Hence we need role-based access to strictly constrain the access to the solutions to the lab as they have similar importance as exam solutions. Students may be allowed to view the labs but solutions have to be excluded from this view.

2.4 EXISTING PROTOTYPE

Julius Polar did a similar analysis 2017 in his master thesis *iLab Exchange Platform* [30]. As his work is mentioned often throughout this thesis it is only cited here, to keep the reading structure cleaner. All mentions of the prototype, or Julius Polar belong to this citation. [30] Based on his analysis he implemented a prototype around the requirements he found. General information can be found here. In this section the requirements found by Julius Polar are presented and discussed. It is also explained how he identified these requirements. Subsequent the implementation is introduced and scrutinised.

2.4.1 REQUIREMENTS

Uploading and Sharing Material When discussing eLearning one of the most important features is the learning material. Without content made with pedagogical principles students as well as faculty members will be dissatisfied with the eLearning process [14]. Most Massive Open Online Courses (MOOC) provide course information in different media. Also most LMS like Moodle are able to present all sort of material, as discussed in chapter 3. This material can be exported in different file formats, in Moodles case XML. The iLab does also not only use text, but also pictures, graphs, external sources and videos, as described in section 2.3. This material must be stored in the exchange platform, which therefore needs to support text and images as well as videos. The prototype implements support for all types of files with the use of Git, more precisely GitLab, as a backend for learning material. As long as the learning material is exported in one file, or folder, git is able to store them.

Role Based Access Control In terms of workflow Julius Polar made an analysis on who should share contents. As a part of iLab² students build their own labs. These are afterwards shared, solved by other students and rated. Thus students will need to have access to the exchange platform. However, since students are not allowed to access the solutions to the labs, this access has to be restricted to

the instructors. Possible access to solutions would be contraproductive to learning and would falsify the grading. The requirement is implemented by using GitLabs RBAC via the GitLab API in the prototype. A problem with the implementation is a possibly unsafe login, which is discussed in the implementation subsection below, subsection 2.4.2.

Material Versioning Today all data is manually copied in/out of the Labsystem to/from a fileserver every semester. As every university hosts their own courses and server there is no exchange given. Since several people can work on one lab in the future the new tool should be able to version all files. Thereby it is possible to revert unwanted changes. One of the main reasons for wanting to change content is the motivation to continuously improve the iLabs. The iLab heavily relies on student feedback to improve over the years. To keep track of changes on the learning material either a manual log could be created or a versioning system could be used. Julius Polar came to the same conclusion and decided to use a versioning system, decreasing the workload of instructors. This was the main reason for using git. With GitLab as a backend for the material this requirement is fully implemented.

List available Material We aim to improve eLearning workflows by providing a platform to exchange courses and their material on. These courses can be downloaded and reused by any instructor. A listing of available courses publicly also seems a good idea for gaining other instructors interest. The reason to get other instructors interested is that the iLab workflow is heavily feedback based. The more students use the iLab courses, the more feedback is gathered and transposed and thus the content improves. Thus the exchange platform shall show general information about the labs publicly. The details of the individual modules should only be visible for registered users, otherwise copyright problems could occur if someone simply copies the labs. The solutions should only be accessible if the role of the user allows it. In the prototype this is partly fulfilled, as there is a list of all labs, but it basically only shows the name and description, but not any additional information.

Search shared Material As described before instructors from different universities can assemble their courses from modules uploaded to the exchange platform. To find specific material most content platforms provide a search bar, or recommendations. Obviously, some sort of search feature has to be implemented in our platform as well. Since the platform discussed in this thesis will specialised in one topic, only a keyword search and tag filter is implemented in the prototype, which is completely sufficient for our scope.

Gather and Analyze Feedback about the material As already mentioned the iLab heavily depends on student feedback. So far this feedback was only gathered by comment boxes at the end of each lab. This is to be changed with the introduction of the exchange platform. Julius Polar described it as "One of the main focus[es]" to gather and analyze student feedback. He discusses different approaches: surveys, email questionnaires, comments, ratings and social media. Surveys surpass the other approaches in different ways. A web based survey surpasses a web questionnaire and reaches about the same level of response as a hardcopied survey [20]. Even though the data used by Polar from [20], being from 2004, is a bit old it is not outdated. Written reviews, as well as ratings, can be included in a survey, which is what he chose to do. The last idea, to use social media for feedback gathering, was quickly discarded, as it is too unreliable in terms of authentication and privacy policy. Other countries, like Vietnam, use Facebook a lot to exchange material and get comments on them as an exchange student of the iLab told. The prototype extends the workflow so far with a link to a survey after each lab. This would fulfill the requirement completely, but still has to be integrated in the Labsystem. So far the links are generated on the exchange platform, but have to be manually handed out to students.

Display processed Feedback As written above feedback is gathered in the exchange platform to help instructors improve their learning material. Thus the gathered feedback should be displayed for each lab individually. The requirement is partly implemented in the prototype. To many charts are shown on a separate page. Some should be moved to the lab detail page to provide an easy overview. Further the overall rating, as well as some additional information should be seen in the catalogue. In the next subsection, subsection 2.4.2, the display of this feedback is further discussed.

Analyze and visualize metadata from the Labsystem Besides feedback, metadata is a common way to gather information about eLearning. As things stand today, a fellow student, Sarah Jost, is writing a bachelor thesis about an improved algorithm for gathering metadata from the Labsystem. In particular the metadata gathered is about time tracking of students answers. As there was neither data, nor much information given about the metadata the prototype only mocks this. This requirement is thus still up for us to implement and taken into account as a new requirement in section 2.5.

2.4.2 IMPLEMENTATION

To give a better overview of the prototype, the current development status is shown. Julius Polar partitioned it into four segments. The first being GitLab as a backend, secondly the frontend based on AngularJS framework, then the Database using MySQL and the integration into the current Labsystem. Additionally we take a closer look at the survey and the representation of the collected data.

GitLab has been specially selected for the **backend** because it is suitable for storing different dataformats. Versioning normally is not available in most LCMSs, but is also made possible with the use of git as backend. Even if most of the feedback is stored in the database described later, it is backedup here as well. Each department can administer its courses in the frontend with an account given the appropriate rights in GitLabs. This integrated role-based access control is another positive aspect of using GitLab as a backend.

To grant the frontend portal access to the data in GitLab it has its own account. The labs are stored in a group, with the solutions as different projects in another group. The GitLab account is only in the labs group with access level "Guest". Thus the token used to get the data from the GitLab can not be used to get the solutions of the lab. Access restriction to guest is sufficient since it does not need to change any repositories. Each instructor should at least get developer status to create and administer labs. It would be easier to even give more rights to trusted instructors, as merge requests do not become an issue anymore.

As of now the prototype uses only the main branch and thus does not allow other instructors to change a lab for their own use. The only possibility to do this is by cloning a lab and uploading with a different name.

The **Frontend** is written in HTML, CSS and Javascript using the AnuglarJS Framework. In this case it is running on an Apache2 webserver. Communication between backend and frontend is done by using the GitLab API. A frontend token is used for authentication to the GitLab.

To protect the data inside the labs user authentication is necessary. Users have to login the frontend using their GitLab account. This way the users are distinguishable and projects can only be accessed by authorized users, resulting in protection of the labs and solutions. Authentication is implemented in sending a post request containing username and password to the gitlab. Even though it is provided in the API, GitLab recommends using one of the other given options [6].

As of now the main page shows a catalog of all existing labs the user has access to. It includes a search bar as well as a tag filter. The search bar searches labs describtion

and name while the tag filter only filters the labs for assigned tags. This page also contains an activity feed showing latest changes to the projects the user has access to. These recent activities are updates made to the git files. This can be rather confusing to non-technical instructors.

In the middle of the main page is a list with the title and a short description of each accessible lab. By clicking on it the page for each lab provides the same information as well as preview of how it looks in the Labssystem. Said review is done by creating an iframe showing the generated HTML page. Underneath this preview are the links to clone this repository to gain access. On the lower part of the page is a link to survey results as well as a comment section.

The implementation of the survey and the representation of collected data is laid out below.

The **Database** is implemented using MySQL. It is used to store the data gathered by the survey. Instructors have the option to generate tokens for the survey in the frontend. These tokens serve as anonymized authentication for students. They provide that each student can access the survey only once.

The database stores these tokens. As MySQL is a independent database the survey data is not included in the versioning of GitLab. Therefore Polar decided to back it up using GitLab.

Labsystem Integration is needed to administer the labs in the exchange platform. The labs are created and updated in the Labssystem and then uploaded to the exchange platform. Conversely, it is also possible to import labs. The integration is also needed for feedback gathering.

To provide students the easiest way to give feedback they each have an anonymized survey link. Each survey link includes a token to circumvent multiple answers by one student as well as anonymize their answers to the instructors.

The main part of the Labssystem integration is uploading and updating the lab content to the exchange platform. As of now this is done with shell scripts manually via a console. Here we distinguish between three scripts. The *export script* is used to create a new git repository in the GitLab. For this purpose it requires a name for the exchange platform as well as the path to the lab folder. Via the GitLab API it checks whether the name is already in use and creates a new git repository if the name is not in use yet. Afterwards it changes the owner information in GitLab to the one deposited in a configuration file. Subsequently it creates an extra project for the solutions and pushes data to both projects. The *update script* is the equivalent of a git push. It thus is a rather simple script. It expects the path to the labs and a commit message and performs git status, add, commit and push. The *import script* is used for the opposite as the

two others. It is used to load data from GitLab into the lab system. For this it expects a folder path as well as a the lab name as input parameters. The script is also pretty straightforward. It changes the working directory to a new directory with the first parameter as path and clones the data from GitLab, given with the second parameter, there.

None of the scripts implement any exception handling yet. If there is a git merge issue it would need some git proficiency to fix it. Consequently this, plus the fact everything has to be done in the console, will result in issues for non computer scientists. This already leads to the first requirement.

<**R01 One Click Import/Export**> function is to be integrated in the eLearning system for broader usage and higher usability.

The **Survey** is used to gather feedback from students. It consists of 5 questions. On a one to five scale it asks students about the difficulty of the labs, how interesting the labs were to them and how lengthy the lab felt. Additionally how much time they spent on the pre-lab and lab. At the end is a one to five rating and a comment box. Beside the comment every question is mandatory.

To provide the integrity of the data each student is only allowed to answer the survey once. Julius Polar looked into two ways of implementation. Either require authentication before answering the survey, or generating a token for each survey entry. As account creation or some sort of entering a password is additional workload for a student he discarded this idea. This probably favours the amount of feedback gathered as the survey is voluntary. Additionally a token provides anonymity to the instructor, as he can't match tokens with students. Thereby privacy is granted.

Data gathered from the survey can be accessed on the details page for every lab. The feedback site shows plain bar charts with average, median, max, min and standard deviation next to them. This presentation of data could provide a lot of information but is not appealing. As the graphs are plain and one below another, instructors do not gain much from this view. Also this section is available to anybody. Even though this provides transparency, the comments could contain information about possible solutions and should therefore be viewed only by staff.

<**R02 Student Feedback**> is to be collected and displayed by the exchange platform.

2.4.3 PROTOTYPE IMPROVEMENTS

The implementation of the prototype leaves quite some space for improvements. The main idea and workflows of iLab could also be applied to a completely different

field of study, e.g. chemistry, the goal of this thesis is not only to provide an exchange platform for the iLab, but to answer the following general research questions.

The superordinate question we have to answer is *how can we improve the student and instructor experience in eLearning, by supporting the exchange of content?* To answer this question we divide eLearning into the following six dimensions. [34]

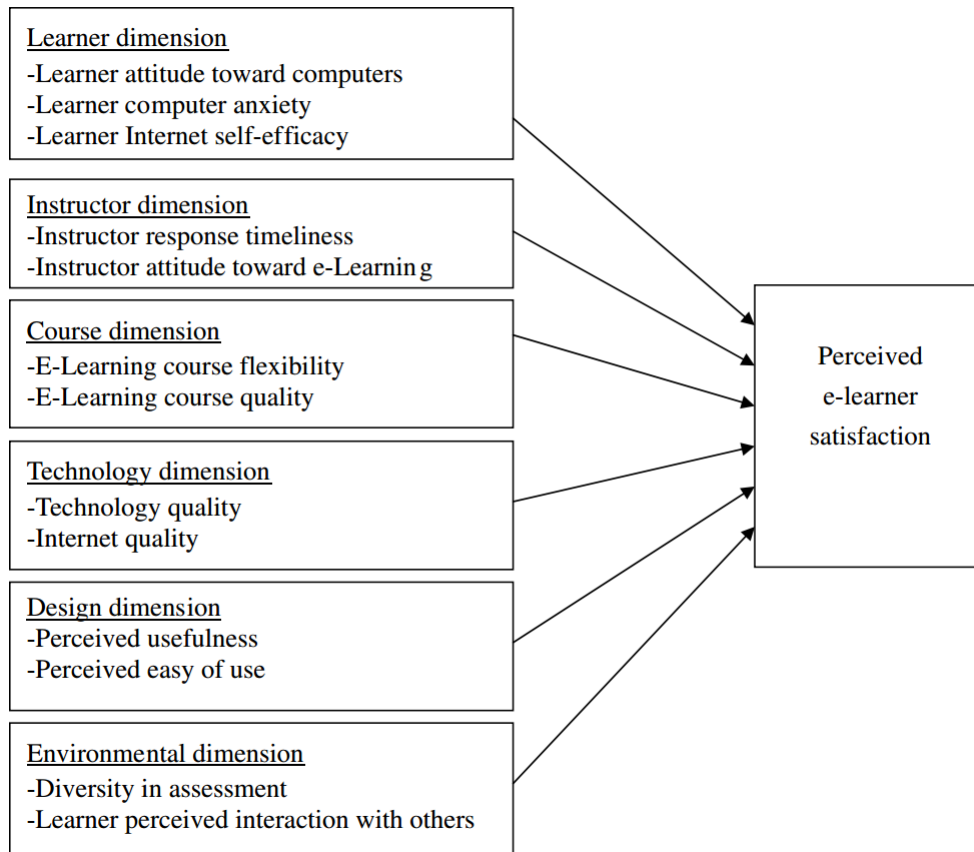


FIGURE 2.1: Six dimensions of eLearning [34]

Learner dimension is not a subject for change to us. Only students can change their attitude towards eLearning. The Environmental and Technology dimensions are mostly relevant for the instructors of iLab, but not for the scope of this thesis. However the other dimensions are relevant for us and help us in finding more detailed research questions.

Instructor Dimension: The way this thesis wants to improve eLearning workflows is by giving the possibility of exchanging content. Except in iLab², this exchange is only carried out by instructors. The expected outcome is improvement in course quality.

To achieve this, instructors have to be able to choose courses according to some data provided by the system. Therefore the question resulting from the above is *What data is relevant to teachers and should be taken into account when selecting courses and how can we provide it?* This can be partitioned in two topics. First the aspect of metadata from the Labsystem. What metadata can be extracted from the Labsystem and how is it displayable to instructors. This is already a requirement found in section 2.4. Second the aspect of feedback from the student survey. The survey implemented in the prototype is a good way to gather feedback. As Julius Polars evaluation showed, the display of this feedback is unsatisfactory.

<**R02.1 Display Student Feedback**> has to be reworked to give instructors a better overview.

