RWTH AACHEN UNIVERSITY

FabCenter

Webapplication to support users and administrators of FabLabs with creating and sharing documentation.

Diploma Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Tim Hemig

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schröder

Registration date: 30.10.2012
Submission date: 11.06.2013

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, Juni 2013
Tim Hemig

Contents

List of Figures

List of Tables

Abstract

With the world wide growing interest in Personal Fabrication like 3D-printing at home or accessible laser cutting in the neighbourhood, people of all kinds of profession are able to create products on their own, and to get inspiration, help or synergy effects out of environments addressing this field. The Internet and modern methods of communication and documentation raise this field to a worldwide omnipresent phenomenon. Fablabs as defined by Neil Gershenfeld, local makerscenes, decentralized hacker clubs like dorkbot, or the traditional DIY-hobbyist at home can form or use the knowledge base of such communities.

Since feedback, comments and documentation from reports of trial and error or the opinion of experts are the fundamental ingredients to create a knowledge base, this work examines the possibilities to raise the grade of documentation concerning a single product design in the field of mainstream personal fabrication. Often the user produces a loose documentation, but the gained insights never find their way back into the community.

We will discuss and extract the basic requirements of a system to support the full cycle of a minimal documentation-step, which is used to ease or even enforce users to document the current step or their design-development. Subsequently we will show the implementation of FabCenter, a web-based service that guides the users and administrators of a fablab environment through their making-experience and provides a way to grant a minimal level of documentation of this process.

Finally a user study shows that FabCenter meets the requirements and users feel guided and persuaded to feed their experience back into the community that is supporting this development in the first place.

Überblick

Mit dem weltweit wachsenden Interesse an Personal Fabrication wie zum Beispiel das 3D-Drucken und zuschneiden von Material mittels Lasercuttern in der erweiterten Nachbarschaft werden Menschen aus allen Fachgebieten und Lebensräumen befähigt eigene Produkte selbst zu erschaffen, und die nötige Inspiration, das Fachwissen und Erfahrung aus Umgebungen zu ziehen die sich mit diesem Thema beschäftigen. Das Internet und moderne Kommunikations- und Dokumentationsmethoden heben diese Erscheinung zu einem weltweiten Phänomen. Fab Labs, wie sie Neil Gershenfeld begründete, lokale Bastler-Szenen, dezentrale Hacker-Treffen wie Dorkbot-Gruppen, oder der traditionelle Heimwerker daheim können die Wissensbasis solcher Gemeinschaften erweitern und nutzen.

Weil die dokumentierten Erzeugnisse aus Ausprobieren, Testen und Expertenmeinungen die Grundlage einer solcher Wissensbasis bilden, untersucht diese Arbeit die Möglichkeiten, die Prozess-Qualität der Dokumentation eines einzelnen Produktdesigns im Kontext der massentauglichen Personal Fabrication zu verbessern.

Oft wird nur lose dokumentiert und der erzeugte Erfahrungsschatz findet nicht seinen Weg zurück in die Gemeinschaft.

Zunächst werden die grundlegenden Anforderungen für ein System zur Unterstützung eines vollständigen, minimalen Dokumentations-Zykluses beschrieben und aufgestellt. Dieses System hat den Anspruch den Benutzer dazu zu motivieren, den aktuellen Schritt in der Design-Entwicklung zu dokumentieren.

Anschließend wird die Implementation von FabCenter als Web-basierter Service vorgestellt, der den Benutzer und Administratoren von Fablabs in den Besucher-Abläufen unterstützt, und eine Verbesserung der Dokumentation liefert.

Zuletzt zeigt die durchgeführte Benutzerstudie, dass FabCenter die aufgestellten Anforderungen erfüllt, und den Benutzer ermutigt die Dokumentation ihrer Projekte durchzuführen.

Acknowledgements

First of all I want to thank my supervisor Dipl.-Inform. René Bohne for his valuable guidance through the process of writing this thesis.
Also I want to thank Prof. Dr. Jan Borchers for being my adviser and providing me with the chance to write my thesis at his chair. The insights and feedback he and the members of the chair were able to give me, were very worthwhile.
Additionally, I want to thank Prof. Dr. Ulrik Schröder for being second examiner.
I want to thank all participants in my user studies and questionnaires.
Next to all my friends and fellow students at the RWTH Aachen, I want to thank all my coworkers and superior for being flexible and giving me the space to combine work and university.
Last but not least I have to express my deepest gratitude to my parents, who gave me all the freedom and support to study computer science and follow my interests.
Thank you everybody!
Tim

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Source code and implementation symbols are written in typewriter-style text.

`myClass`

The whole thesis is written in American English.

Wherever the masculine form is used, it applies to the feminine form as well.

Chapter 1

Introduction

1.1 Background

With the movement of Personal Fabrication and maker-scenes becoming more and more a publicly visible phenomenon, many of new members are on the brick to join this field of Do-It-Yourself (DIY) and open hardware communities. The Internet presented with the Web 2.0 revolution a convenient way to communicate fast and easy over the span of the whole world. Before this there was of course the Internet and other ways to communicate around the world, but this ways of communication like fax, e-mail or websites never reached the huge user base and popularity to be known to everybody in day-to-day life.

Many small groups or hobbyists at home formed not necessarily small, but limited islands of communities that swap information about DIY. Magazines with a broad channel and wide reach may distribute information, but the single user at home would be only a consumer of that information, and only in rare instances he would be a producer.

For the currently evolving maker scene for the broad population it is important that everybody can contribute to each others projects. Everyone may be an expert on a narrow field of personal fabrication, or may have made similar experiences that would help another maker.

But not everybody is also a writer and able to create exact and robust sets of instruction to recreate a specific product, or is able to put his general knowledge into an article for a scene-magazine or website.

1.2 Personal Fabrication and Fablabs

Personal Fabrication can be explained analog to the phenomenon of Personal Computing. Before the 1980s computers were huge expensive and only owned by big companies or universities. Many people wanted to process data on it and had to queue up and schedule time on the CPU (Central Processing Unit). With the game changing impact of the first personal computers the industry developed bit-by-bit many different and competing computers. The Size and price of computers was reduced by a big factor. At the end of the 1980s computers got affordable by private households or small businesses. They are now called Personal Computers (PC) and the first applications left the scientific- or business-context to the field of private entertainment. This means not only games, but also software to manage the finances of the family, or to write books and stories. More and more people gained access to such machines and the number of use cases for computers grew.

When the Internet got accessible by the broad population in the 1990s, the flow of information broke through. Everybody connected to the Internet was able to find a place of exchange covering the topic of desire. The breakthrough of search engines at the end of the 1990s completed the revolution of personal computing, since now a user only needed to know what he wants to learn about, and was able to find information by anyone who made it accessible to the Internet.

Personal Fabrication took a very similar way, as described by Mota [2011]. In the past only big companies and perhaps universities were able to build and finance machines for mass fabrication of innovative products. Of course a single person could also design and produce own products, but he would not be able to compete or create a successful

business out of it without being rich or the help of investors. For personal purposes this is also too expensive.

With the accelerated flow of information through the Internet and first fabrication devices being affordable to little communities institutions like fablab were able to provide everybody interested in producing and creating with access to fabrication devices. Knowhow, designs and knowledge got exchanged in the digital world such that fablabs are not spread in few colonies, but begin to emerge everywhere world wide. The interest grows steadily. Again the Internet is the place to go to and self-educate in the topic of Personal Fabrication.

Fablabs as coined by Gershenfeld [2005] do not only provide access but also enable users to have own fabrication devices by producing the necessary parts in fablabs or similar communities. With that the Fabrication has reached the stage of Personal Fabrication. Anyone is able to gain access and produce what he likes.

A fablab can be a closed or open group of users that usually shares a set of more expensive, but no longer industry-only devices. To gain access to devices like laser cutter, professional 3D-printer, CNC-mill etc., users form communities and share the costs. Either there is an institution having these devices at hand already, and just opens the access to the lab for a period of time to the public, or a closed group of users forms a club in which one has to be a member (mostly by paying a periodically fee to support the maintenance and financing of the devices and rooms etc.) to have access. In general these groups have no financial profit as person or institute at mind, but seek to generate synergy where all members or users may profit. They also want to provide access to devices and perhaps something like a knowledge base of recent and older projects with their documentation. In appendix A the complete fablab charter can be found, which defines to motivation of fablabs (figure A.1).