Course Dimension: To provide the highest course quality for students, quality assurance is made with student feedback. It is vital for a good eLearning course to take feedback from students [21]. To get this process of quality assurance this question has to be answered: *How can we gather as much student feedback as possible, while keeping privacy and data integrity?*

This question was already answered in Julius Polars thesis. As mentioned above the display of student feedback has to be reworked in order to give instructors a better overview. Additionally an import aspect of course dimension is the length of a course. The lab system measures this roughly, but as of now there is no way to display it. <**R03 Including Metadata from the Labsystem**> to gain an even better overview of course.

Design Dimension: To achieve the above to the best of our abilities we need to answer *how can we achieve high usability for instructors and students, while keeping data secure and privacy granted?* One big part of this is the import and export function. Already mentioned by Julius Polar in his future work, it has to be included in the Labsystem. This also coincides with the technical view of eLearning from Berge, as he sees one of the main tasks of a facilitator is to get all users comfortable with the system [3]. This does apply to instructors in this case, but is mostly student-centered.

There are two student interactions with the system. One is making a lab for other students and trying the other labs, the other is the survey at the end of each lab. For both tasks a better integration is necessary, to keep the focus on learning. For his the import and export function, as well as the link to the survey, have to be integrated in the Labsystem, to achieve more transparency in the system [3].

2.5 REQUIREMENTS FOR THE EXCHANGE PLATFORM

<**R02.2 Survey Integration**> in the Labsystem to provide more transparency and easier access.

Apart from those dimensions the system should not only be usable for iLab, but also for courses of different topics and universities. Therefore it shall look less technical, but more appealing. To make it more interesting for other universities a Landing Page must be added, because going directly to the catalog is not informative for instructors who might want to take part.

Additionally it was requested in Polars evaluation to have more information directly in the catalog. For this the catalog has to be reworked.

This results in one requirement, which is split up in three, as seen in section 2.5.

<**R04 Enhanced Browsing Experience**> is the superordinate requirement for several smaller subtasks

The last requirement is already mentioned in Polars 'Future Work'. It is the branching of labs. Out of the box, git "allows the content of the repository to diverge in many directions" [25] with branches. This means a project can have a branch split up, for example by another university, which differs from the original one, while maintaining the same base. Later these branches can be merged back together. This would allow instructors to change labs without interfering with each other. Currently this is only possible by importing a lab, changing it, and uploading it with a different name. This workflow could be improved with the use of git branches. <**R05 Branching of labs**> shall be possible to skip unnecessary steps with exports and imports.

2.5 REQUIREMENTS FOR THE EXCHANGE PLATFORM

In the previous sections a definition was found for our exchange platform for being a LCMS. Additionally the labsystem workflows were described as well as the existing prototype. This lead us to the research questions mentioned at the end of 2.4. With all these components in mind the requirements for the exchange platform were derived.

The Research Questions are displayed in Figure 2.2. The question at the top is the main research question of this thesis. Below are the questions that lead to the requirements, dividing the main question in three categories: Information for Instructors, Usability and Feedback gathering.

The Requirements are also listed below, in the order they were found. They are also described shortly and, if necessary, subdivided into smaller requirements. Each requirement is followed by the question from which the requirement originates.

R01 One Click Import/Export At the present moment the prototype implements an export/import function for the Labsystem. It is a script run via the linux shell. This is to be changed to an easy to use UI solution.

Result of Q2, Usability

R02 Enhanced Browsing Experience The exchange platform is to receive a complete design overhaul. This requirement is therefore divided into the following subrequirements.

R02.1 Add Landing Page The exchange platform shall be extended with a landing page. This page is for giving an overview what the iLab is about and what possibilities it offers for instructors.

Result of Q2, Usability

R02.2 Catalog rework The catalog shall give more information about a lab. Further details shall only be available after logging in. The lab details page shall also include student feedback.

Result of Q1, Information for Instructors

R02.3 Overall Design Rework The exchange platform shall get a new layout, including themes, to be more pleasant to work with.

Result of Q2, Usability

R03 Including Metadata from the Labsystem To give teachers an extended overview over each course, metadata is collected in the Labsystem. This metadata should be accessible and easy to evaluate for teachers on the exchange platform.

Result of Q1, Information for Instructors and Q3, Student Feedback

R04 Student Feedback Survey In following Marc Oliver's approach, great importance will continue to be placed on student feedback. This requirement is divided into the following two subrequirements.

R04.1 Displaying Gathered Feedback The current feedback display provides instructors with little information and only after viewing it on a separate site. The feedback should therefore be prepared, and displayed, in such a way that instructors receive all important information at a glance on one page.

Result of Q1, Information for Instructors and Q2, Usability and Q3, Student Feedback

R04.2 Survey Integration In order to collect this feedback, the existing survey shall be better integrated into the Labsystem, while continuing to ensure privacy and data integrity.

Result of Q2, Usability and Q3, Student Feedback

R05 Branching of labs Branching labs shall make instructors able to not just copy another module, but change it. They shall be able to create their own branches of another instructors work.

Result of Q2, Usability

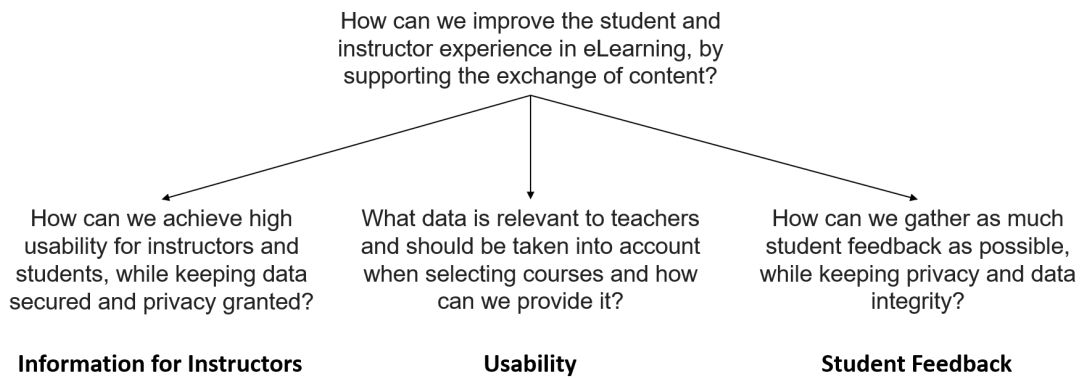


FIGURE 2.2: Research Questions

CHAPTER 3

RELATED WORK

This chapter evaluates systems geared towards content management in eLearning. As Julius Polar already did a similar analysis of a broadened spectrum of eLearning systems this is kept rather short. Also we are not expecting to find a system fitting exactly to our requirements, but are rather looking for ideas on how to implement our requirements. Well known in eLearning are also Massive Open Online Courses (MOOC). Even though they are very popular, they are not taken in consideration, as they only fulfill very few requirements and are already extensively described in Polars thesis [28].

For every related work there is first an overview of each system and a consideration on how well they fit the requirements depicted in section 2.5, as well as the requirements given by Julius Polar. After each section the source of the contained information is given. Afterwards is a written summary and a table listing key features.

3.1 MOODLE

The first system we have a look at is Moodle, an abbreviation for "Modular Object-Oriented Dynamic Learning Environment". This open source eLearning platform was developed in 2002 and is used by many universities, for example TUM. The TUM implementation can be seen in Figure 3.1, with the authors courses in 2015/2016. The design of the system is rather plain and functional. Due to its modular structure it can be tailored to fit special purposes. It was already covered in Julius Polars thesis, so this analysis will not go into detail.

Considering Moodle is an eLearning platform it allows instructors to share all kinds of content easily, including text, pictures and videos. As the name already suggests,

Moodle is open for many different modules, some of which are explained here. The lesson activity module for example provides testing options including multiple choice and free text. The questions are saved in a database, so they can be reused at some point. Even though this sounds fitting to an exchange platform the database only saves the questions, not the complete courses. Also the database is only accessible by an admin. There is a feature provided by Moodle which could be interesting for the purpose of an exchange platform. But this 'Course Sharing' is only for Moodle courses. As the Labsystem is too well developed and supports features we cannot replace with Moodle switching the iLabs to Moodle would not work. But one good idea that this feature has is the option to either make a course downloadable, or for people to enroll in. This way courses can be spreaded better and more easily. Although this is a good idea, it is unfortunately not convertible to the iLab, as the courses require hardware and are not only for teaching but also have to be corrected. Another noteworthy feature is the feedback tool included in the feedback activity module. Instructors can create questions for multiple choice as well as text answers. These can be used to model student surveys for gathering feedback about the courses.

The import and export of courses can be done with XML files. Moodle also supports the Common Cartridge Standard [13]. Despite the modular structure of Moodle and its prominence in Europe there seems to be no popular exchange platform for Moodle courses so far. This probably is due to the structure of moodle. It is designed to implement and show single courses to students, so each instructor has its own course. The labsystem on the other hand consists of many small modules for one course.

Even though Moodle implements most of the requirements needed for a LCMS it does not implement the most important ones, which are collaborative authoring and exchange of ideas. The exchange of courses between instructors is only possible via the database mentioned above and thus not usable enough as an exchange platform.

All information written above was obtained from moodle.org, as well as hands on experience in two different moodle set-ups.

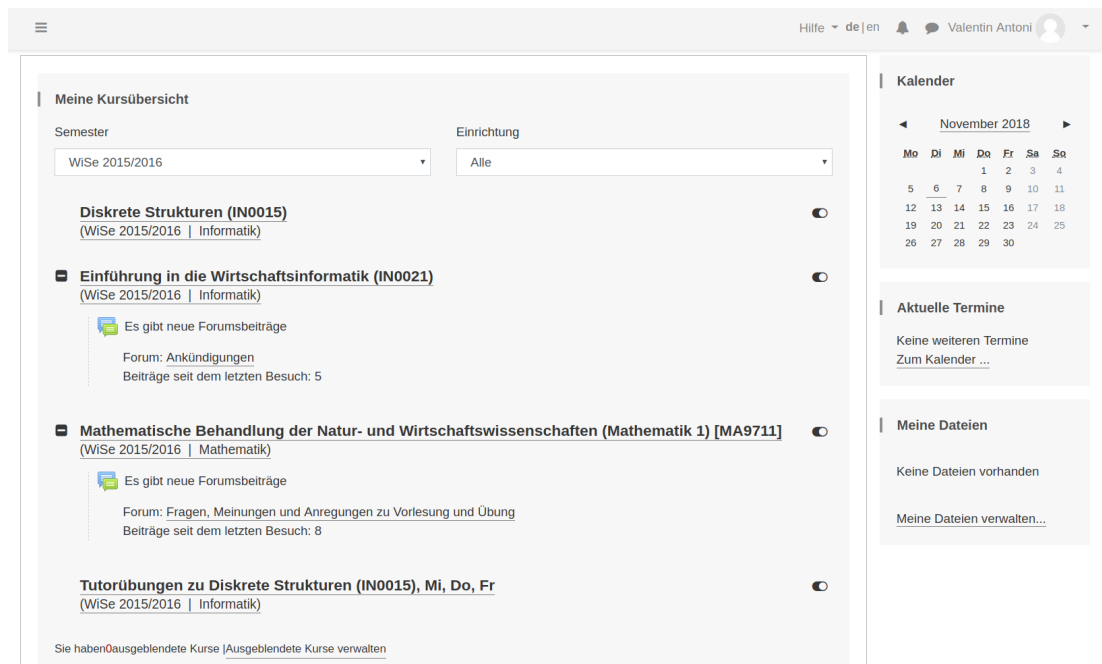


FIGURE 3.1: Authors Personal Moodle Page for courses in 2015/2016

3.2 BLACKBOARD

This section is divided into two subsections. The first part deals with Blackboard Learn, whereas the second part deals with Blackboard Open LMS.

Blackboard Learn is a very popular LMS used mostly in the USA. It does not only target universities, but also learning for children, government and businesses. In contrary to Moodle and many other tools it is not Open Source, but has software as a service (SaaS) as business model. This implicates running costs but also support for setting everything up etc. On the other hand it provides an appealing design, seen in Figure 3.2, which distinguishes it from most open source systems. This design looks appropriate to be oriented on it.

As all other LMSs Blackboard provides an eLearning platform. Relevant learning materials can be uploaded in many different file formats. The platform is, like Moodle, structured by courses and students can access courses, if enrolled. Students and instructors can communicate either via message, or discussion boards. Data exchange happens in different ways. Students can either download learning material from the webpage, or can link their Dropbox/One Drive account to automatically download it to any device. This way all data is automatically up to date. Even though this is an interesting idea it uses external resources not always available to all students.

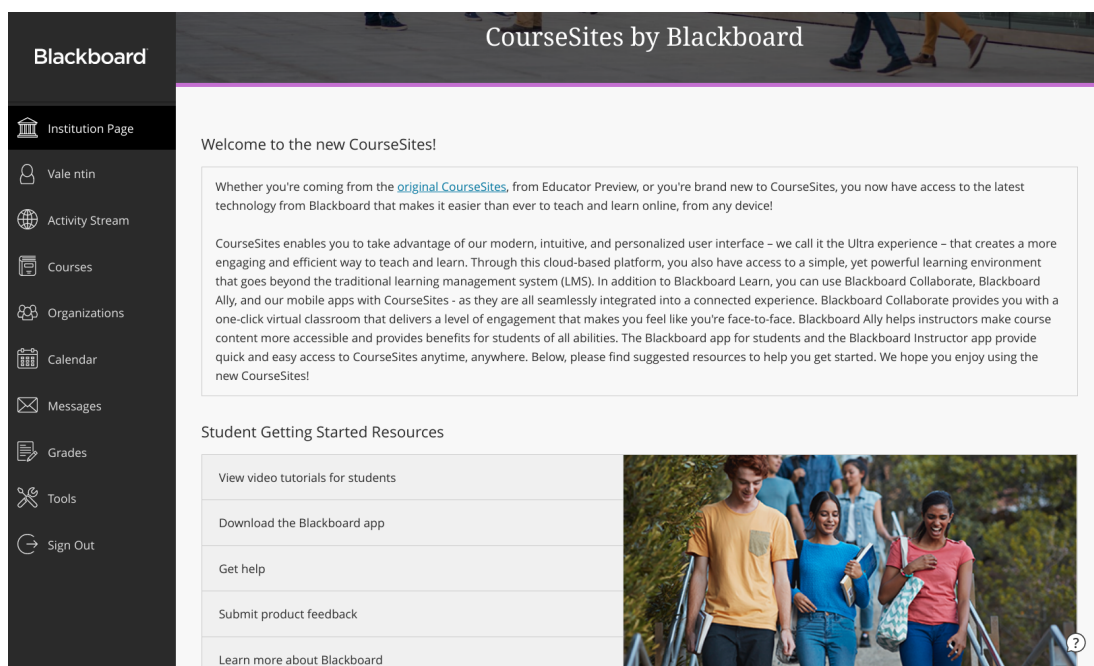


FIGURE 3.2: Blackboard LMS Landing Page

The other solution made by blackboard is Blackboard Open LMS. It uses Moodle LMS as a base platform and removes old features and improves existing ones. With adding a new theme it looks more appealing and promises easier handling. X-Ray Analytics provides insight in performance of all students. By logging logins, combined with previous grades and answers to homeworks and quizzes at-risk students can be identified. One key aspect of this is the metadata that is available to instructors this way. An estimated time of work put in the course, as well as grades. Even though this opens up new ways of supporting students it is also a great risk in terms of privacy.