1.3 Open Hardware

In context of the rising Personal Fabrication and fablabs there is an evolving Open Hardware scene in the Internet and fablab-like communities. Open Hardware describes the publicly availability of digital (hardware) designs under free licenses. These licenses are mostly analog to the models of licensing known from the Open Source software movement. Open Hardware is a motor of innovation in the field of Personal Fabrication like Open Source software did it for personal computers. Free software is free to use and enables more users to use it without constraining barriers like commercial licenses. Some Open Source licenses also are matched to the context of Open Hardware Under certain circumstances everyone is free to use and evolve designs by others. An exemplary license (CERN Open Hardware License) is attached in appendix B. With this kind of knowledge transfer the broad population gains access to digital design that can be user for Personal Fabrication or contribution in terms of evolving the design.

1.4 Thesis Overview

In chapter one we gave an overview about the problems of mass-developing open hardware designs which is the basis and motivation of this thesis. The following chapter will focus on the principles of several open hardware or fablab supporting platforms and related scientific work. From the examples we will gather some conditions for the objective of this thesis. In chapter three we develop the requirements for a documentation supporting system and involve the conditions defined before. We will describe the development process of FabCenter with its essential functions and technologies. In chapter four the concept is evaluated with the help user studies and we determine if the requirements were met. The last chapter five summarizes the whole work and its contributions. It will also show how the system may be improved in future work.

Chapter 2

Related work

In this chapter we compare existing systems or concepts, which partially deal with our wanted process to enable users to document in an easy way.

2.1 SHARE

The SHARE project Toye et al. [1993] at Stanford University picks up the concept of the knowledge sharing by Vannevar Bush in his article "As We May Think" Bush [1945].

In general the concept is to manage the overwhelming amount of experience, knowledge, research-questions and -results. Back in 1945 science emerged to be a field of strong specialization. It got more and more hard to be an expert on multiple growing fields of research. Forcing scientists into digging deeper into more narrow subjects, the exchange and documentation become a necessary need.

SHARE is a prototyped system that is aiming at supporting engineers or designers in collaborating over networked and computerized infrastructure that fulfills the vision by Bush in Bush [1945] By defining two generic templates for documents all information gathered by coworkers an be stored in a digital notebook, providing an easy to navigate through information base. Al participants work on custom

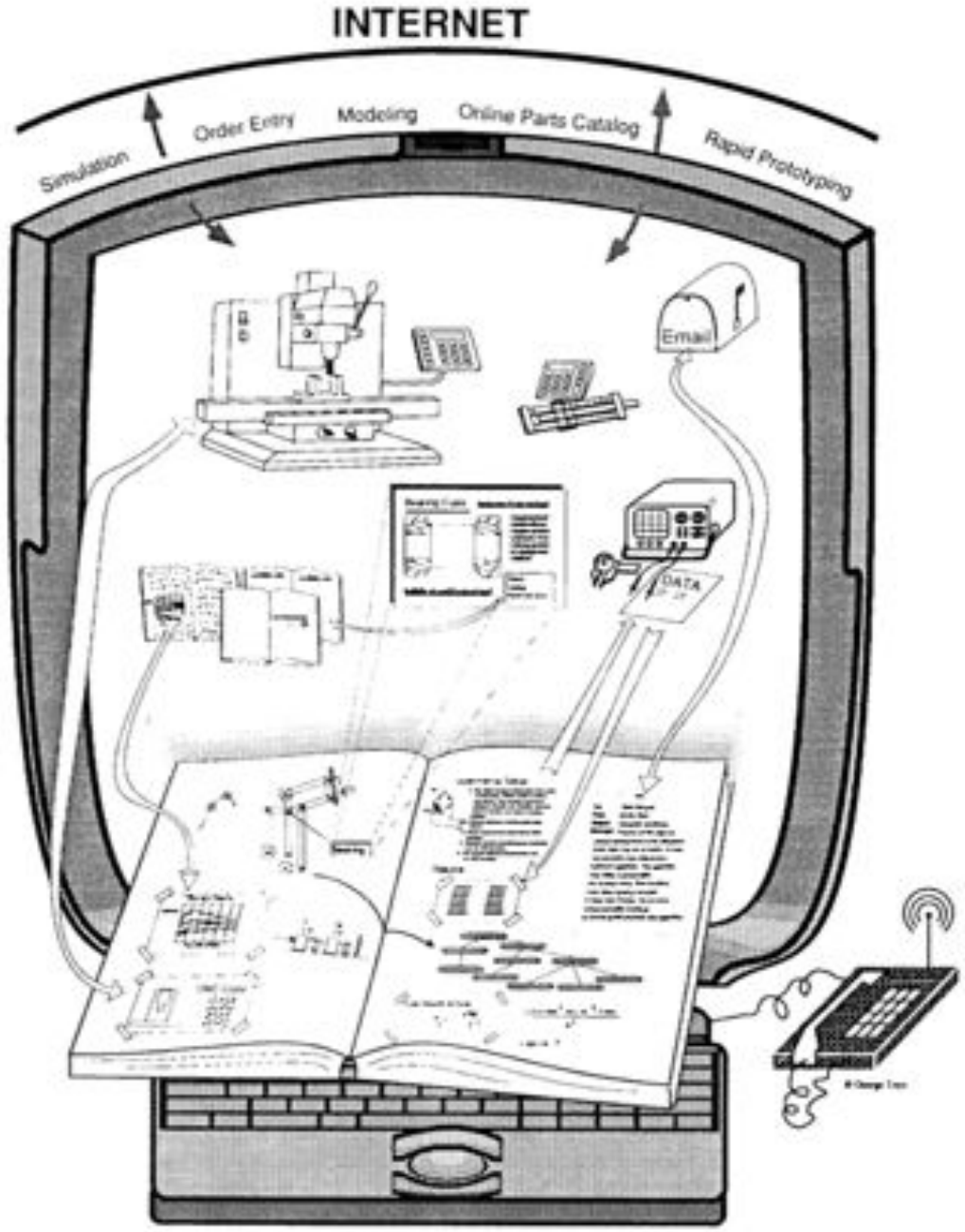


Figure 2.1: Sketch of the concept of SHARE. Exchanging and sharing of notes and other documentation of knowledge via networked computer applications

adapted client-software to the central document- and mail-server in the system hosting all the aggregated data. Re-thinking of processes or investigating topics double times can be avoided, and the efficiency of a team can be raised to higher levels.

2.1.1 Conclusion of SHARE

This project including all the inspiring work underlines the need and importance of sharing and adding knowledge via highly networked and fast communicating technologies. Todays modern Internet is the most multimedia based and rapidly evolving communication medium.

The route for basically helping out the Open Hardware community is clearly defined by supporting the sharing of even the most little pieces of feedback and gained insights with the information seeking makers and designers. Help is needed in terms of having a good base of documentation on the fast growing number of small and big projects and designs.

Considering that most documentation consists of a series of photos and describing text, our system should provide a way to feed photos and a short description of them back into the project.

2.2 Online Documentation & Sharing Tools

2.2.1 FabML & FabMoments

Developing open & distributed tools for fablab project documentation

FabML, concept for uniform sharing of project data.

Määttä and Troxler [2011] describes a unified document format FabML as a machine readable format It is an exten-

sion of the Extensible Markup Language (XML)[1] . XML follows tree-like structured tags, which may follow a separate defined set of rules arranging the nesting of tags and their properties. This set of rules grants any reading party some minimal structure to expect. All data in paths that are not defined by the rules may be ignored, but do not destroy the core-information that follows the rules. This format supports the aspect of sharing between several user groups maintaining their own local system (the islands mentioned in section 1.3). Every local system would define a list of public projects and feed this list in the described format to others via the Internet. Since there are already prototypes of this data source present (i.e. fablabs in Nuernberg and Amsterdam), any new resources of project-data can be used instantly. Also, adapting to this format seems to be more desirable the more information can be accessed.

FabML is a concept of basic project information

This format is an feature that should be supported to connect not only groups using the new system, but also the ones using the current prototypes or own systems without any changes. Of course the format would probably limit the possibilities in representing the single project but there should be enough space for extensions to transmit a goof project description and guide the user to the full content if needed.

Fabmoments (Prototype using FabML)

As mentioned, there are some prototypes using the idea of the FabML format to exchange data about projects between user groups. In figure 2.2 an example of a used aggregation of different sources is shown.

Fabmoments as concrete implementation of FabML

There are three labs aggregating their information (FAU Nuernberg[2] , FabLab Utrecht[3] , FabLab-Leuven[4]) through exchangeing RSS-feeds (Rich Site Summary[5]). RSS itself is also based on XML, so the technology is quite usable and

[1] http://www.w3.org/XML/
[2] http://fablab.fau.de/project
[3] http://protospace.nl/fabmoments
[4] http://fablab-leuven.be/?q=aggregated-fabmoments
[5] http://www.rssboard.org/rss-specification

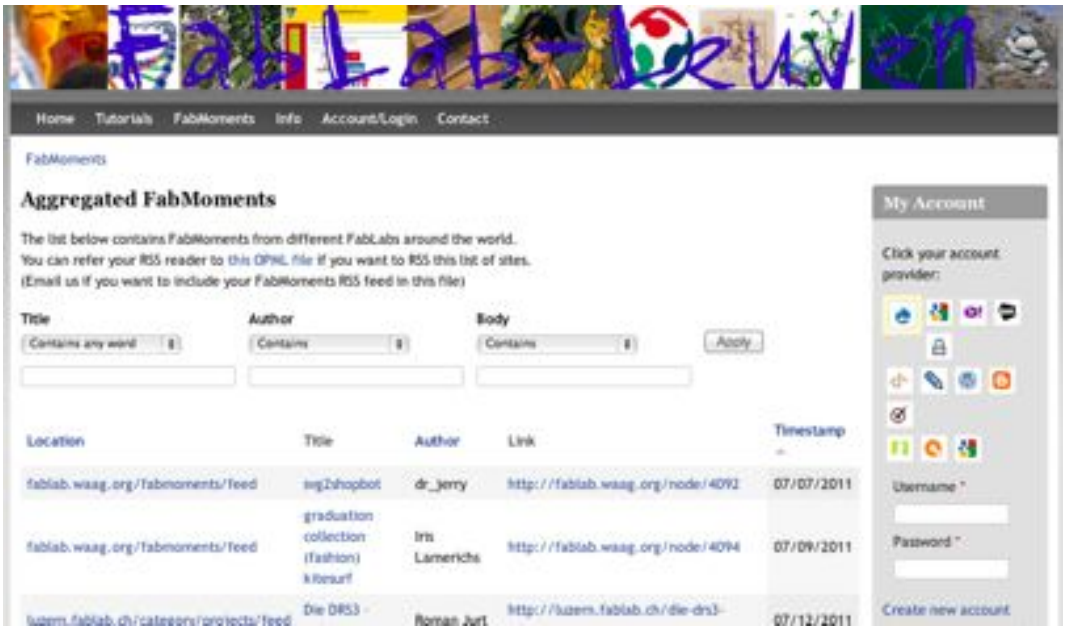


Figure 2.2: Aggregated list of Fabmoments on the webpage of fablab Leuven[a] ready to be filtered.

[a] http://fablab-leuven.be/?q=aggregated-fabmoments

extensible beyond the current state without excluding older systems. Such older systems would just ignore the added information and be able to operate the used way.

Conclusion of FabML & FabMoments

This example shows that this way of exchange and collaboration is possible. Currently, there are no really large networks of exchange forming. If a group adds a Feature, other Groups may adapt them too, or even improve them. We need a higher critical mass to push this format to a wider used standard.

Fabmoments are working, but not spread wide enough

The concept of servers or services mutually telling each other about the rough details of the projects they know about is a good extension to all kind of automated services in the context of fablabs. Without the opportunity to write feedback into the source, this system does not really provide an fully capable API, although reading is allowed

Fabmoments as a chance to get our system involved by other systems.

by requesting the feed from the public URL (Uniform Resource Locator), of course. We could have the system provide a list of projects connected to a fablab or institution that uses this service.

2.2.2 Fabiji

Figure 2.3: The Fabiji prototype. An iPad enclosed in a kiosk-stand cut out of medium-density fibreboard with a laser cutter

Concept of Fabiji

Fabiji[6] is a prototype of a kiosk system (as shown in figure 2.3) that lets the user easily make photos of his or her latest prototype or creation and was subject of the master thesis by Zhao He in March 2012 at the RWTH Aachen.

The system covers the following six requirements (He [2012]):

- R1: Help users to create simple documentation in a short time non-intrusively.
- R2: Encourage users to create documentation when they are at the fab lab.

[6] http://hci.rwth-aachen.de/fabiji

- R3: Encourage users to explore others projects when they are at the fablab.
- R4: Give users opportunities to meet in real life.
- R5: Help users to take better photographs.
- R6: Easy to deploy and configure for other fablabs.

With the exception of R4 all requirements show that this concept enables every fablab to have the users or visitors of a fablab create visual documentation with append-able text.

There is a local project management included, such that multiple users are able to use it, and manage their multiple projects. As a product the taken photos of the latest created project in the fablab are put into the system and linked to the project.

An Application Programming Interface (API) is defined, but the data is not published in an automated way into the Internet. It is not made available to the public.

The system is not always active and reachable for other services, so currently this information cannot be used to automatically extent the documentation of a project originating outside the system. Projects from the outside cannot be imported in the first place.

Conclusion of Fabiji

Freshly created documentation should be available to the public almost immediately. As soon as this process is stalled, or pushed further into the future, the feedback looses its temporal connection to the experience. It may also never get published since it is forgotten, or held for inaccurate data in relation to the more developed design at this later point in time.

We learned that our system should provide a mechanism to feed back the created data into the source of the project

as soon as possible after creating it. The Fabiji System is a great basis for operation inside a fablab, but is does lack the instant feedback into an online resource. These online resources can be (or are already) indexed by search-engines from other services persistently. Also the import from other sources should be possible. With having and import and export of updates are a condition, we could also drop the local project management at all, and just take care of adding directly to an existing project-storage. Handling the project-data this way we would only need an interface to existing services.

2.2.3 Thingiverse

Concept of Thingiverse

Thingiverse.com[7] is a community platform for publicizing and storing hardware designs. It is offering free accounts for users to enable them to upload their designs.

Thingiverse is a popular publishing platform for digital designs.

The Thingiverse-landscape is very popular[8] in the scene and attracts users from the beginner, which consumes projects by others, to professionals, which create and publish on that platform. It is very centralized and the company running the site offers the user full control over the kind of licence concerning their projects in exchange for a full license to the design for the company itself.

Trough the huge base of users, the similar high count of already published designs, the search function and some social networking enabling voting methods make the platform a very high ranked place to go to. The site attracts users that want to find project templates or even complete designs for their current need. Therefore not only designers and self-planning makers will be attracted by this platform. Also consuming users of the more and more easier to build or buy 3D-printers and laser cutters would gain interest. They may not be able to create own designs, but it became very easy to provide users with parametric design