Another noteworthy feature is Blackboard Open Content. It allows authoring, sharing and reusing of content and courses. This fulfills the definition made in section 2.2. By setting parameters and filters, content on this platform can be shared with various institutions or made available globally. This content can then be searched for by level of difficulty, subject and other criteria.

With X-Ray Analytics and Blackboard Open Content, Blackboard Open LMS provides answers to the main research question on how to improve eLearning for instructors, as well as what metadata should be analyzed and presented.

Blackboard is an LMS and provides a handy tool with the new Blackboard Open LMS. As the name of the latter suggests it is mostly a platform for eLearning, but is also the

best match for a fitting exchange platform. Even though it is the most fitting it does not support versioning and integration in the Labsystem workflows.

All information written above was obtained from <https://www.blackboard.com/learning-management-system/> and <https://www.blackboardopenlms.com/higher-education/> as well as hands on experience at <https://preview-learn.blackboard.com>

3.3 ACONTENT

AContent is an open source project. It is a php written LCMS made along with the LMS ATutor, but it can be used as a standalone tool for other eLearning systems as well. Like Moodle it is made with adaptability in mind. Thus it supports many standards, which can be viewed on their website.

AContent implements RBAC with a Regular User, Translator, Author and Administrator. A Regular User only has basic access like managing their profile and view open courses. A Translator can do this as well as translate a course into another language, from which any user can choose. The author can also do everything already mentioned, but can also create and edit courses. Administrators are of course able to do everything mentioned before, as well as managing users and changing the sites.

Being an LCMS AContent implements listing, uploading and searching materials with limited access to the given users. The software also provides a Web Service API for software development. Searching, previewing and importing/exporting on another platform is possible that way. The API further can be used for authentication with OAuth2, providing a single sign-on on exchange platform and learning platform. The stored data can be transferred to any platforms adapting the Common Cartridge Standard. SCORM, which stands for Shareable Content Object Reference Model, is the predecessor of Common Cartridge [13]. These standards were adopted by some learning platforms, like Atutor, Claroline and Moodle [13].

As AContent is an LCMS it meets some of our requirements, but would not fit for the labsystem integration well. Besides versioning it does not support the gathering of feedback and metadata how we need it. One feature we could derive ideas from is access control, but since this is already implemented with GitLab we can skip it.

All information written above was obtained from atutor.ca/accontent.

3.4 SAKAI

Sakai is another open source LMS. It provides support for over 20 languages and is already used in over 350 institutions. Being written in Java it can be modified and extended rather easily, as it is one of the most popular programming languages [7].

Even though Sakai is described as a LMS, the information on their website even suggests it being a LCMS: "Instructors and learners share knowledge, collaborate and engage in a wide spectrum of technology-enabled learning experiences, supported by an intuitive, modern user interface." This would fit the definition of an LCMS in section 2.2 very well, as content could be collaboratively worked on. But it does not fit our purpose. Sharing of content is done with a integration of Dropbox and thus provides no easy course selection, nor any overview of courses. Additionally the uploaded content is not versioned, as just the current versions are being uploaded to Dropbox.

The Roger Williams University in Bristol uses Sakai since 2010. It is an additive to their onsite courses. Like the Technical University Munich they have an administrative system coupled to their eLearning system. Every course created in the system is also created in the eLearning system called Bridges. A student enrolled in a course is likewise registered in the associated course in Brigdes. The Software is also used for tests and as a library for important links. For the latter it is extended to provide an overview especially for first-year students. Despite the slogan, about sharing and collaborating content, mentioned above Sakai does not seem to be fit for any of our purposes. Therefore, it is also not suitable as a reference for any of our requirements.

All information written above was obtained from sakaiproject.org, as well as hands on experience in the online trial set-up.

3.5 COMPARISON

As a conclusion of all related work it is showing that aside from moodle it is rare to have surveys regarding the content of a course, as opposed to survey testing the knowledge of a student. All try to improve the experiece and workflows in eLearning, yet none of the platforms provide a fitting way to exchange data between instructors. As there is no exchange between instructors there is also no urgent need for versioning, as there are less authors working with one project. Therefore most requirements do not fulfill most of our requirements. Especially versioning is fulfilled by none of the mentioned systems. Also the gathering of student feedback on content is almost not given at all. Consequently most of the tools above could fit as a partial replacement for the Lab-

system whilst fulfilling some of the exchange platforms requirements, but none as a decoupled exchange platform.

To summarize this section all related works mentioned above are listed and assessed in Table 3.1. Blackboard is divided to Blackboard Learn and Blackboard Open. Additionally added is whether a software is open source or not. The table shows what requirements are implemented by related works. The old and new requirements are relevant, as the system should fulfill as much as possible. The last row is about noteworthy features, that give ideas on how to find solutions to our problems.

	Moodle	Blackboard Learn	Blackboard Open	AContent	Sakai
Open Source	yes	no	yes	yes	yes
Import/Export	many standards	no	many standards	SCORM standard	Dropbox
Feedback	survey, forum	forum	survey, forum	no	forum
RBAC	yes	yes	yes	yes	yes
Versioning	no	no	no	no	no
Metadata	no	no	yes	no	no
Branching	no	no	no	no	no
Noteworthy features	modularity, standards	appealing design	modularity, standards, analytics	SCORM standard, more roles	none for us

TABLE 3.1: Summary of related work in eLearning

CHAPTER 4

DESIGN

The main goal of the system is to support the exchange to improve workflows in eLearning. In chapter 2 we found the research questions and identified the necessary requirements to achieve this goal. This chapter focusses on the design of the implementation of the exchange platform.

The first part of this chapter is about the renewal of the prototype. This means, what parts of the implementation shall be kept and what has to be reworked, or programmed from scratch. To keep the structure clear, the rest of the chapter is structured according to the requirements portrayed in 2.5. Since the requirements do not overlap, each can be discussed and designed separately.

4.1 REDEVELOPMENT OF THE PROTOTYPE

To decide whether to use the existing frontend, or redevelop it, we weigh the advantages and disadvantages of further implementing the prototype. First the **advantages** of keeping Julius Polars System. It is a working system. It is connecting the given components and thereby fulfills most of the requirements. It also provides us with code to keep building upon. Since the author has no experience with frontend development yet, it is not of considerable relevance which components he uses for this purpose. Thus keeping the prototype is a logical consideration.

The **disadvantages** of continuing to work with the prototype predominate. One of the main drawbacks of the work by Julius Polar is the use of AngularJS. AngularJS was first released in 2010, eight years ago, by Google. It is maintained by Google and an

active open source community [15]. Besides Google as founder it is one of the most starred projects on GitHub [5]. Even though the popularity of AngularJS seems as an advantage, due to more guides and online support, there are some disadvantages to it. There are options to upgrade AngularJS to Angular 2+ which was the first idea when starting to rework the prototype. Consequently AngularJS is compared with Angular, especially Angular 6 which is the newest version of Angular 2+. From here on Angular 2+ is referred to as Angular. Angular is an open-source project by Google, which had its first release in 2016 [16]. Angular is written in Typescript, which is a superscript to Javascript, the main language of AngularJS [15]. Further on it is faster, supports dynamic loading and is a little easier to learn [17]. Especially the last aspect is important, as the author has not worked with neither Angular, nor AngularJS.

As there was no handover of the project, possible gaps and with the old implementation are not known. This lead to some difficulties when first getting the exchange platform to run locally. Possible security flaws are also not known and the implementation, as it had little documentation and commenting, is hard to start with.

Furthermore the design of the prototype is rather rudimentary. As it is only using Bootstrap for design it does not look appealing and is thus not enjoyable to work with. Also the representation of survey data has to be reworked completely. All in all the rework of the platform would thus be quite time consuming.

As mentioned before the prototype already integrates all components. The login is made via the GitLab API. Here, username and password are used for authentication. GitLabs documentation recommends not to use this method, as there are more secure alternatives. Another possible security flaw are SQL injections. It is not documented how exactly the database is integrated, thus we do not know whether security measures were taken.

The advantages and disadvantages of continuing the work on the prototype are summarized in Table 4.1.

Advantages	Disadvantages
Existing start	AngularJS from 2010
Connects all systems used	Only Bootstrap for design
	Dirty Code
	No Documentation
	No Information regarding Security
	Unfamiliar with the Code

TABLE 4.1: Pros and Cons of Continuing work on the Prototype

This leaves us three possibilities. Either keep the frontend, learn deprecated AngularJS, fix all flaws and keep on building upon the existing prototype. Or start a hybrid Project

4.1 REDEVELOPMENT OF THE PROTOTYPE

with ngUpgrade, a official library to support a hybrid framework of AngularJS and Angular. Or build the frontend from scratch, maybe using some code, but using with a more modern framework.

As the disadvantages of keeping the prototype clearly outweigh the advantages the first possibility can be rejected. The hybrid approach brings the disadvantage of learning AngularJS and Angular. Instead of learning both we decide to use only the more modern Angular.

Even though this means a huge workload it is best to rebuild the whole frontend. After deciding to rebuild the frontend the other components can also be either discarded or reused.

4.1.1 BACKEND

The backend implementation can stay largely unchanged. A SQL-Database is an ideal solution for storing our survey answers and tokens and GitLab as a backend for our labs provides us with many useful services. Storing of all file formats we need, versioning them and it provides authentication. Therefore GitLab is reused, as the service fulfills all of our needs. One thing that shall be changed is the authentication flow. GitLab provides in their own documentation that sending the username and password is the least secure option and should not be used [6]. Therefore we can either setup our own database, or use another of GitLabs authentication flows. Setting up a database to administer users is extra work, but has the advantage of building our own design for user data. On the other hand it could bring additional risks if there are errors in our implementation. Therefore we decide to keep using GitLab for authentication. But instead of using the "Owner Password Credentials Flow", the "Implicit Flow" is used for our application [35]. Overall this brings some security advantages and simplifies the implementation of authentication. The implementation is especially easy as ngx-admin is used as a frontend template.

4.1.2 FRONTEND

After deciding building upon the prototype is not a valid option we decided to keep on building the frontend with a Javascript framework, as some of the code of the application can be reused. Angular is a Typescript framework that meets this requirements. The newest version of Angular 2+ shall be used, Angular 6. To give the implementation a jump start a template was also used. This template should already support authentication, a tidy layout and be written in Angular 6. The choice was the ngx-admin template, which supports all of the above [22]. To support the implementation, some ideas were

taken from the official Angular example *Tutorial: Tour of Heroes* [17]. Additionally it already supports ChartJS, a Javascript library for displaying a variety of different charts, out of the box. This can be used for representing data. The template uses the Nebular package, which in turn provides additional documentation and features.

As written above the Implicit Flow is used for authentication, ngx-admin template has OAuth2 as an implemented feature, instead of the Owner Password Credentials Flow. This in turn means the login redirects to the GitLab Server instead of providing login information on the exchange platform, which sends it to the GitLab.

4.1.3 PROTOTYPE REQUIREMENTS

This subsection provides a short overview on how the requirements found by Julius Polar will be implemented in our design.

The **Uploading and Sharing of Materials** is unchanged with the use of GitLab as backend. The upload is a direct flow between GitLab and Labsystem and thus remains the same.

As mentioned above **Role Based Access Control** is still implemented with GitLab as well, but changed from Owner Password Credentials Flow to the Implicit Flow.

Also **Material Versioning** is not affected by the new implementation as it is only done by GitLab.

On the other hand the **Listing of available Materials** is a frontend issue. The available labs are loaded using the GitLab Api. Some changes are to be made in terms of which data should be shown and design choices. These changes are described below in section 4.4.

The **Search of shared Materials** shall be implemented as before with a search bar searching through lab names and their descriptions. The "Recent Activity" module shall be dropped as it only provides technical, rather unnecessary information.

The survey for **Gathering and Analyzing Feedback about the Materials** shall be implemented in the same manner as in the prototype. Information gathered shall be saved in the SQL Database.

The **Display of processed Feedback** has to be reworked completely. As it is recorded as a standalone requirement it is described below.

The same applies for **Analysis and Visualisation of metadata**.

After a similar point was reached in the implementation, as the state of the prototype, each requirement is to be designed and implemented as follows.

4.2 IMPORT AND EXPORT OF LABS

The first step to increase the usability of the prototype is to include the import/export function. As mentioned in section 2.4 the prototype already includes working scripts. In this section the scripts are shortly explained again. These scripts should only be modified slightly and integrated in an appropriate UI solution. The first question arising is on which system to implement the scripts. As the exchange platform has no direct knowledge of which eLearning platforms are connected, it only makes sense to implement this functionality in the Labsystem. It should therefore also be executed from the Labsystem. Another advantage of using the Labsystem as a host for the import/export scripts is the availability of relative paths. This way users have to give even less input. To keep the structure inside the Labsystem simple, the user interface for the scripts is the same as for the integrated the current import/export. The existing export function exports the chosen lab as a folder containing all content. The folder also contains a `preview.html` file showing the exported lab the way it will look in the Labsystem. The export function is currently reviewed by Oliver Scheit as an Inter Disciplinary Project. As this is not part of this thesis it is only referenced here. To make the folder structure clearer, the exported folder will have fewer files due to the concatenation of related files. This way the versioning will become a lot clearer.

After the files are exported the scripts come into play. From here on the three scripts differ slightly. A detailed description of each script can be found in subsection 2.4.2. In short, there is one for creating a new repository on the exchange platform, one for updating the repository and one for importing a lab from the repository. Since the scripts were to be executed by console they do not include any error handling yet. They do also not include an export of metadata. After exporting the files with the given export function, the relative path to the exported folder is given in the Labsystem. The export to the exchange platform thus just needs to run this existing export from the lab and afterwards the script made by Julius Polar. The same goes for import the other way round.

4.2.1 AUTHENTICATION

For all scripts there has to be some kind of authentication, as not everyone is allowed to clone the labs, especially the solutions. The most common way to do this in git is by adding a ssh key to the repo. This would have to be configured in the Labsystem, which is possible, but unhandy. Another way would be to use username and password, but this is just additional security risk. The easiest way is using the OAuth token. As users

who want to download a lab have to be logged in they have to have an OAuth token. This token can then be displayed in the downloadlink. The link will then look like this: *https://<token>@GitLabURL/username/bar.git*. This guarantees authentication while being completely user friendly.

After obtaining the link to the repo, users head over to the Labssystem and enter the link and whatever additional information is needed, click a button and the respective script is run. After the script is run a page opens and displays whether the script ran successfully, or otherwise an error message.

Below every script is listed, with a simplified activity diagram showing the steps of the scripts. The first step, with the person attached, is the execution of a script with the according user input.

4.2.2 SCRIPTS

The first script is the **export** script. It needs the project name for the new GitLab project and the path to the exported folder. First it performs a http post to find out whether the name is already taken. If the name is taken, it returns an error on the console. This is to be changed. Either the user has chosen a same name at random, or should perform a git update instead if the expected behaviour was updating the files. This will be displayed on the error message on the opening site.