[7] http://www.thingiverse.com
[8] http://en.wikipedia.org/wiki/Thingiverse

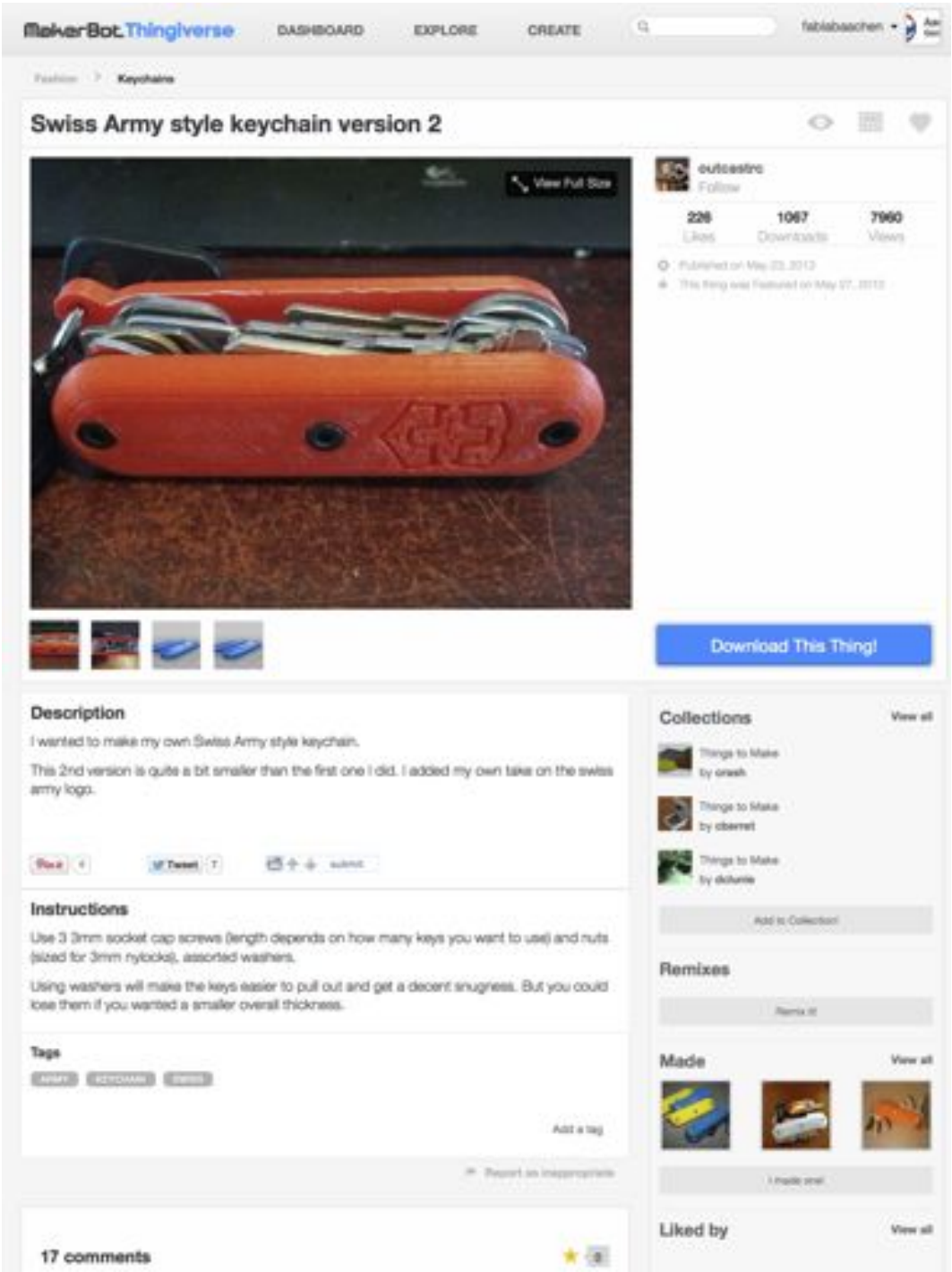


Figure 2.4: Project overview inside of Thingiverse

and enable them to configure their own variation of a more complex design for their own use.

Thingiverse presents each project in its current state consisting of:

- Design files
- Images of prototypes or representations of the design files
- Title and description of the project
- A set of instructions to produce the final product
- list of copies or remixes by other users
- some social networking information as, e.g., "likes" by others

Feedback and documentation through networking.

By linking the copies by others, the feedback is gathered out of the information pool of other users. Even the list of remixes adds to the pool of feedback, since other users may improve the concept of a design or follow another, perhaps better approach. No direct access to the original designer's account is needed to add a copy or remix (although some cases will depend on the license).

Conclusion of Thingiverse

The broad mass of users in the growing open hardware scene are in most cases already on Thingiverse with their projects or will most likely be attracted by this platform early in their phase of getting started in this field because of the many already existing and good indexed projects. Our system shall integrate Thingiverse via API, to have the user bring in all his online projects, or to give him or her a place to store the first designs without having them to upload the file multiple time to different platforms in future. Double existing projects would only complicate the process of giving and gathering feedback and would increase the investment by having two or more places to keep up to date.

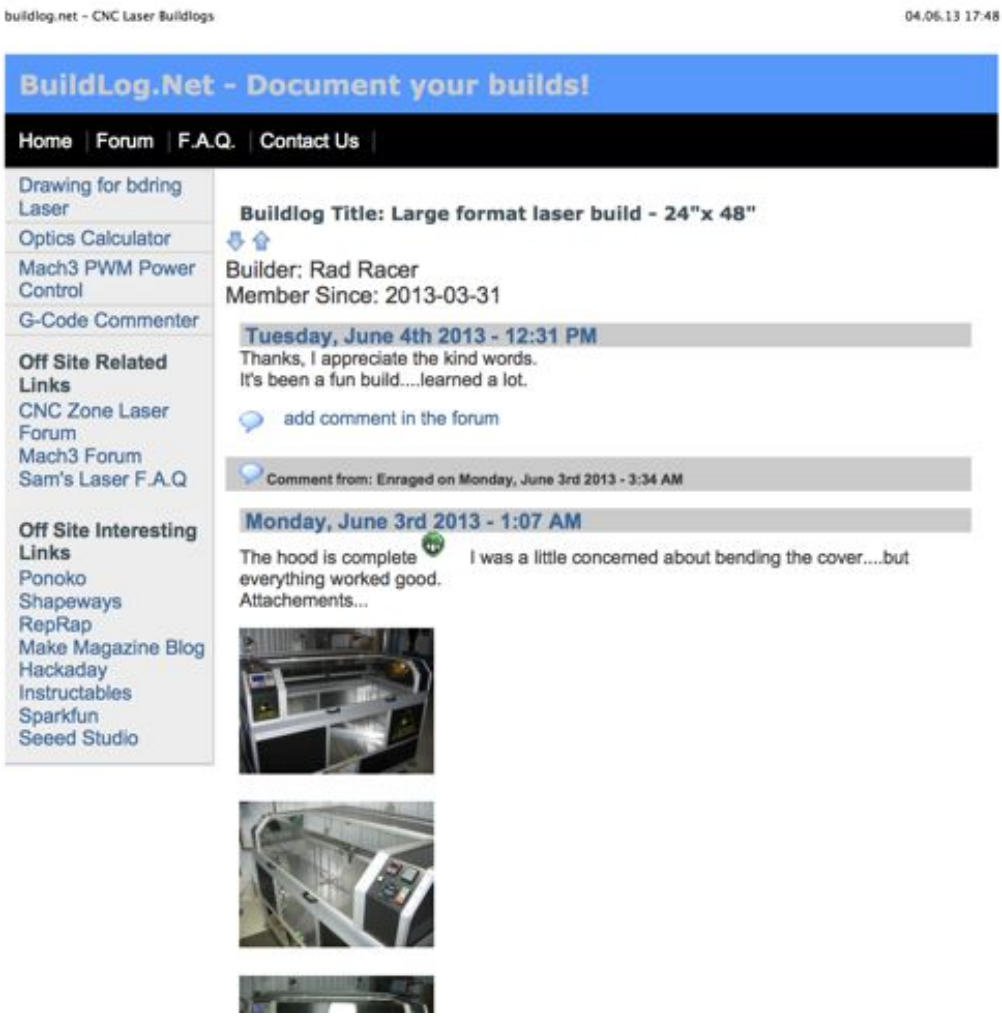


Figure 2.5: BuildLog website in the default project view

The copy-function (the website calls the action "I made one") provides a hook for our system. The user can feed all documentation out of one process during a fablab visit back into thingiverse as a "copy", if he or she is not the owner of the project. Otherwise the information can be added to the description in case of text, and to the image gallery in case of photos.

2.2.4 BuildLog

Concept of BuildLog

This website[9] works similar to thingiverse.com as described before, but is smaller in terms of user base (around 2.700 users according to the website itself) and is connected to a community of CNC-Lasercutter-builders. The difference is the focus on the documentation of the design, not the current state. The development process is shown, other users can contribute to it, and the whole data is presented chronologically reversed like a diary with the newest entry on top (this is reversible after the page loaded). Under the hood the data is edited in an online forum software, and passed into the front end to show it in a different way to the website visitor. A thread in the forum represents each Project. Contributions by the thread starter will be treated as entries into the projects journal. All other entries are comments to the corresponding entry they belong to, if a user replied to a selected entry, or just the latest entry if no reference is there other than the entry belonging to the thread.

System does not support onetime-visitors in the first place.

Approaching users would be able to recapture the whole design process from the first idea to the most current state, and may understand certain decisions during the process by the designer better. But in this case you have to read from the bottom to top or reload the page to get the correct order for recapturing the process. The website intention is clearly to show the often visiting user the latest updates in the topmost entries. If we see the process of a user using this design as a one cycle-interaction of finding the project, downloading it, perhaps adapting it, making it and giving feedback the user maybe does not follow the project necessarily, because the current need could be satisfied.