If the name is not taken it creates a new project, initializes a git repository, adds the solutions to .gitignore, adds all items to git index and pushes the initial commit. The step of asking whether to push the solutions is to be taken out, as the solutions should always be pushed. However they are pushed to a different group. The frontend token has no access to the solutions group.

After these steps the metadata is also taken as an SQL dump from the Labssystem and send to the exchange platforms database API. A chart of this script can be seen in Figure 4.1.

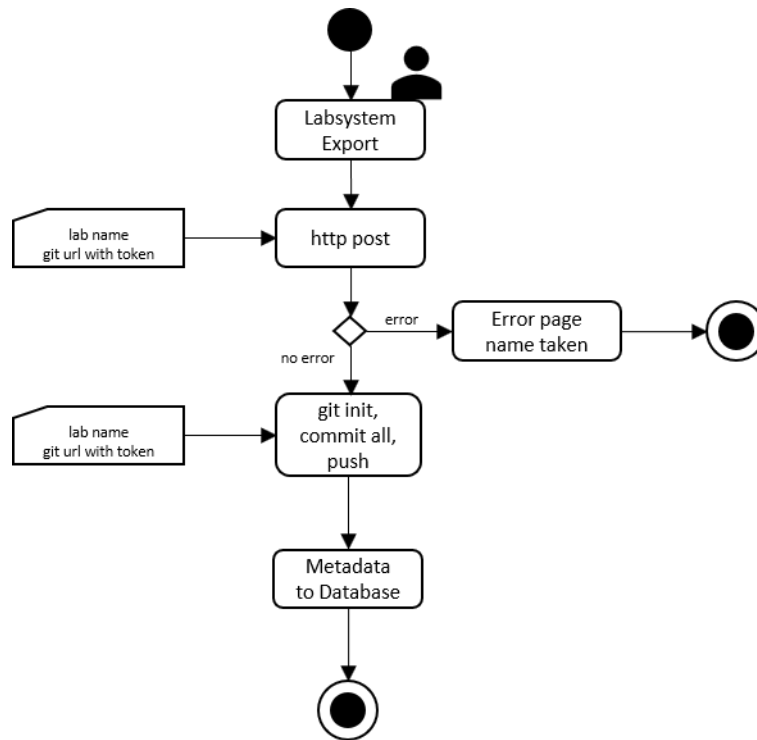


FIGURE 4.1: Export Script

Even simpler is the **import** script. It is used to checkout labs from GitLab. It only requires one parameter. The link to the GitLab project including authentication. The folder path of the Labssystem, which was needed in Polars script, is not needed anymore as it is a relative path given in the Labssystem. The script performs a git clone, creating a new folder which is then imported to the Labssystem. Here we need to consider if it should always clone the solutions as well. As there are exercises where students have to import labs they should not have access to the solutions. This can be avoided by not giving them access to the solution group in GitLab. As a consequence they cannot clone the lab. It could also be avoided by having a different option for students only. If one would choose the second way, students would be able to read the solutions in GitLab which is to be prevented. Consequently we decide on the first idea of not giving them access to the solutions group.

The Import script does not import metadata, as the Labssystem generates all necessary tables and data. Therefore there is no mention of metadata in Figure 4.2.

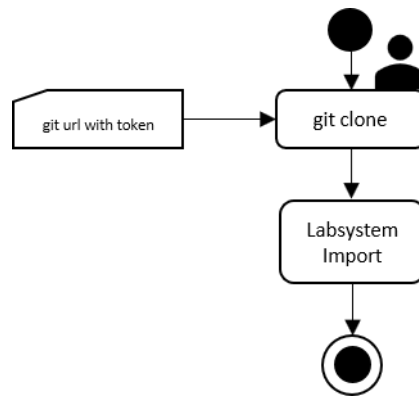


FIGURE 4.2: Import Script

A bit trickier is the **update** script. It needs to: Update data from the Labsystem to the exchange platform, Update data from the exchange platform to the Labsystem and update metadata from the Labsystem to the exchange platform, if there is any. This last aspect was not mentioned before as it only affects this script. This script is more difficult than the others is the possible occurrence of errors that could not occur in the previous cases. Julius Polars version of this script only pushes data to the exchange platform and thus decreases the error susceptibility. But if more people from different instances of the Labsystem want to work on one lab this does not suffice.

The script needs the path to the directory, as well as the link to the GitLab project and a commit message as parameters. It combines the export and import scripts before, without creating a new repository.

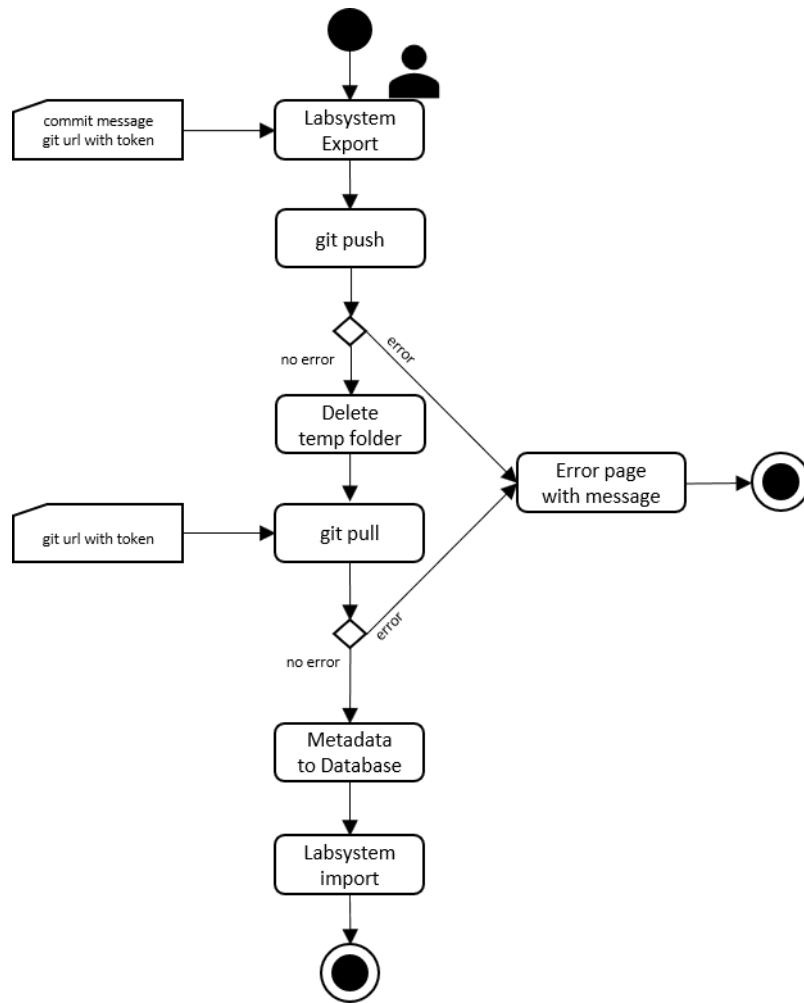


FIGURE 4.3: Update Script

As a summary Table 4.2 below lists the scripts with their respective flow of data and needed user input. Additionally it lists possible errors, under the assumption of link and token being valid and therefore ruling '404 not found' and '401 unauthorized' out.

Script	Dataflow	Input	Possible Errors
Import	Exchange -> Labsystem	Path to GitLab (including token) Future Name in Labsystem	Solutions not cloned
Export	Labsystem -> Exchange	Path to GitLab (including token) Future Name on Exchange Platform	Name Taken
Update	Exchange <-> Labsystem	Path to GitLab (including token) Current Name in Labsystem	Merge Issue

TABLE 4.2: Dataflow and needed Input for Scripts

4.3 STUDENT FEEDBACK SURVEY

4.3.1 DISPLAYING GATHERED FEEDBACK

This feature was already implemented in the Prototype by Julius Polar. Even though his requirement *R06 Display processed feedback* is somewhat implemented it does not provide instructors with an easy overview over the gathered data. There are two major issues:

The survey results are presented on an extra site. Even though the data is only one click from the lab detail page, the button to the survey results might easily be missed. The simple solution to this problem is just moving the results to the lab detail page.

The second issue is the presentation of the gathered data. As Polars survey showed, the current display of survey data is unsatisfactory. To address this issue the graphs are completely reworked from scratch. The questions stay the same, which leaves us with three questions in need of a graph.

How difficult was the lab for you? How interesting was the lab for you? How would you rate the lab?

The other questions are displayed in textform as a graph would make them more confusing. According to Freeman et al. a person interested in data, without having good statistics knowledge, will find graphs easier to understand than tables. Additionally a bit of variety in the graphs can help to make the data more user friendly and faster to process [10]. Therefore we decided to use a mix of graphs. A pie chart for the rating, and bar charts for the other questions.

4.3.2 INTEGRATION IN THE LABSYSTEM

Currently the exchange platform only offers the possibility to generate the URLs for the survey. Then the instructor can give them out to students at will. These URLs contain the lab to which the survey link belongs, as well as a token used to authenticate the user as a student. Each token can only be used once and is afterwards removed from the database. After more than one student answered the survey there is no way of telling which student answered what, as the database does not save this data. Consequently this provides an anonymous one-time survey for each student.

The survey links are now provided to the instructor and have to be given out to students somehow. This could be done two ways. Either by sending an email to each student, or by integrating a distribution of survey links in the Labsystem.

The best way to collect as much real data as possible with a survey is to show the link to the survey directly after the assignment [31]. Therefore the Labsystem should

get the links from the exchange platform and distribute them to each student. This can be included in the update and import script at the beginning of each semester. An instructor would then only have to give the number to the script and the links are automatically distributed. This would also result in greater privacy, as the instructor does not even get in touch with the links.

4.4 ENHANCED BROWSING EXPERIENCE

One of the arguments for completely reworking the exchange platform was being able to redesign the whole system. The goals of the design are: to be informative for other universities and to be pleasant to work with. Upon opening the prototype a user was directed to the catalog with all available labs. If somebody only visits the page for information this setup would only lead to confusion.

Therefore start page is needed that contains information about the courses and arouses interest. This is the requirement [**R02.1 Add Landing Page**]. An instructor coming to this page will search for answers to: What is this? Why should I be interested? These questions are to be answered at the top of the page. Ideal for answering FAQs on a landing page is Nebular Stepper Component [24]. The content to answer these questions will be filled out, but can be easily changed, as it is only basic html.

The other question to answer is why an instructor from another university should care. This is partly answered with the questions, but should somehow be supported with data. The best data to present here is the number of students over the years. A graph shows the number of students per university and course. It also gives an accumulated overview. This landing page can easily be extended easily, as it is just html code.

The current prototype implements a catalog with a search function, the labs as a list in the middle and recent activity on the left. As it provides little information requirement [**R02.2 Catalog Rework**] is for rebuilding the catalog, to look more pleasant and provide more information. This recent activity showcase shall be dropped as it provides no real further information. It would show any recent work on git files, which confuses people who do not know about the git backend.

The main catalog looks very simplistic, as it shows a list of boxes including the labname and a short overview. As an instructor this overview gives neither information about whether students like the module, nor how long it takes and how difficult it is. Therefore the overview is to be extended with additional information. A bar chart, with one bar, so basically a progress bar, showing the rating of the students and in one sentence length and difficulty. As there is no measurement of difficulty we use the overall credits achieved in percent.

The last subrequirement is [**R02.2 Overall Design Rework**]. In chapter 3, the best frontend was Blackboard LMS, shown in Figure 3.2. Therefore we orientate our implementation to their design. This can be done by using the themes provided in nebular [23]. Additionally reworking the catalog and lab detail page already improved the overall design. Using the ngx-admin starter kit provides everything for this task, so it is not further discussed here.

4.5 ANALYZING AND PRESENTING METADATA

The outcome of this design should answer the question of what metadata is relevant for instructors when choosing their courses. Metadata is used to describe other data. As Bargmayer et al. state this definition of metadata is easy but not precise. The important part for metadata is context [2]. In case of eLearning the current standard for metadata is IEEE LOM. The IEEE Standard for Learning Object Metadata [2]. It defines 80 fields. These are categorized in General, Lifecycle, Meta-Metadata, Technical, Educational, Rights, Relation, Annotation and Classification. This standard is often extended as need be. As we only need human readable metadata we have a lot smaller scope. The other aspect that restricts us is the data collected in the Labsystem. Especially the second one restricts our scope to the following:

Points achieved and time taken.

The simpler of those two topics is **points achieved**. It should show as the mean of points achieved per student in percent. This way a instructor can decide which kind of difficulty the students should solve. The data is saved in the SQL database of the Labsystem. To get this data in the Labsystem it is exported together with the labs. This way on each export/update the current version is available on the exchange platform. As for representation, it should be represented in the catalog, because it is interesting for most people. As the catalog should be as simple as possible there cannot be any complicated charts. The simplest of charts is a single bar, which is sufficient and fits the design.

The other relevant metadata gathered by the Labsystem is the **time taken** per lab. As mentioned before Sarah Jost is writing her bachelors thesis, Smart eLearning Analytics, to improve the current time tracking algorithm implemented. As her thesis is still work in progress the following description is about the current implementation of said algorithm.

The Labsystem has a huge aggregation of log files logging basically every click of a student. Consequently we are provided with a lot of data. The algorithm takes the first click in the lab as starting point and the last click as end point. If no click occurs

for more than 30 Minutes the algorithm treats the last click as end time. If a student continues, e.g. after a break, the duration of the first period is added to the second. Even though this algorithm is rather inaccurate, it provides us with a rough overview. The time taken for each lab is transferred to the exchange platform the same way as the achieved points. It could be also shown as a chart, but as it is aggregated to one value it is just shown beneath the lab description in text format.

4.6 BRANCHING OF A LAB

One of the aims of the exchange platform is to support the exchange of learning content between universities. So far the system provides the possibility to import/export labs to/from a Labsystem. One case that may occur is that an instructor would like to take a lab and slightly modify it, e.g. delete one question because the lab seems too long. The question arising is how to implement the branching of labs on the exchange platform. The easiest way of doing it is just downloading and uploading again. But this bears some workload for an instructor and having the same name twice is not possible. As we already use git, one possibility would also be to use branches in git. This way one can have a branch off of a lab which can be modified in any way. Although that seems like it could fulfill the requirement, this holds one major issue. The consistency of gathered feedback. Branches in git usually aim to be merged back together at some time. This will not be possible with feedback in mind, as one lab will most likely have changes to it which will affect the feedback of students. Thus we have the option of allowing branches which will never be merged again, or allowing branches that will be merged at some point with dropping the feedback, or keeping all feedback. Another problem that comes with this is the complication in using the GitLab API. So far we get project name and id with the API. This would not be complicated a lot if we allow branches, as the main branch would have to have the old name and any other branches different names. The main problem occurring here would be the whole routing would have to be changed, as it is oriented on the project ids.

Since the thesis already implements a complete renewal of the prototype, the feedback and API problems become a too big challenge, so this requirement will not be addressed further.

CHAPTER 5

IMPLEMENTATION

5.1 SETUP

This section briefly explains what components are used in the implementation and how they interact with each other. As the implementation is very similar to Julius Polars, this section is kept rather short. Nevertheless it should help with generally understanding the system and installation. In Figure 5.1 a simplified summarization of the setup is given. Afterwards an example scenario is given. This scenario is copied from Julius Polar, as there were no changes in the workflow.

Backend: The main backend for the implementation is GitLab. In GitLab we have two groups. One for the lab files, another one for the solution files. This provides easy administering of access to solutions with the use of tokens. These tokens are generated when creating a GitLab account and can be set according to the user role.

The export script, in the Labsystem, creates a new project in GitLab in both groups. This way the access is automatically set to the group. Update and Import script work as depicted in chapter 4.

The frontend uses the GitLab API for showing the catalog. This is explained below.

Frontend: The frontend is written in Angular 6 [16]. As starting point ngx-admin [22] is used, together with nebular [23] for components and themes. Nebular additionally provides an OAuth2 workflow already implemented. For further information see [24]. This documentation also provides everything for further development.

The whole frontend is written with modules instead of the components used by Julius

Polar. This helps further development, as modules can easily be added and removed as wished.

The data in the catalog is retrieved using the Gitlab API. For this the frontend has an extra GitLab account with rights to only see labs in the lab group, **not** the solutions group. Even if one were to retrieve the token, there can not be done any harm with it. The feedback is imported via Spring API.[29]

Database: For the feedback a simple SQL Database is used. Connected to the Frontend with the Spring API it allows to read survey data, but injecting is only possible if the token in the survey URL is existing in the database. After a token is used it is deleted in the database.

Labsystem Integration: The Labsystem and Frontend do not have direct interaction. All interaction with the Labsystem so far are the scripts discussed in chapter 4. In the future the feedback URLs could also be integrated in the Labsystem.

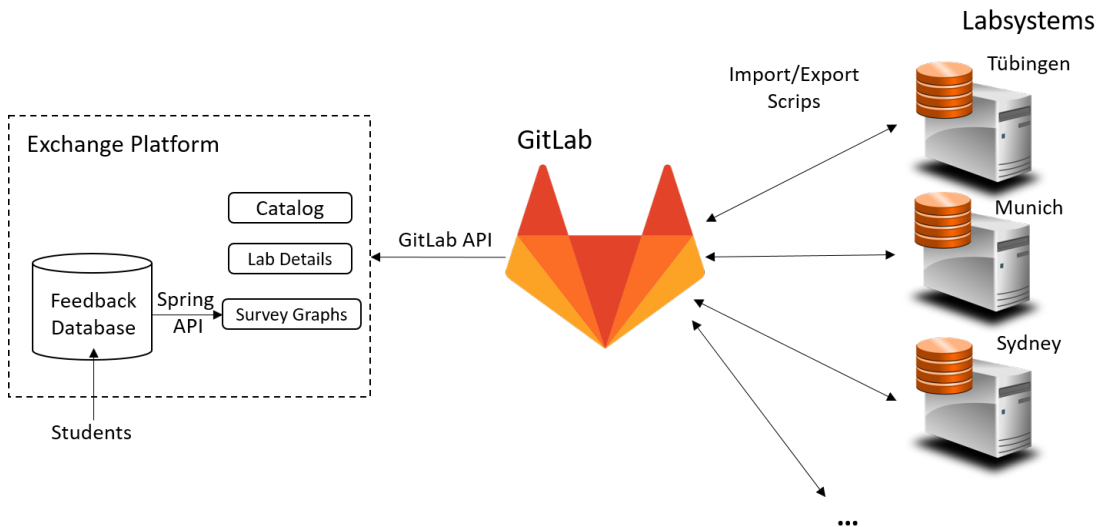


FIGURE 5.1: System Components and their interaction

Example Workflow: In Figure 5.2 the workflow of setting up a new lab and generating survey URLs is given. This workflow was already shown in Polars thesis. An admin creates a new lab on the Labssystem and exports it to the exchange platform. Then he/she logs in the exchange platform and generates the survey URLs. These URLs have to be sent to students via eMail. After a student answered the survey, the token is automatically removed from the database.

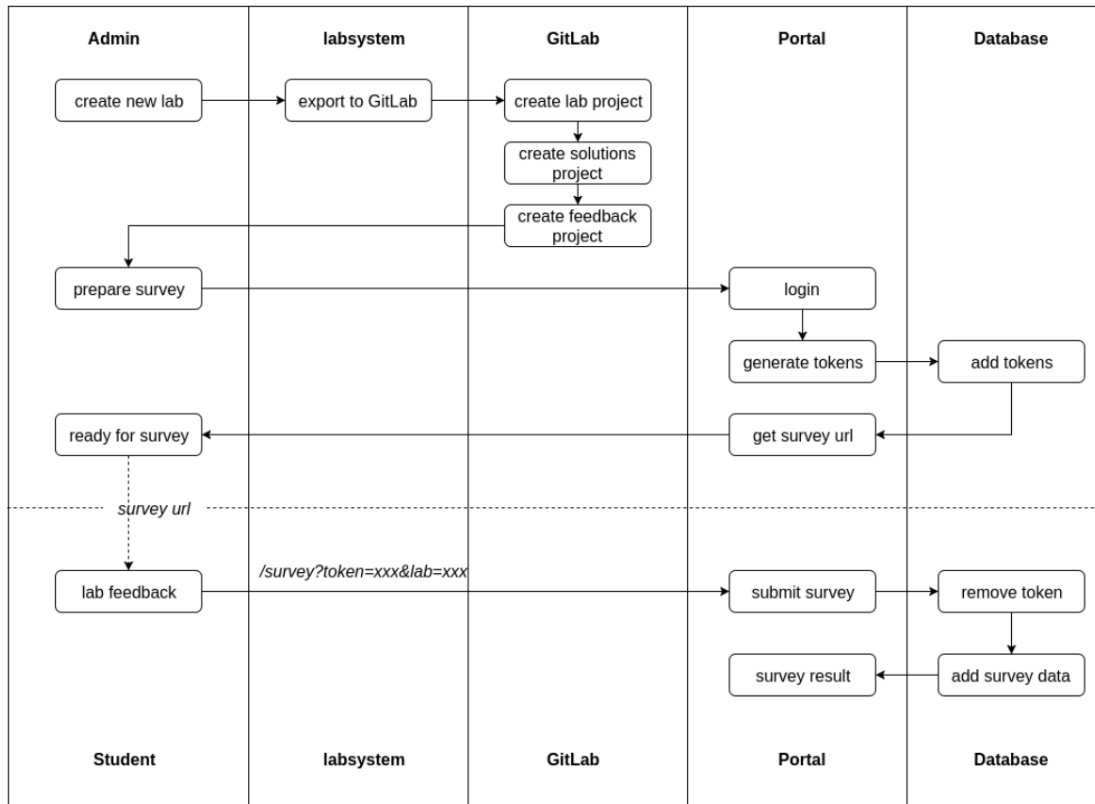


FIGURE 5.2: Sequence diagram of setting up lab for survey [30]

5.2 COMPARING PROTOTYPE AND NEW IMPLEMENTATION

This section briefly compares parts of the prototype to the new implementation. This comparison shows the enhancement in terms of the design and usability made in this thesis. By displaying some screenshots the following workflow is shown on both systems: An instructor, already logged in, comes to see what overall feedback students gave on a lab. The two implementations do not share the same backend data, but the different datasets suffice for this comparison. First the workflow of the old implementation is shown and afterwards the same workflow with the new implementation.

PROTOTYPE

When opening the exchange platform the user is directly shown the catalog of all labs, as seen in Figure 5.3. Bearing the advantage of having one less click to get there, it is not very comfortable for new users. The website overall is rather simplistic.

The old Catalog only shows labs and their description. Tag filter and Recent Activity are also shown. Clicking on the card of a lab opens the according lab detail page.

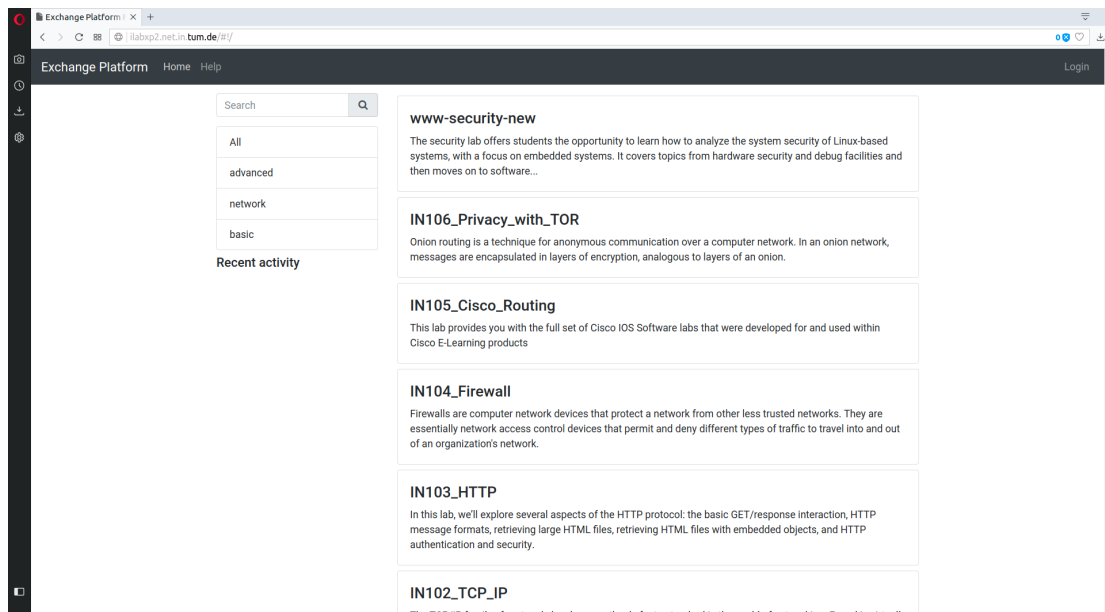


FIGURE 5.3: Catalog of the prototype

The corresponding picture in the new implementation is Figure 5.7.

5.2 COMPARING PROTOTYPE AND NEW IMPLEMENTATION

The lab detail page shows the description of the lab, which has to be changed in the GitLab. Also an iframe and git access is shown, as seen in Figure 5.4. Below that is a button to view the survey results. On click the survey results are shown. It does not matter if a user is logged in, this data is publicly available.

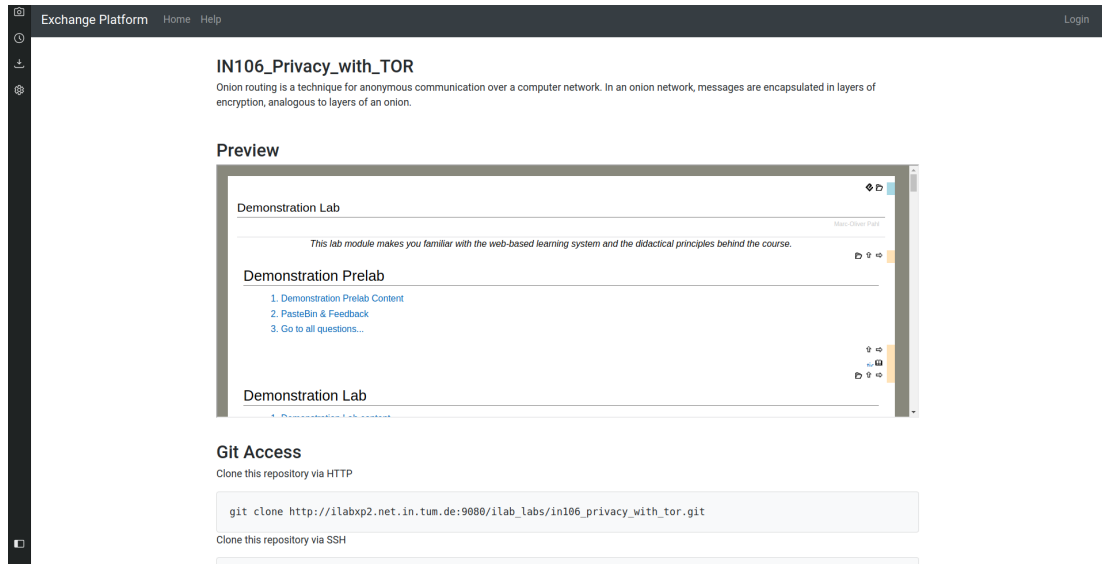


FIGURE 5.4: Details of a lab

The corresponding picture in the new implementation is Figure 5.8.

On the top of the page the instructor can choose which branch he wants to view, even though technically only one is possible. For each survey question there is a bar chart. Additionally calculated are Average, Median, Max, Min and Standard Deviation. With having some statistical knowledge the instructor has achieved his goal of viewing the feedback. The feedback site is seen in Figure 5.5

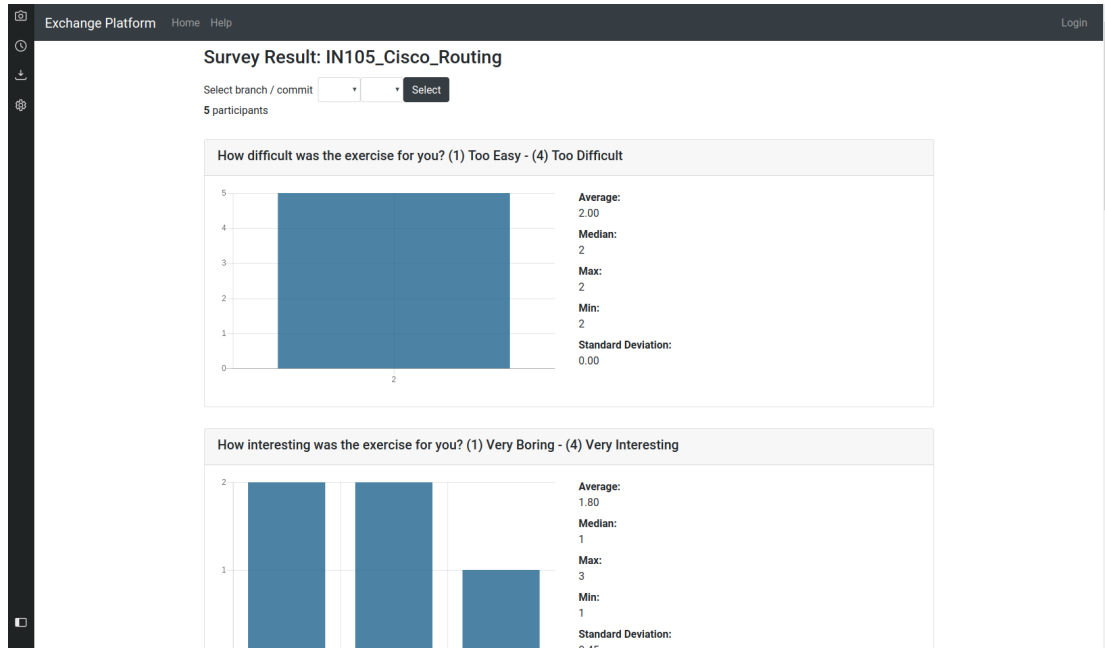


FIGURE 5.5: Student Feedback of a lab

There is no corresponding picture in the new implementation, as the feedback is displayed at the detail page of a lab. The feedback is therefore included in Figure 5.8.

5.2 COMPARING PROTOTYPE AND NEW IMPLEMENTATION

NEW IMPLEMENTATION

The same workflow is shown in the new implementation. Some details on the pictures below may still be subject to change.