Conclusion of BuildLog

Documentation should be in chronologically order with latest entry at bottom.

Since the one time (or only a few times) visiting user is

[9] http://www.buildlog.net

more likely in a broad scene, so the idea of adding the documentation to the project itself, if the owner is acting within our system, shall handle the update in chronologically order with the latest action at the end. Therefore the latest information should be added to the bottom of the representation as far as it is supported. In case of text this can easily implemented by adding a passage at the end containing the known details of the visit in the lab and some words commenting this step in development.

BuildLog does not provide an API.

With BuildLog not providing any way of remotely interacting with the system by API we cannot integrate this service into a new system.

Feedback on Thingiverse still possible, if current user does not own the project.

In case of Thingiverse as a target data-pool adding copies from others (not the owner) the feedback is still given, since every approaching user will have a list of existing copies (or slight variations) and remixes (like mutations). This forms a pool of experiences that will help the user to improve the own copy or variation of the design.

2.2.5 Instructables

Concept of Instructables

Similar platform, but wider field of topics in projects.

Instructables.com[10] is also a very similar platform to Thingiverse and BuildLog, but it does not aim at the 3D-printing and laser cutting scene at all, and takes a different approach on presenting the projects and their intention. The users publish step-by-step instructions on topics of the general DIY-field, or arts and crafts. Reading is public, contributing is allowed to everyone registering for free, and owner of a commercial pro account get some comfort-features.

Step-by-step visibility, but restrictions for non-paying customers.

As clearly visible shown by figure 2.6, each step is shown as an own page from where you can get to the next or previous steps, but the users also are able to show all steps at once, or even download a file in the Portable Document Format (PDF), if you are a paying member.

[10] http://www.instructables.com

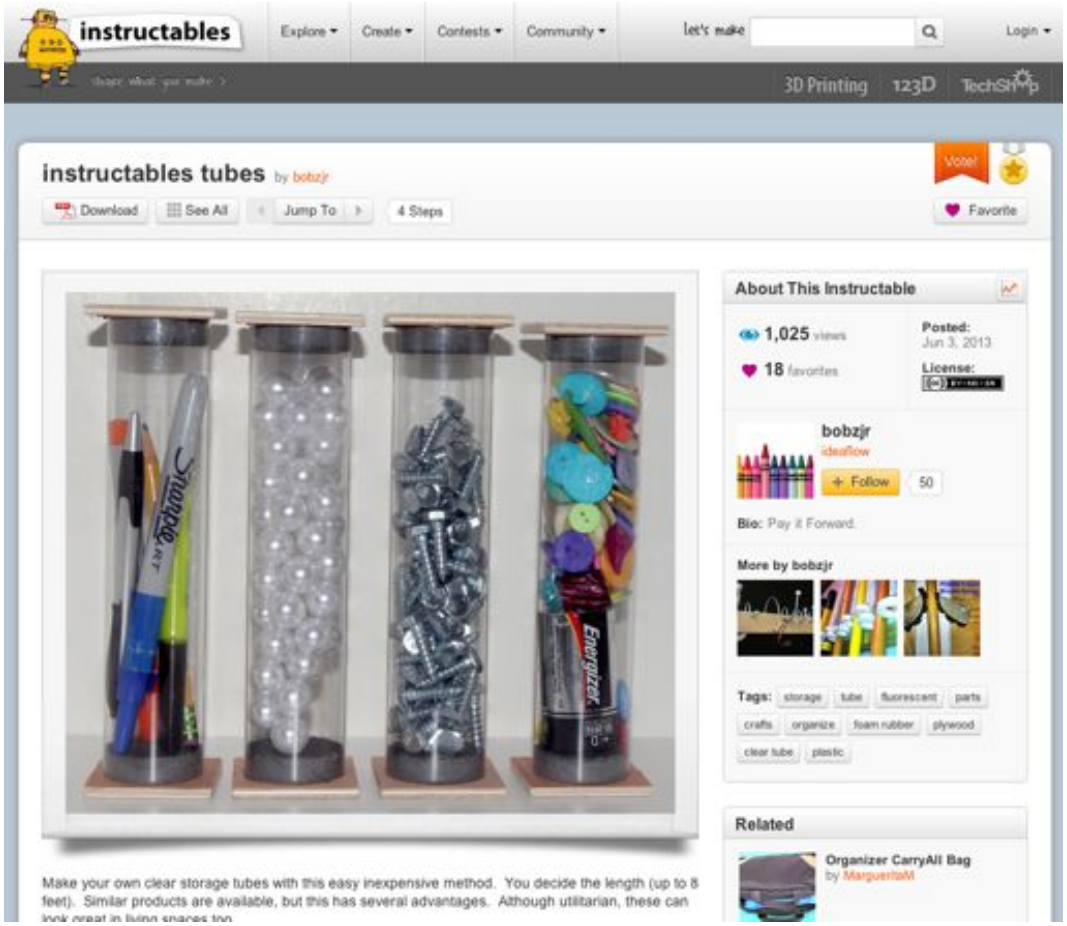


Figure 2.6: Instructables website showing first of four steps int he project view.

Bigger potential user base trough wider topic coverage.

Since there are not necessarily expensive or complex machines involved, even younger children and people not involved in technology as most makers seem to be can join the group the user group in this case. With more projects of a very easy skill level and low entry technology barrier also beginners in the field of DIY are involved.

Conclusion of Instructables

Interesting system, but no API provided.

With a bigger addressable potential user group this platform is an interesting candidate for a project source and documentation-target for our system. But with the lack of an API, there is currently no way to integrate this ser-

vice. The Platform should be observed and perhaps be integrated in a later point in the further development of this project.

Step-by-step instructions lower barriers, but are hard to update without reviewing the whole work.

The step-by-step approach seems to give the user the opportunity to have the instructions broken down into small less challenging operations. In general this is a good feature while presenting the whole project, but a visitor in a fablab already is interested in only a partially operation. Most of the projects in fablabs consist of making or getting a design, improving it, actually making it and reentering the Design-Implementation-Analysis-Cycle (DIA-Cycle). Additionally the Instruction set does not represent the development of the Project in form of a diary. It is a well-formulated text to instruct implementing the design. If the design changes, the instruction set may change at all. Since we want to add information in each prototyping step, we cannot work with this kind of representation in an ideal way. The owner of the project or someone with the necessarily permissions would have to review the whole text each time a piece of documentation would be added. Previously used states of the project, or diary-formats are easier to extent.

2.2.6 Summary of Online Documentation & Sharing Tools

We now have seen different existing systems and can compare them over some features of interest for our system.

In table 2.1 the different systems are compared by their user base, kind of project representation and the existence of an API to communicate with the system. Green cells indicate a property usable by our system, yellow shows only partial usable features and orange marks the features, which are not fulfilled in context of our system. While the project representation in almost all cases are clearly enough to work with, most of the systems lay an API. In the matter of the user base the systems that are most widespread are more preferable, since we want to include as many users as possible.

Following the argument it becomes obvious that we should

System vs. Feature	User Base	Project Representation	API
FabML & Fabmoments	Serveral Fablabs	Simple State	Read only
Fabiji	None (Prototype)	Full State of Project	None
Thingiverse	Mainstream	Full State of Project	Read & Write
BuildLog	Small	Full History of Project	None
Instructables	Mainstream	Steps of Instructions	None

Table 2.1: Online Documentation & Sharing Tools Comparison

integrate Thingiverse's API to use the stored projects of the existing user accounts. With these resources in place we can handle the authorization process and provide a channel to feed back the documentation into the project as described in the conclusion of 2.2.3.

2.3 Online Scheduling Tools

2.3.1 Doodle

Concept of Doodle

Easy and accessible platform for scheduling.