Upon opening the webpage the user is greeted with a landing page shown in Figure 5.6. On this page instructors can find general information on the iLab and the Exchange Platform. Additionally a graph shows the number of users over the years.

Besides providing a 'left-to-right' <-> 'right-to-left' menu switch, every page holds the possibility to switch to one of three different themes. Clicking on 'Catalog' in the menu brings the user to the catalog.

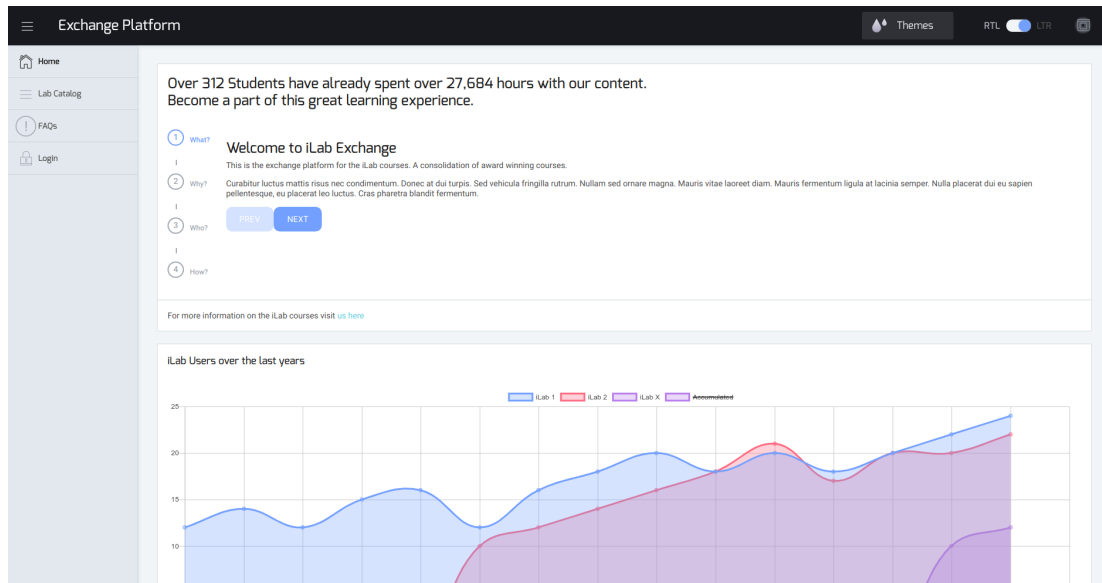


FIGURE 5.6: Landing Page in the new Implementation

There is no corresponding picture in the old implementation, as the prototype did not feature a landing page.

CHAPTER 5: IMPLEMENTATION

The Catalog, shown in Figure 5.7, displays all labs with their description. Additionally it shows the overall rating, as well as average time and points. Recent Activity and Tag Filter are removed, while one could think about reimplementing the latter. If logged in, clicking on the card of a lab opens the according lab detail page.

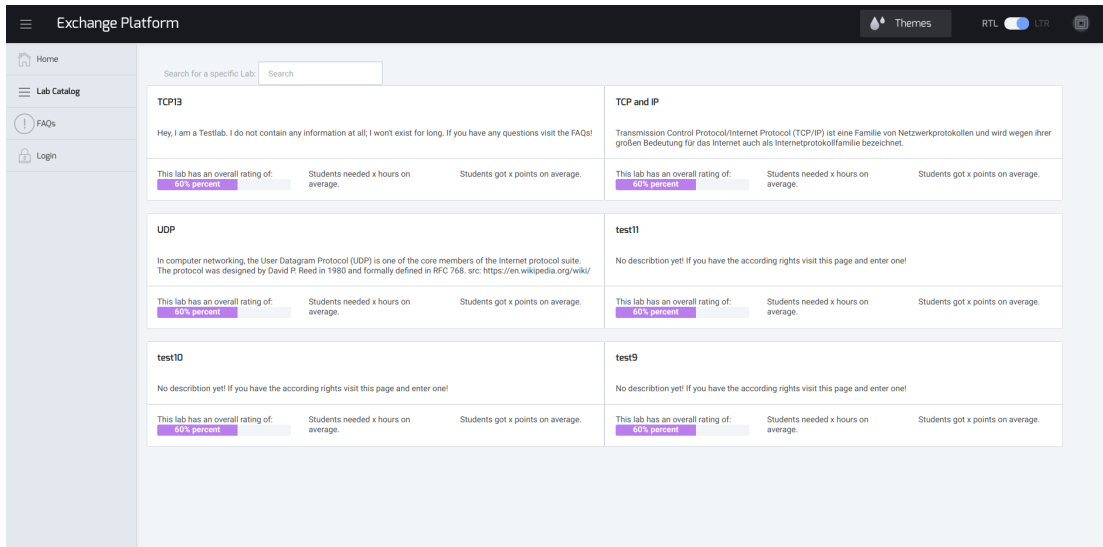


FIGURE 5.7: Catalog showing all labs

The corresponding picture in the old implementation is Figure 5.3.

5.2 COMPARING PROTOTYPE AND NEW IMPLEMENTATION

The lab detail page shows the description of the lab, with the link to change it. The same information as in the catalog is repeated and a chart for every survey question shown according to the findings in subsection 4.3.1. Below is the iFrame for the preview and the download link for the Labsystem. The site is shown in Figure 5.8. Here the instructor sees all information necessary on one page. His goal of checking the student feedback is achieved.

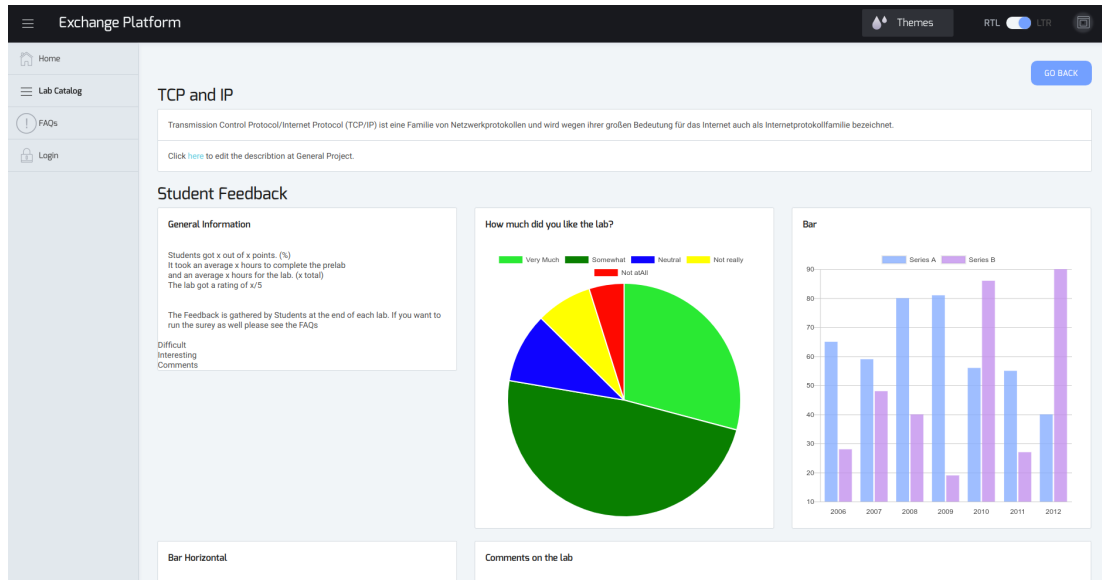


FIGURE 5.8: Lab details

CHAPTER 6

EVALUATION

In this thesis the frontend of the prototype was completely redeveloped. Thus we need to evaluate against the old and new requirements. To assess the design and implementation of the exchange platform we start with the evaluation according to the old requirements. Since Julius Polar has already implemented and evaluated these requirements, and we are following up on his implementation, we can compare the implementations on the basis of his evaluation. For each requirement we show where we have exceeded, or are missing something, compared to the prototype. Afterwards we evaluate what Polar has not yet, or not satisfactorily, implemented in his work.

6.1 OLD REQUIREMENTS

In this section we only evaluate the requirements we found as fully implemented by the prototype in chapter 2. There are two parts in Polars evaluation. First a comparison with related work, and afterwards a user evaluation survey. Below we run both evaluations with the new implementation.

6.1.1 COMPARING TO RELATED WORK

The assessment of these requirements is carried out in accordance with the work of Julius Polar. We will analyze the workflow like he did, if there are no major changes. If a part of the workflow was one of the new requirements in this thesis, it will be evaluated afterwards. He analyzed the following workflows:

Giving Lab Feedback is implemented similar in both theses and thus evaluated here.

The requirement 'Survey Integration' is handled separately as a new requirement. Most online courses provide feedback opportunity with a rating and comment, e.g. coursera [9]. For this, students need to be registered and enrolled for authentication. Both exchange platform implementations do not need this kind of authentication due to the token system. As the URLs are neither personalized nor the token for each survey is saved, privacy is fully given and integrity of the data studentwise is guaranteed. But an instructor could still generate tokens and answer surveys. Since the exchange platform will only be used in non-commercial applications, we count on the reliability of the instructors, especially since the feedback is in their own interest.

Viewing feedback result was completely reworked, so it will be evaluated with the new requirements.

Browsing labs was fully implemented by Julius Polar and done similar here, so it will be evaluated against his work.

Like most MOOC (Massive Open Online Course) platforms he chose to have a catalog displaying the available courses publicly [32]. The implementation in this thesis even surpassed this catalog, by adding more information to it, without it getting too crowded. A search function is also provided, a tag filter, as well as a grouped courses can be considered for the future, but are not implemented in this thesis. A comparison of the catalogs can be found in chapter 5.

Creating and sharing labs is a workflow not directly specified as a requirement, but is also evaluated against his work.

As before, labs are created in the Labsystem. Transferring it to the GitLab is done via the export script. Here we are exceeding the prototype by including the script in the Labsystem. This was also mentioned in Polars future work. Moodle for example also has a separate page for exporting content, but also to export only specific content, but as the labs are too specific this does not fit our scope.

Importing labs is also a workflow and not a directly specified requirement. As the one above it was newly integrated in the labsystem. Moodle, IEEE ComSoc Lab Exchange as well as the exchange platform all have a portal site to browse shared labs. The download is also in all three in two separate systems [30]. Our implementation exceeds the prototype by integrating the script in the Labsystem. An instructor now only needs to copy a link to the Labsystem and click import. The proposal for including a lab search in the Labsystem is not implemented,

as the exchange platform and Labssystem are intended to act as two low coupled systems.

Versioning labs has not changed. As it is only feature of GitLab there is not much possibility for change here. It is not especially displayed in the prototype, but much rather an option if some lab files are tempered with.

6.1.2 USER EVALUATION SURVEY

The other method of evaluation in Julius Polars thesis is a survey. Here he separates evaluation of the survey and of the Labssystem integration. As we reimplemented the survey almost identical, this survey is especially interesting for us. The survey about Labssystem integration is only briefly looked at to give an overview, but the assessment of Labssystem integration is done with the new requirements.

The first survey is about the **feedback survey** for students. Participants are asked to run the survey as a student. For this, they are provided a link, just like a student would. Afterwards they should inspect the survey result as an instructor in the exchange platform. The user experience form provided by Polar contains the following 8 questions. These questions are shown in Table 6.1, whereas 'Likert' is a likert scale in 5 steps from strongly disagree to strongly agree.

Question	Style
1: The survey process is easy to follow and intuitive	Likert
2: The user interface of the survey process is intuitive and informative	Likert
3: My opinions on the subject is represented in the survey questions	Likert
4: The data visualization is intuitive	Likert
5: The data visualization helps me understand the data	Likert
6: The data visualization shows information relevant to my interests	Likert
7: Rate your experience	Rating
8: Do you have a particular feedback?	Comment

TABLE 6.1: Julius Polars Feedback Survey Questions on Student Feedback

As this is an overview, the diagrams are not displayed, but only roughly summarized for each question.

1. *The survey process is easy to follow and intuitive*, was mostly answered with agree and strongly agree. As we implemented the process identically this also applies to our implementation.

2. *The user interface of the survey process is intuitive and informative*, was also mostly answered with agree and some answered with strongly agree. We extended the information flow even more, resulting in an even better user experience.
3. *My opinions on the subject is represented in the survey questions*, is answered around agree, with more neutral answers than strongly agree. Even though we are providing the same questions, these are easily exchangeable in the sourcecode. There only has to be an html edited, as well as one line added in a typescript file.
4. - 6. were answered very diverse. For most participants the implementation of visualizing data was insufficient and confusing. As we redeveloped this completely it is evaluated below.
7. *Rate your experience*, does not make much sense to discuss, as half the survey content is changed.
8. *Do you have particular feedback?*, This feedback is almost completely about the visualization being insufficient. One comment is about missing form validations, which were implemented in the new exchange platform.

Summarizing the above to the new implementation: Any components we implemented identically were perceived as good. Insufficient parts were reworked and the very unsatisfactory feedback display was completely reworked.

For the complete survey results consult [30].

6.2 NEW REQUIREMENTS

After analyzing the base of the new implementation, the additional features and new requirements need to be analyzed. As they are almost completely unattached we can easily assess the implementation one requirement after another. To evaluate new requirements, as well as requirements we did not see fulfilled in Polars thesis, we divide this chapter in two. The first part is a comparison with related work. For every requirement we compare with the best solution from any of the related work from chapter 3. Afterwards we conduct workflow walkthroughs with testers with different level of technical experience. As the frontend was completely re-implemented not all requirements are implemented. For the purpose of completeness, these requirements are nevertheless listed here.

6.2.1 COMPARING TO RELATED WORK

With Polars evaluation as example, we start comparing our implementation against the design choices of related work. For this we choose the closest related work(s) and find advantages and disadvantages in both systems. As this is more a comparison than evaluation, this section is kept rather short.

ONE CLICK IMPORT/EXPORT

During the design of this requirement, we were told that Oliver Scheit was currently revising the lab system. So he was given the updated scripts. The implementation isn't finished yet, so only the concept will be evaluated.

The main goal of this requirement was to get rid of the console, which we achieved. An instructor now has to get the link from the exchange platform and can import it in the lab system. Even though this is not 'one click', the link is on the second platform, this requirement is fully implemented. The same workflow is also implemented in Moodle and AContent. On both platforms some reference has to be entered, which is not directly found at the import function.

Overall we saved instructors a lot of time when exchanging labs. The reduction of steps for one exchange can be seen in Figure 6.1. On the left side is the old workflow with 7 steps, on the right the new workflow with only two steps.