Doodle.com[11] provides a fast and easy way to organize meetings or democratic decisions through a collaborative scheduling system. Users can start a poll that may include the question for a specific date or a solution to a problem out of a range of possibilities. Also you can manage lists of, e.g., attending participants with their comments to an event. This all is possible without registering any account with the service, since the polls are addressed and protected by a secret ID that is only shared with the participants contained in an URL. A second ID is used to administer that poll if needed. With doodle not requiring the user to cre-

[11] http://www.doodle.com

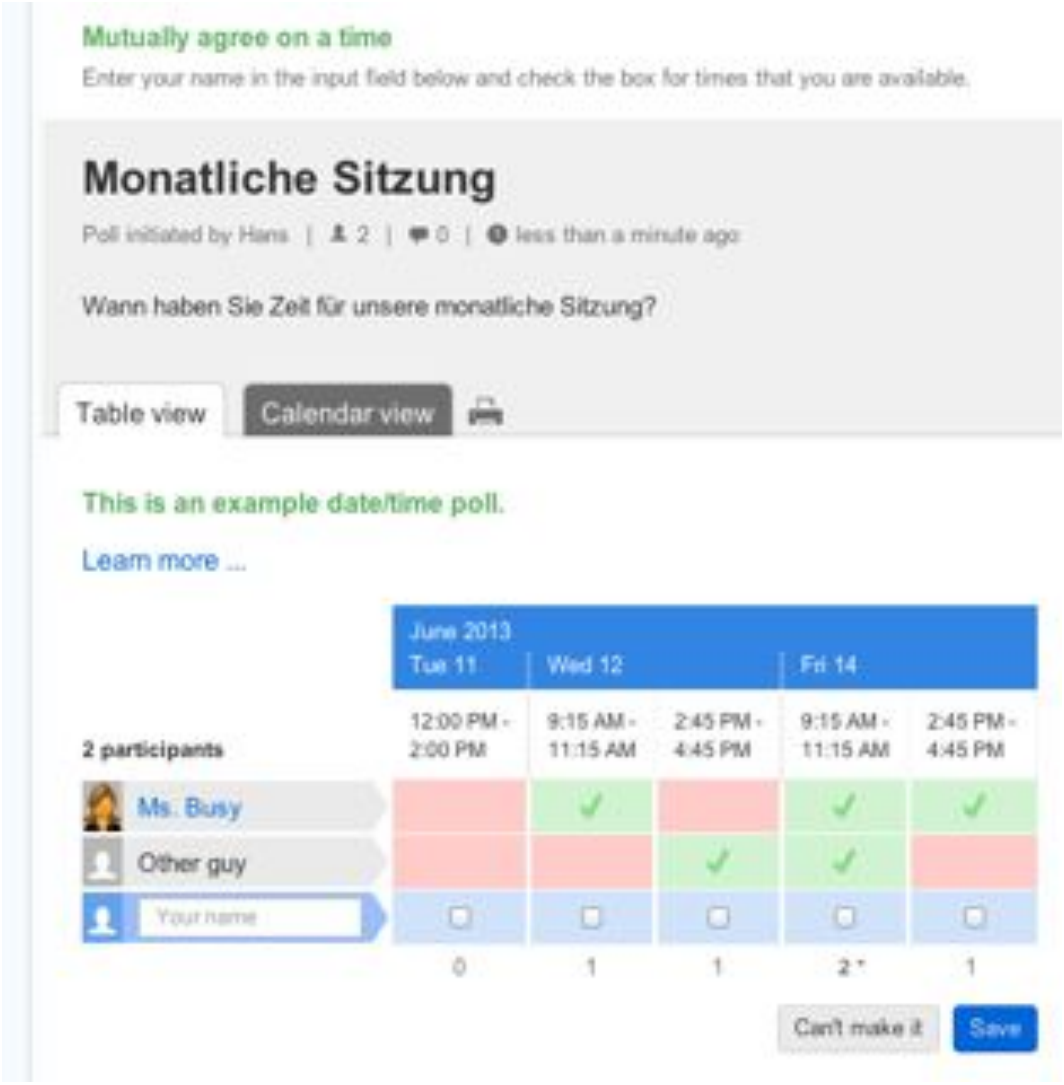


Figure 2.7: Part of Doodle website showing a poll for a monthly meeting

ate an account for most use cases, the interaction with this service is very easy. Accounts are possible to make, and they provide more comfort by, e.g., listing all polls by a user without having to manage all IDs manually. There is also an account model with payment and more cooperate features, but the normal user is enabled to use the key features already. With more features the user is enabled to connect the platform with a separate calendar from another online service to integrate with this service properly.

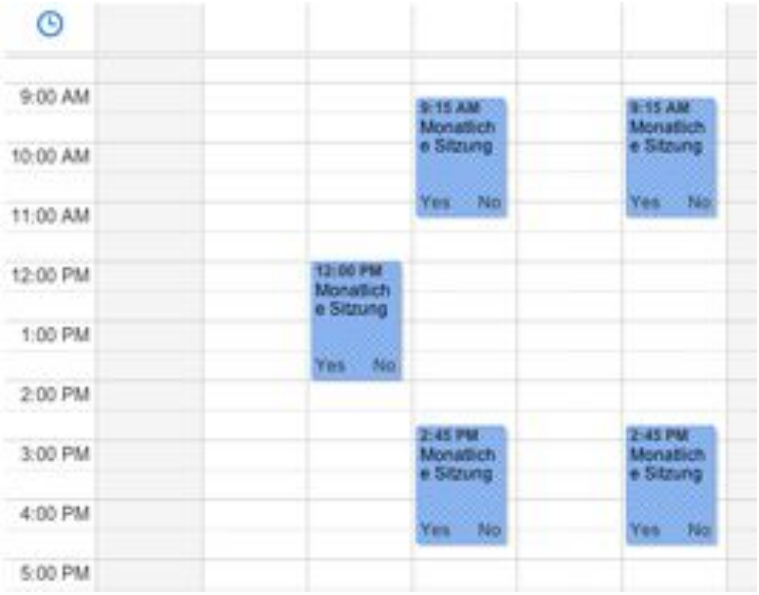


Figure 2.8: Same poll as in figure 2.7 shown as calendar view

Less complexity by less preset options.

With some well chosen combinations of variations the user can model almost every scheduling-process or poll into the given structure and gains a comfortable way to organize a larger set of responses, especially if the response currently entered is depending on the ones already made at this point of time. The process is made easy by cutting the mostly complex scenarios down to some preselected cases that the poll-starter defined. Also the complexity of having a list of options is lower than the one of a calendar showing many options scattered over several weeks or pages. The different views are shown in figure 2.7 and figure 2.8. Depending on the data either view can be of advantage. From Doodle we learned that complex scenarios can be handled by providing simple presets and conditions.

Conclusion of Doodle

Preserving the user experience by conditioning the visit.

For our system this method of scheduling is of interest. In most cases a fablab has only certain devices and there will probably not be a higher number of devices in each

device class. If a visit to a fablab is scheduled wen can avoid overbooking of devices and perhaps staff, too.

Limited capacities at fablabs encourage scheduling to avoid overbooking.

The capacity of a fablab is certainly limited, and to have a good user experience, some conditions like a free device or a free staff member to help should be met. This is of course depending on the local practice in he lab. If all users are able to use each device on their own and enough devices are provided, a scheduling system would not be needed and users may come and go as the opening-hours allow it. Depending on the practice of the fablab a user should be able to schedule a visit with the lab while stating which device is needed and which project they will work on. From Doodle we can learn that presets take complexity out of the scheduling process. Since the connection to an own calendar or the API

2.3.2 Schedule Once

Concept of Schedule Once

Concept does not differ to much from the one of Doodle, but is strongly aimed at business customers.

Similar to Doodle the main function of Schedule Once[12] is to provide scheduling between multiple parties. By aiming at business organizations or paying customers as a user base this platform differs from the first one. There is a trial-program for a few weeks per account, but after that the user is supposed to get a plan to pay by month depending on the features wanted. You may connect your account to existing private calendars like before and the strategy to reduce the complexity of scheduling is again reducing the possible amount of options and limiting the conditions of a meeting by the users preferences. A meeting could be set to a duration between 16 or 120 minutes in steps of 30 minutes, has to be planed four hours ahead, has to take place in the next month, and only may be placed inside the dedicated office hours and not-busy-time where no other calendar entries or meetings block the users time.

[12] http://www.scheduleonce.com

Conclusion of Schedule Once

Schedule Once provides scheduling for services, with or without requiring the service provider to validate the requests. In addition to that confirmation or reminder e-mails can be sent to give the user and the provider an instant update on events. We can learn from Schedule Once that sending e-mails is a sufficient and lasting feedback. Users and providers are able to archive the e-mail or even use them in other automated processes to integrate this information into their workflow.