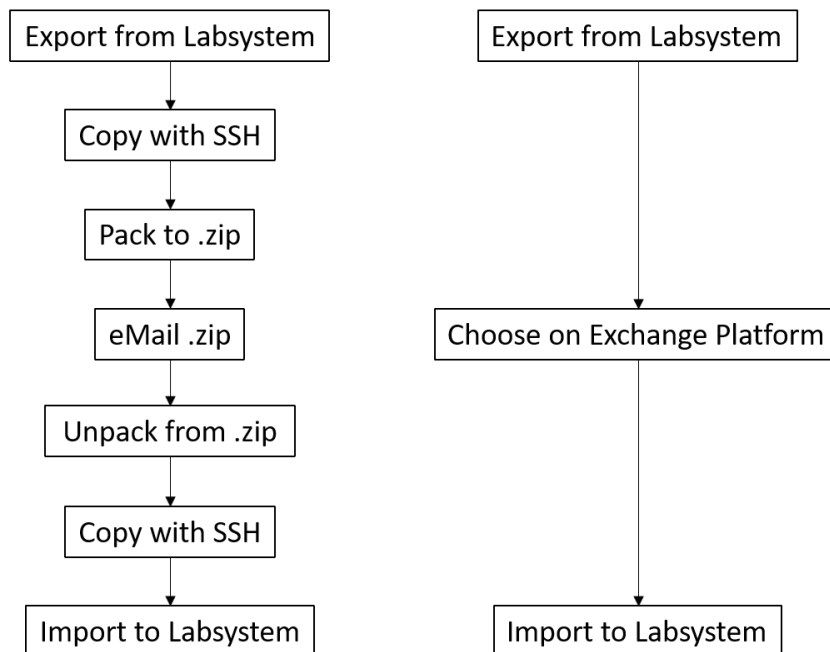


FIGURE 6.1: Workflow Comparison of exchanging a lab

ENHANCED BROWSING EXPERIENCE

This requirement was subdivided in three smaller chunks. Each of them is evaluated individually.

Landing Page: A landing page was added to the exchange platform. It provides information about the iLab, the exchange platform and the universities taking part. It also includes links for further information. As our related work does not implement similar ideas it is not possible to compare it with them. The evaluation is thus done with the workflows walkthroughs below.

Catalog Rework: The Catalog was slightly reworked. One definite improvement is the inclusion of metadata. No related work include a similar catalog, but there can be one found at Class Central [8], a website for open courses for Blackboard. This catalog shows the name of the course, the university at which it is held, a start date and an overall rating. If available it also shows the duration of a course. In our implementation we included the duration and the rating to the catalog as well. We also include a short description, which can be changed in the lab detail page, which was not possible in Polars work. These improvements are also asked for in Julius Polars survey. Overall this lead us to a design closer to MOOCs in general. Not only Class Central, but also Coursera, one of the biggest MOOC providers, provide similar information on their catalogs [9][8].

One thing we do miss is some sort of tag filter for the labs. This would be especially useful for filtering universities. It is mentioned in future work. The other thing we miss, compared to Polars work, is the recent activity display, which we decided to discard.

Overall Design: Design improvements are hard to measure, as anyone has their own opinion. For our implementation we oriented ourselves on the design of Blackboard [4]. Additionally we used the ngx-admin theme switcher to give each user the option to choose their preferred design [22]. Additionally we included chartJS to implement better looking graphs. Overall this should result in a better browsing experience. This will be evaluated in the walkthroughs as Nielsen recommends [26].

Like Julius Polars work, the exchange platform is designed to be easy to use even on a mobile phone. This is especially important for students, as they are able to fill out the survey on their way home.

STUDENT FEEDBACK SURVEY

The main research question for the survey is: *How can we gather as much student feedback as possible, while keeping privacy and data integrity?* The german 'Bundesamt für Sicherheit in der Informationstechnik' (BSI), sees privacy fulfilled, as long as only a person itself has the possibility to publish private data to anyone else [18]. As we do not save any connection of a person to data, but only the data given by the person, privacy is kept.

Display of Feedback: Julius Polar's survey showed that this part of the implementation needed the most work. We moved the graphs to the lab detail page and implemented different graphs, illustrating the survey answers more fitting. Schonlau et al. state that there is no big difference in which graph to use for a specific variable. [33] Therefore different graphs were used to give the page a more pleasant appearance. This becomes particularly clear when comparing the walkthroughs with the survey. Feedback from the test group that tried the new implementation was consistently positive, while the other group found the prototype unsatisfactory in this respect. A direct comparison can be seen in section 5.2.

The only related work implementing student feedback on courses is moodle. The new implementation surpasses their feedback tool by providing more information and different, tailored graphs for each question.

Survey Integration: Was not implemented, as the time was too short after re-implementing the whole frontend. It is thus listed in chapter 8. The recommended workflow for the current implementation is sending the links out via email at the beginning of the week and setting up a reminder at the end of all labs.

METADATA

Julius Polar already wanted to include some sort of metadata in his work. He also came to the conclusion, that almost all MOOCs display metadata in their catalogs. The two examples before, Coursera and Class Central, both show the overall time of the courses [9][8]. This was also implemented in this thesis.

We extracted the metadata from the Labssystem and it is transferred to the exchange platform when the course is updated. Even though we don't know how exactly related work do this, it is sufficient for us. The display of metadata is done similar to MOOCs and also to Blackboards Open LMS [4]. They also include data about students, which

is not possible for the exchange platform.
Overall this fulfills the requirement.

BRANCHING

As the possibility of branching manually exists, meaning downloading a lab and uploading it with a different name, this requirement was assigned the lowest priority. Due to the difficulty of merging the feedback of two different labs this idea was discarded and thus not implemented. It is listed as an idea in future work, but not recommended to implement, as it could make the feedback of the different modules muddled.

6.2.2 WORKFLOW WALKTHROUGHS

To evaluate the implementation regarding usability, we decided to use walkthroughs which are part of the recommendations by Nielsen [26]. During the development we implemented the requirements using heuristics. The workflow walkthroughs help us to evaluate the system. A full test of the system would be perfect with real students, but is not possible with the given time frame.

The general **idea** of the walkthroughs is to simulate the usage of the platform with users of different experience level. For this we have three testers and three tasks. All three testers are not familiar with the system yet.

Before a task the testers are only given the necessary information. This way they have to use the platform to find out the information wanted. During their tasks they are asked to think out loud. The whole process is protocolled. The protocol can be found in Appendix B. After each task the testers are asked what they liked, what they did not like and how they would rate their experience. In between the workflows minor tweaks were done to the platform, which only contain wording.

For our walkthroughs we have three **candidates**, with different technical experience. One Law-Student with no further experience, besides usual social media, smartphone and laptop usage. One business consultant who worked in IT some time ago, but has not been up with the latest trends and can thus be described as a more experienced user. And one information systems student, who is a very well experienced user. The three different levels of technical affinity shall represent different views and expectations to a websystem. All three were selected from the authors peer group.

The walkthroughs are divided in three **tasks**. The first task is to inform themselves after they heard of the course and exchange platform. For this the testers do not get additional information, or details on the platform and thesis.

The second task is to fill out the student feedback survey. Even though this is a short and simple workflow it is really important, because students have to feel comfortable giving feedback, if maximum amount of feedback is wanted.

The last task is to have a look at labs and act like an instructor browsing labs. This means to take a look at different labs and decide which one to take into account for downloading it to a course.

FIRST TASK

The first task is getting information about the exchange platform. For this the testers are provided the following:

Scenario 1, Role: Instructor

You have just been on a conference about the Internet of Things. A colleague of yours came up to you and recommended you an eLearning course about the Basics of the Internet. He gave you a link to a platform where they exchange their courses. As you are an instructor at a university you are interested and want to inform yourself.

You got the link via email. Feel free to use any device and inform yourself.

We **expect** the testers to go to the main site and read through the info panels on the first site. As this task is work related for an instructor, most testers will probably use their computers for this, not their mobile devices. Meanwhile they may digress briefly and look at the chart at the landing page. Afterwards we expect them to click on *visit us here*, to gain more information about the labs and maybe contact people accordingly. It is also possible that testers might have a look around the site, but we expect them to find back to the start page.

The actual **outcome** of the experiment was mostly as expected. There was some dissatisfaction with the text, but wording is still subject to change. Only one tester went on and clicked around the exchange platform, the others directly clicked on the 'visit us here'.

Overall the testers were rather satisfied. They liked the design of the website and felt informed. Most things they did not like were fixed and reimplemented.

The workflow was implemented satisfactorily. Some work could be done to the wording, but as it is only html it can be adjusted easily at any time.

SECOND TASK

The second task is just filling out the feedback survey after a lab session. For this the testers are provided the following:

Scenario 2, Role: Student

(Out of protocol the tester was shortly informed about the iLab workflow) You just finished your lab. After this long day of work you are reminded at the end to have a look at your emails and fill out the feedback survey. Again, you just finished 6 hours of intense work with your teammate.

You got the link via email. Feel free to use any device and fill out the feedback survey.

We **expect** the testers to open the link sent to them. As they just finished their work we anticipate at least one tester to do this on mobile. After filling out the survey, maybe trying to leave out some slots, they send it and are done. Afterwards it is possible they have a look at the catalog, but as the workflow is very straightforward there is not much room for error.

The actual **outcome** of the experiment was just as expected. Only one tester did the task on his smartphone and was very satisfied. One tester missed a question, while another tried to leave it blank on purpose. Both were positively surprised of the error handling. The most dissatisfied tester was the inexperienced one, but that was due to the overall dislike of surveys.

This workflow has been implemented very satisfactorily.

THIRD TASK

The third task is browsing the catalog for interesting labs and getting ready to import them to the Labsystem. For this the testers are provided the following: *Scenario 3, Role: Instructor*

(Out of protocol the tester was shown the Labsystem briefly) It is the beginning of a new semester. You agreed to take part in the iLab courses the instructor recommended you in Scenario 1. The Labsystem is already setup. Your professor asks you to put together a course for your students.

Go to the exchange platform, inform yourself about courses and ready the download links to import the courses to the labsystem.

In this last scenario we **expect** the testers to visit the catalog and have a look at the available labs. If any name, or description looks promising we expect them to open the lab, log in, and take a look at the feedback given by students. If they like what they see, we expect them to find and copy the link for the import. As the import function

will be implemented by Oliver Scheit the task is done here. Even though this task is also not very difficult we expect no mobile device usage. Additionally, we expect users, when they digress, to look only at the FAQs and afterwards come back to the catalog, or lab details.

The **outcome** of the last task was also quite positive. The rework of the display of survey data was very well received. Overall the testers felt quite welcome and all necessary data was given.

The negative feedback of the experienced tester was mostly implemented and the workflow is now fully and satisfactorily implemented.

SUMMARY

To summarize the workflows again, the system was very positively accepted. The feedback from the users was mostly quick to implement and has already been implemented accordingly. Compared to Julius Polar's exchange platform, we have made great progress and put the platform into a productive state. To fully evaluate the platform it has to be used for a semester and put into the real workflow. Only then will it become clear how effective the system is.

CHAPTER 7

CONCLUSION

In this thesis, we analyzed eLearning workflows with particular focus on content handling. We found some major flaws in content exchange. These flaws have also shown up in the analysis of the iLab workflows. An analysis of the Labsystem lead us to the requirements to answer our reasearch questions, mainly, how to improve content exchange in eLearning.

A comparison to related work showed that no popular system sufficiently implements our requirements. Some LCMS do support the exchange of standardized files, but no platform provides versioning. Also gathering student feedback and displaying metadata from the corresponding eLearning platform is hardly ever supported.

Therefore we developed our own system. We analyzed a prototype, which was designed to tackle the same problem. This prototype was built by Julius Polar in 2017 as part of his masters thesis. During this analysis we discovered major flaws in some parts of the system. Backend and Database were both fine, but the frontend had to be completely redeveloped. Our new implementation was build with further development in mind by using the well known framework Angular and building the system with a modular structure.

We evaluated this new implementation in two steps. The first was evaluating according to the prototype and the second step was by having testers doing walkthroughs on given tasks.

The new implementation surpassed the prototype in nearly all aspects. It also implemented almost all of Polars future work and improved everything the prototype got negative feedback on. At the end of this thesis, we evaluated the usability of the system by having testusers doing walkthroughs on different tasks. To gather diverse opinions

CHAPTER 7: CONCLUSION

the testers were of different technical affinity. The users gave very positive feedback, but the system shall be completely evaluated in a real scenario. This complete evaluation is out of the time scope of this bachelors thesis.

In summary, we developed a productive system to complement the Labsystem. It shall mainly be used to improve the workflow of the iLab courses, but it also provides a general eLearning exchange platform for other courses.

CHAPTER 8

FUTURE WORK

Initially this thesis aimed to build upon the prototype. As the prototype had to be completely re-implemented, the estimated time got too short for a bachelor thesis to implement everything planned. Consequently this thesis provides some concrete ideas of future work. Since the front end is built with a high degree of modularity, further development is simplified.

The most urgent next step is to integrate the distribution of survey links into the Labsystem. This would provide a much more convenient workflow. Also the unused tokens have to be removed by hand as of now. Simultaneously with the distribution, the tokens could be given an expiration date on which they are deleted from the database.

Julius Polar mentioned branching of labs, which was also discussed in this thesis. Consequently, it is mentioned here, but it is not recommended to implement it without a reliable concept preventing the loss of feedback data. For now, the workflow of downloading a lab and uploading it with a different name is sufficient.

A few days before the hand in an issue with the theme was discovered. After reloading the page it is switched back to default. This can be fixed by listening to the theme switch event, storing the selected theme and on page reload reset the theme at the top level component. The nebular documentation should provide information to help with this issue [24].

Finally, a further idea would be to offer grouped courses on the platform. This way instructors can just import one package instead of downloading many different courses. Additionally the tag filter could be implemented again, with one group covering one, or more, tags.

CHAPTER A

SOURCE CODE

The sourcecode for the frontend can be found at:

https://github.com/Vallenstein/exchange_platform_frontend

The readme file provides information on how to set it up and how to work with the exchange platform.

The sourcecode for the database can be found at:

https://github.com/Vallenstein/exchange_platform_database

Some information can be found in the readme, but all necessary information is found in the repository of the frontend.

CHAPTER B

USER WALKTHROUGHS

This is the protocol of the walkthroughs for chapter 5:

General Information about the protocol:

A text without anything before indicates a task -This indicates users answers and thoughts

*Indicates actions

*(Indicates answer to the tester)

Workflow Walkthrough I, 30.10.2018

Can you please state your name and profession for the protocol.

-Jürgen Fleischmann, 57 Business Consultant

Do you agree that your data can be included in this Bachelor thesis?

-Yes

To evaluate the implementation of my bachelors thesis, I would like to go through three workflows with you. Before every task we will present you with a small scenario.

You will get to know the system with learning by doing, I will not explain any details, just your setting and will give you only the minimal information. The goal is to evaluate usability of the system. Thus I would like to think loud every thought.

After each task I will ask you what you liked, disliked and will ask your for an overall rating.

Scenario 1: Role: Instructor

You have just been on a conference about the Internet of Things. A colleague of yours came up to you and recommended you an eLearning course about the Basics of the

CHAPTER B: USER WALKTHROUGHS

Internet.

He gave you a link to a platform where they exchange their courses. As you are an instructor at a university you are interested and want to inform yourself.

You got the link via email. Feel free to use any device and inform yourself.

Which device would you prefer to use?

-I prefer a computer to do such tasks. Bigger display as well as easier handling with mouse and keyboard. Mostly Tablet and Notebook. I just like the usability more, easier working environment.

*Opening the Website

-Okay, so.. I have a short look at the entry text, click on next quickly as the first text often provides only little information.

*Goes to second quickly, only peers at first one.

*Reads all info

-Hmm okay I expected more information, and something happening after the last next. Feels a little like a dead end.

*The user did not see the graph below, he is made aware of it

*zooms out a little

-Okay, little information down here, would be seen if not a 13" laptop maybe.. But gives the starting page a nice touch.