Longterm usage with needed features only by charged account

There is a basic plan that allows accounts to be free of cost, but the included features are not sufficient to provide a integration into our system. Schedule Once allows embedding its service or sending automated reminders only with higher priced account plans. To integrate with especially this service would mean for our system to cost money, or would enforce it to gain a budget by alternative ways. Since most fablabs are operating at low cost and try not to charge the users over the cost of material and machines this situation is clearly to be avoided.

2.3.3 Google Calendar

Concept of Google Calendar

The calendar service by Google[13] is effectively a complex calendar back end, which is topped by an accessible web application for users to manage private or businesses calendars. Through a wide support of different protocols to gain access to that data this service is highly integrable into applications on the desktop or in the Internet. The own web interface by Google has got a high usability through short chains of interaction for the most use cases. Also it enables the user to use a very efficient user interface using, e.g., the drag & drop metaphor. Also Google Calendar integrates with the wide application landscape provided by Google and can easily be integrated by an API into third

[13] https://support.google.com/calendar/answer/2465776?hl=en

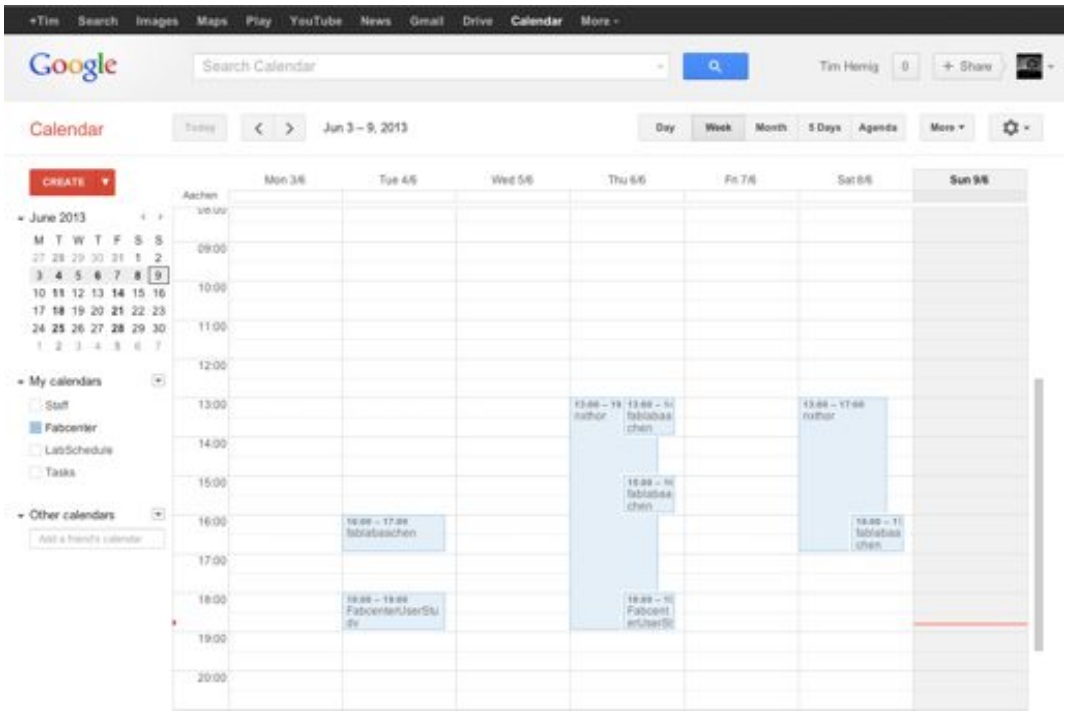


Figure 2.9: Google Calendar - View of the current week

party applications. Google provides a subscription service for external calendars that are described by a URL to a calendar file. Also sharing calendars between Google-users is possible with read-only- and write-access.

Conclusion of Google Calendar

Google Calendar is not targeted to be a scheduling platform (although the system provides functionality to achieve that), and does not provide a user interface like Doodle or Schedule Once. But it provides a very functional API to use the services and access calendars from within other Applications or platforms. A System to manage resources is only provided to Google Apps for Business and Google Apps for Education accounts. Either the account would cost money, or the fablab or similar institution has to be an institution at a university. There is a workaround for normal users.

2.3.4 Summary of Online Scheduling Tools

System vs. Feature	Schedule Meeting	Schedule Resources	External Calendars	API
Doodle	Yes	With charged Account	Yes	Yes
Schedule Once	Needed comfort only with charged Account	With charged Account	With charged Account	No
Google Calendar	possible	not for everyone, otherwise with workaround	subscription only, provides own calendars	Yes

Table 2.2: Online Scheduling Tools Comparison

Concepts do fit our needs.

In general the compared systems in table 2.2 meet the requirements, but Doodle and Google only provide a complete support with API only under the condition of a certain account. All three presented systems have a very clear strategy and seem to lower any barrier in scheduling meetings between multiple parties. To avoid costs we should re-implement a very basic version of this concept into our system and use the benefits of the Google Calendar API.

Mapping the concept on the system's conditions

With a given preset to choose from the system could show the possible time slots in the close future. The Visitor would request one or more alternative time slots for the lab's administrator to choose from. The latter one would accept one of these time slots, and the system would inform the user about this choice by e-mail. To communicate more or less preferred time slots, the user could use an comment-field. Depending on the preferences of a fablab a free device or a free staff member would be condition to a visit.

Chapter 3

Own work

In this chapter we will use the insights and conclusions of the previous chapters to design FabCenter, a web platform to guide users trough their fablab experience and the included documentation step. At first we present the results of a user study to identify the conditions of the use cases and the different potential user groups and stakeholders of the system. From that we will derive the system requirements and describe the design and implementation of the system in detail.

3.1 Initial User Study

At the beginning of this thesis it was the intention to create a project-oriented platform for fablab users. This was overwhelmed by the announcement and publication of the Thingiverse API at the end of the design phase. This API opens the door to a huge amount of already existing projects and a growing community of users contributing to projects and evolving them.

An initial online survey was conducted.

Before this event we conducted an online questionnaire using the online forms from Google Docs[1] to examine the different operating modes of fablabs in conjunction to their

[1] https://docs.google.com

own and their user's needs. The form of the questionnaire was compiled in English and German (see appendix C) and it was strongly targeted at users and administrators of fablabs. Since we decided to drop the project-management (see later in section 3.2) in order to import project-data from and export the documentation to Thingiverse, some of the questions are no longer relevant, but we now want to summarize the insights of this thesis. The detailed results are appended in form of diagrams and unfiltered listings in appendix C.

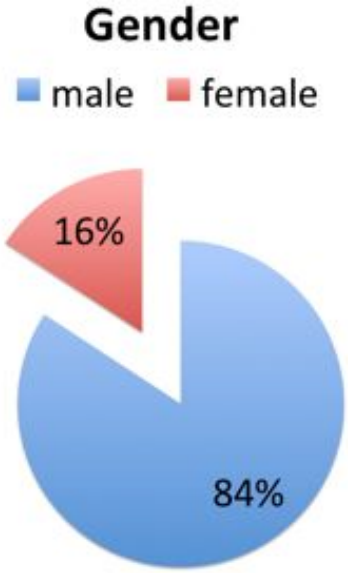


Figure 3.1: Gender distribution of participants in questionnaire

Interesting insights from agree-/disagree-questions:

- The questioned group consists out of 38 people, 16% female and 84% male participants, 55% users and 45% administrators of fablabs (see figure 3.1 and 3.2)
- The majority is a successful DIY-person, does not use platforms like Thingiverse, but would like to use a system, if provided by the fablab (see figure C.13).
- Most users do not have problems to get an appointment at their fablab, and have rarely to leave undone

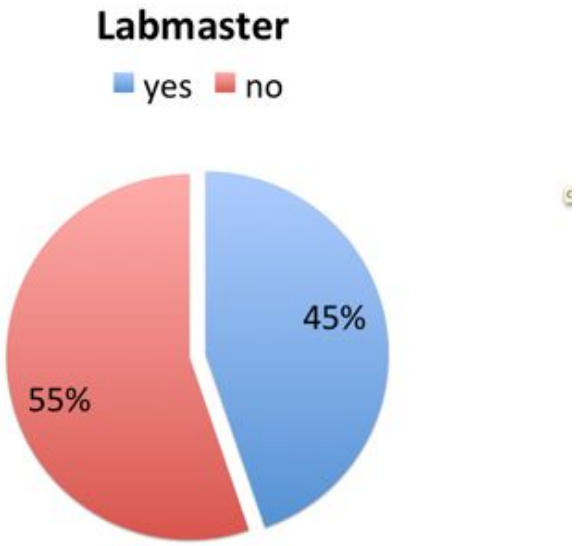


Figure 3.2: Rate of lab administrators in questionnaire

because of an overcrowded lab (see figure C.13).