What did you like about the sytem?

-So we had a slick design, nice overviews. I like the idea of having themes for informaticians. They often work in the more gloomy. Also quite easy handling. I feel welcomed

What did you not like about the sytem?

-So the only think that really stood out to me was being stuck after this last "Next". As Instructor I would not know what to do

How would you overall rate your experience on this task? (1 best, 5 worst)

1-2

Scenario 2:

Role: Student

(Out of protocol the tester was shortly informed about the iLab workflow) You just finished your lab. After this long day of work you are reminded at the end to have a look at your emails and fill out the feedback survey. Again, you just finished 6 hours of intense work with your teammate.

You got the link via email. Feel free to use any device and fill out the feedback survey.

Which device would you prefer to use?

-The lab is done with a computer right?

*(Yes)

-Okay then I would just do it here quickly.

*Opens Survey

-Okay so I just fill it all out..

*starts to fill in Survey

-Hmm can I just not answer something?

*error

-Okay you got me

*Survey send

What did you like about the sytem?

-Question verification was good

What did you not like about the sytem?

-Nothing, there was not much to it.

How would you overall rate your experience on this task? (1 best, 5 worst)

1

Scenario 3:

Role: Instructor

(Out of protocol the Labsystem was briefly explained the tester)

It is the beginning of a new semester. You agreed to take part in the iLab courses the instructor recommended you in Scenario 1. The Labsystem is already setup.

Your professor asks you to put together a course for your students.

Go to the exchange platform, inform yourself about courses and get ready to import them to the labsystem. As you know the system you would use your computer.

*Opens Platform

*Opens Catalog without thinking or searching

-Okay so when I look at the labs the cursor does not work that bugs me

*Clicks lab, login screen opens

*Clicks Login

*Login Data entered

*Opens Lab

-Hmm okay.. general infos.. some graphs about the survey.. good, where do I get this..?

CHAPTER B: USER WALKTHROUGS

Ahh here; One could think about having the download further up, but okay. So I would just take this and import, okay easy.

What did you like about the sytem?

-Easy Overview in Catalog, also I would be interested in Time to effort, so that's nice. I also like the colorful graphs. Does not look that boring.

What did you not like about the sytem?

-Some of the boxes on the labpage have different sizes. Nothing huge, but it's something.

How would you overall rate your experience on this task? (1 best, 5 worst)

2

Workflow Walkthrough II, 31.10.2018

Can you please state your name and profession for the protocol.

-Luisa Fauth, 20, Law-Student

Do you agree that your data can be included in this Bachelor thesis?

Yes

To evaluate the implementation of my bachelors thesis, I would like to go through three workflows with you. Before every task we will present you with a small scenario.

You will get to know the system with learning by doing, I will not explain any details, just your setting and will give you only the minimal information. The goal is to evaluate usability of the system. Thus I would like to think loud every thought.

After each task I will ask you what you liked, disliked and will ask your for an overall rating.

Scenario 1:

Role: Instructor

You have just been on a conference about the Internet of Things. A colleague of yours came up to you and recommended you an eLearning course about the Basics of the Internet.

He gave you a link to a platform where they exchange their courses. As you are an instructor at a university you are interested and want to inform yourself.

You got the link via email. Feel free to use any device and inform yourself.

Which device would you prefer to use?

-I would first take a short peek at it with my phone, but would switch to my laptop quickly.

*Opening site

*Reading text

-Okay, a nice platform, won some awards.. next

*Looks at table

-Confused about colours and iLab1, 2 and x

-I am confused by the table.. ahh I got it, funny I can change the graph

-Okay the table is cool, but I don't get everything

*Reads next at the top

*Not really satisfied with the wording (which is still subject to change)

-Okay so I'd click on next?

*(just keep going what you want to do)

*Reads next two paragraphs

CHAPTER B: USER WALKTHROUGHS

-Okay if i'm interested i click on visit us here

What did you like about the sytem?

Nice Overview, maybe little plain; Nice with the next buttons, very informative; The graph, even though I don't understand it completely; (after short explanation, which will be added:) I like the seeing the progress of tha iLab

What did you not like about the sytem?

The wording; Either a little dark, or much white

How would you overall rate your experience on this task? (1 best, 5 worst)

2,5

Scenario 2:

Role: Student

(Out of protocol the tester was shortly informed about the iLab workflow) You have just finished your lab. After this long day of work you are reminded at the end to have a look at your emails and fill out the feedback survey. Again, you just finished 6 hours of intense work with your teammate.

You got the link via email. Feel free to use any device and fill out the feedback survey.

Which device would you prefer to use?

As I am already using a computer I would fill it out there.

*Opening System

-Puh not really fan of surveys

*fills out survey completely

*does not fill out rating

-Okay submit, please answer question five.. I don't want to, but okay I guess

What did you like about the sytem?

Nothing special, but overall okay

What did you not like about the sytem?

Having to answer all questions

How would you overall rate your experience on this task? (1 best, 5 worst)

2,5 because I did not like to answer all questions

Scenario 3:

Role: Instructor

(Out of protocol the Labsystem was briefly explained the tester)

It is the beginning of a new semester. You agreed to take part in the iLab courses the instructor recommended you in Scenario 1. The Labsystem is already setup.

Your professor asks you to put together a course for your students.

Go to the exchange platform, inform yourself about courses and get ready to import them to the labsystem. As you know the system you would use your computer.

*Opening website

*Clicks home

-Oh I already am at "Home", okay lab catalog

-Uhh that's a lot of labs.. confusing

*Clicks Lab

-Okay.. whats this? Login.. do I do this?

Yes

*Logs in

*Chooses Course

-Oh that's cool! If I had to choose a course I would look for how students rated that

-I'd choose a course with feedback and level of difficulty

What did you like about the sytem?

Charts

What did you not like about the sytem?

Catalog a little too much

How would you overall rate your experience on this task? (1 best, 5 worst)

1-2

CHAPTER B: USER WALKTHROUGHS

Workflow Walkthrough II, 03.11.2018

Can you please state your name and profession for the protocol.

Felix Maurer, 21, Computer Science Student

Do you agree that your data can be included in this Bachelor thesis?

Yes

To evaluate the implementation of my bachelors thesis, I would like to go through three workflows with you. Before every task i will present you with a small scenario. You will get to know the system with learning by doing, I will not explain any details, just your setting and will give you only the minimal information. The goal is to evaluate usability of the system. Thus I would like to think loud every thought. After each task I will ask you what you liked, disliked and will ask your for an overall rating.

Scenario 1: Role: Instructor

You have just been on a conference about the Internet of Things. A colleague of yours came up to you and recommended you an eLearning course about the Basics of the Internet.

He gave you a link to a platform where they exchange their courses. As you are an instructor at a university you are interested and want to inform yourself.

You got the link via email. Feel free to use any device and inform yourself.

Which device would you prefer to use?

- As there is probably a lot of information I will use my laptop for this

*Opens website on laptop

-I was not told much (probably) so I read the text. Some grammar errors

*(Wording will be worked over)

-Okay now I know what and who..

-Next I would take a look what this platform contains, as I've seen a link to the catalog.

-I've seen the catalog, so I'll visit this

-Okay I click a course and I have to log in

-It says to contact somebody working with the system, so I do is

What did you like about the sytem?

Fast so no waiting times, slick overview, I like the stepwise information

What did you not like about the sytem?

Text is a little small, Little confused about the count up, but that's just me. I don't know I can click to a lab, the curser stays a cursor, not click cursor

How would you overall rate your experience on this task? (1 best, 5 worst)

2

Scenario 2:

Role: Student

(Out of protocol the tester was shortly informed about the iLab workflow) You have just finished your lab. After this long day of work you are reminded at the end to have a look at your emails and fill out the feedback survey. Again, you just finished 6 hours of intense work with your teammate.

You got the link via email. Feel free to use any device and fill out the feedback survey.

Which device would you prefer to use?

-I would like to go home, so I'll do it on my Smartphone

*opens platform on mobile

*reads text, fills out information

-Ah okay I've skipped a question, damn

-Okay done

What did you like about the sytem?

I liked that there were not too many questions, design perfectly fits my screen, I like having a scale behind the questions, I like having the information the data is anonymized

What did you not like about the sytem?

Did not see the "hours" in the survey

How would you overall rate your experience on this task? (1 best, 5 worst)

1

Scenario 3:

Role: Instructor

(Out of protocol the Labsystem was briefly explained the tester)

It is the beginning of a new semester. You agreed to take part in the iLab courses the instructor recommended you in Scenario 1. The Labsystem is already setup.

Your professor asks you to put together a course for your students.

Go to the exchange platform, inform yourself about courses and get ready to import them to the labsystem. As you know the system you would use your computer.

-As I need to "download" something and use multiple systems I'll use my laptop *Opens platform

-As I need to look for something I'll visit the catalogue

-I like UDP, still miss the click cursor

*Clicks UDP lab

-I am not logged in, but as we are taking part I just log in with GitLab

-Ah nice, already filled out

-Oh I'm back at the landing page, so I go to catalog, UDP

-Ah I see the first page of the lab

-Okay now I got the link, I copy it and put it in the Labsystem?

*(yes)

-I would like to have some information like "copy link for labsystem"

What did you like about the sytem?

Preview, the graphs look nice, go back button

What did you not like about the sytem?

Still missing the cursor, LogIn takes me back to Landing, not catalog, Link for importing is too far down

How would you overall rate your experience on this task? (1 best, 5 worst)

2

BIBLIOGRAPHY

- [1] Firouz Anaraki. “Developing an effective and efficient elearning platform”. In: *International Journal of the computer, the internet and management* 12.2 (2004), pp. 57–63.
- [2] Bruce E Bargmeyer and Daniel W Gillman. “Metadata standards and metadata registries: An overview”. In: *International Conference on Establishment Surveys II, Buffalo, New York*. 2000.
- [3] Zane L Berge. “The role of the online instructor/facilitator”. In: *Educational technology* 35.1 (1995), pp. 22–30.
- [4] Blackboard. *SourceSites*. URL: <https://preview-learn.blackboard.com> (visited on 10/20/2018).
- [5] Hudson Borges et al. “On the popularity of GitHub applications: A preliminary note”. In: *arXiv preprint arXiv:1507.00604* (2015).
- [6] GitLab B.V. *GitLab API / GitLab Docs*. URL: <https://docs.gitlab.com/ce/api/> (visited on 10/26/2018).
- [7] Stephen Cass. “The 2015 top ten programming languages”. In: *IEEE Spectrum*, July 20 (2015).
- [8] Class Central. *115 Open Education Online Courses*. URL: <https://www.class-central.com/provider/open-education> (visited on 11/01/2018).
- [9] Coursera. *Coursera / Online Courses & Credentials by Top Educators*. 2018. URL: <https://www.coursera.org> (visited on 10/24/2018).
- [10] Jenny V Freeman, Stephen J Walters, and Michael J Campbell. *How to display data*. Vol. 26. John Wiley & Sons, 2011.
- [11] D Randy Garrison. *E-learning in the 21st century: A framework for research and practice*. Routledge, 2011.
- [12] Cynthia Gautreau. “Motivational factors affecting the integration of a learning management system by faculty.” In: *Journal of Educators Online* 8.1 (2011), n1.
- [13] Victor Gonzalez-Barbone and Luis Anido-Rifon. “From SCORM to Common Cartridge: A step forward”. In: *Computers & Education* 54.1 (2010), pp. 88–102.

- [14] Thavamalar Govindasamy. “Successful implementation of e-learning: Pedagogical considerations”. In: *The internet and higher education* 4.3-4 (2001), pp. 287–299.
- [15] Google Inc. *AngularJS - HTML enhanced for web apps!* 2010. URL: <https://github.com/angular/angular.js> (visited on 09/16/2018).
- [16] Google Inc. *One framework. Mobile & desktop.* 2016. URL: <https://github.com/angular/angular> (visited on 09/16/2018).
- [17] Google Inc. *One framework. Mobile & desktop.* 2016. URL: <https://angular.io> (visited on 09/28/2018).
- [18] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz*. URL: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/itgrundschutz_node.html (visited on 11/06/2018).
- [19] Johan Ismail. “The design of an e-learning system: Beyond the hype”. In: *The internet and higher education* 4.3-4 (2001), pp. 329–336.
- [20] Michael Kaplowitz, Timothy D Hadlock, and Ralph Levine. “A Comparison of Web and Mail Survey Response Rates”. In: 68 (Apr. 2004).
- [21] Janet Leckey and Neville Neill. “Quantifying quality: the importance of student feedback”. In: *Quality in Higher Education* 7.1 (2001), pp. 19–32.
- [22] Akveo LLC. *Admin dashboard template based on Angular 6+, Bootstrap 4.* 2016. URL: <https://github.com/akveo/ngx-admin/tree/starter-kit> (visited on 10/14/2018).
- [23] Akveo LLC. *Angular 6 Components, Auth and Security.* 2016. URL: <https://github.com/akveo/nebular> (visited on 10/14/2018).
- [24] Akveo LLC. *Nebular Documentation v2.0.1.* 2016. URL: <https://akveo.github.io/nebular/docs> (visited on 10/20/2018).
- [25] Jon Loeliger and Matthew McCullough. *Version Control with Git: Powerful tools and techniques for collaborative software development.* " O’Reilly Media, Inc.", 2012.
- [26] Jakob Nielsen. “Usability inspection methods”. In: *Conference companion on Human factors in computing systems*. ACM. 1994, pp. 413–414.
- [27] Marc-Oliver Pahl. *Labsystem.* 2010. URL: <https://github.com/m-o-p/labsystem> (visited on 10/27/2018).
- [28] Laura Pappano. “The Year of the MOOC”. In: *The New York Times* 2.12 (2012), p. 2012.
- [29] Pivotal. *Spring.* URL: <https://spring.io> (visited on 11/01/2018).
- [30] Julius Polar. “iLab Exchange Platform”. In: (Sept. 2017).
- [31] John TE Richardson. “Instruments for obtaining student feedback: A review of the literature”. In: *Assessment & evaluation in higher education* 30.4 (2005), pp. 387–415.

- [32] Douglas C Schmidt and Zach McCormick. “Producing and delivering a course-era MOOC on pattern-oriented software architecture for concurrent and networked software”. In: *Proceedings of the 2013 companion publication for conference on Systems, programming, & applications: software for humanity*. ACM. 2013, pp. 167–176.
- [33] Matthias Schonlau and Ellen Peters. “Graph comprehension: An experiment in displaying data as bar charts, pie charts and tables with and without the gratuitous 3rd dimension”. In: *Social Science Research Network Working Paper Series* (2008), pp. 1–16.
- [34] Pei-Chen Sun et al. “What drives a successful e-Learning? An empirical investigation of the critical factors influencing learner satisfaction”. In: *Computers & education* 50.4 (2008), pp. 1183–1202.
- [35] J. Bradley W. Denniss. *The OAuth 2.0 Authorization Framework*. RFC 8252. Internet Engineering Task Force (IETF), 2017. URL: <https://tools.ietf.org/html/rfc8252>.
- [36] Elizabeth T Welsh et al. “E-learning: emerging uses, empirical results and future directions”. In: *international Journal of Training and Development* 7.4 (2003), pp. 245–258.