- Contrarily to this the majority would like to schedule the visit beforehand (see figure C.13).
- If appointments were made, almost nobody would have to wait and was able to begin on time (see figure C.13).
- Almost all administrators offer open lab-days, where visitors have free access to the lab (see figure C.14).
- There is a slight majority stating that visitors often have questions about the soft- and hardware capabilities and need help getting started in the lab (see figure C.14).

Some participants also gave feedback via private e-mail, from which we learned that not every fablab requires the visitors to schedule a visit before using the lab. Others do not take in account that devices are already in use, since there are multiple devices of the sample class present.

We also asked for a wanted feature list of a system supporting the users and administrators of a fablab. The unfiltered compilation of all mentioned features is contained in the appendix (see figures C.15 and C.16 of the appendix C). We want to list the most mentioned and most interesting features:

- Save, share, view, send, sync, upload, search, derivation of projects
- File management
- scheduling, cost approximation
- community, accessible, wordpress, documentation system
- easy to use

3.2 Early Paper Prototype and Brainstorming

Also before the publication of the Thingiverse API a paper prototype of the initial system-idea was created. The intention was to provide a platform that would connect the user with the fablab and external web-based platforms. These external platforms would enable the user to use maker-devices like 3D-printer and laser cutters by web applications in the local network of a fablab. FabCenter was considered to provide user-, staff- and project-management, scheduling, and documentation-capabilities while offering an API itself to integrate with the mentions web applications. Figure 3.3 shows an early version of the scheduling-dialog a user would use to plan a visit with the lab by providing all needs and selecting requests from a list of possible dates.

Redesign of the system based on brainstorming session with experts.

With publication of the Thingiverse API the prototype was used in an internal brainstorming session with professionals of the field of Human Computer Interaction (HCI). As a base for discussion a short presentation with a rough diagram of a system architecture, a conclusion of the initial

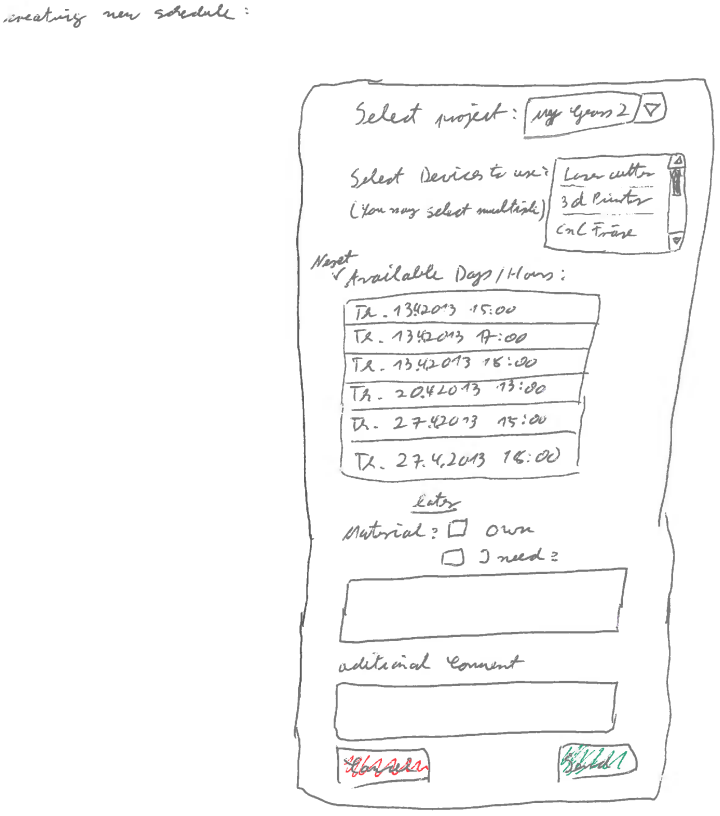


Figure 3.3: Scheduling-dialog of the very early paper prototype

survey from section 3.1 and several user stories was used. The slides of the short presentation are included in the appendix D. The original concept was dropped, and we decided to redesign the system upon the Thingiverse API to gain from the huge landscape of fast upcoming third party applications integrating into the platform of Thingiverse.

These third parties' applications would also help our system, since users are able to use the variety of tools to create designs in a more comfortable way. To mention one example, users are already able to create parametric 3D-object designs inside Thingiverse to vary a generic design by another user. With existence of the API, it is no longer worthwhile to start an own landscape of systems, when a huge community already is creating an even bigger one that is

usable by everybody. FabCenter could also be contributing to this landscape by providing a higher quality of feedback on the projects inside Thingiverse.

Integration of Thingiverse has got pros and cons.

Integrating only Thingiverse accounts would mean to leave some users out, since they probably are disinclined to create an account. FabCenter would need to recognize the user anyway, so having any account to facilitate this is a given situation. For a first system to begin with, Thingiverse provides the best infrastructure that is able to cope with projects. In later stages of the development of FabCenter multiple other project-hosting platforms can be included, or even other kinds of platform may be integrated, if the provide an API to store project related data in any way.

Rethinking the stakeholders of the system.

The brainstorming session also redefined the stakeholders of the system inside a fablab. Starting from the user stories in appendix D we developed a picture of the participating parties.

The user or visitor wants to print or cut a design, needs access to the tools, is interested of other projects and seeks help through sharing and discussing own and others projects. The administrator of the lab needs to organize the schedule of visitors, lab activities (including staff if present) and the devices. He needs to know about material stocking and tries to raise the lab's internal experience and knowledge by documenting the lab's activities.

Less different platforms or accounts for users.

It is very unlikely that a user wants to upload his designs into multiple platforms and manages to keep all sets of data up to date at the same time. Also the targeted mainstream-user would most probably have or easily get an account on Thingiverse, so he or she will not have to create a second account on another system.

The complete set of sketches of the paper prototype can be seen in appendix E. By deciding to use the Thingiverse API the complete management of projects, mentioned in the wanted feature list in section 3.1, is already a solved issue. Users can pick up projects from others, evolve them, give feedback, derive them into own projects and, of course, manage all the files in the project including photos and

other media. Also there is an existing community, a very accessible platform that is capable of representing a whole documentation of a project, to which we can contribute with our system.

Scheduling will be an unsolved issue for our system, since Thingiverse cannot provide that, and the compared services do not completely fulfill our needs (see table 2.2). As discussed in section 2.3.4, we will implement the needed scheduling system on top of the Google Calendar API.

3.3 System Requirements

As already mentioned, we decided to implement our system with the premise to use the Thingiverse API for project management.

Furthermore we need to implement a basic version of the presented concept of scheduling strategies by Doodle and Schedule Once that is able to consider available staff members and devices in a lab.

As the conclusion of the SHARE-project in section 2.1.1 described, we want to feed information back into the project, so we need to intercept the user directly after making the current project in the lab and, if possible, before leaving the lab. We need to guide the user into finishing the small step of documentation by providing photo and text to append the information to the project. With the proof of concept through the work of Fabiji by He [2012], as described in section 2.2.2, we can expect to have a fablab provide a feed of pictures showing the creations of the day in a lab. Therefore we need to import photos from such a stream to easy the task of creating documentation.

The system should help the administrators to organize the lab's schedule and devices.

We can formulate the main requirements of the system as:

- M1: Support of the documentation process in fablabs

- M2: Support sharing of documentation the Internet
- M3: Help organizing internal processes of fablabs
- M4: Offer a usable system

To specify some additionally requirements the system should meet related to the main requirements mentioned above we can define:

- R1: Integration of Thingiverse via API
- R2: Basic scheduling System for Visitors of the fablab according to the fablab's settings
- R3: Mechanism to guide users into documenting the current visit to the fablab
- R4: Provide access to many fablabs or alike institutions
- R5: Enable the administrator to organize staff and devices
- R6: Provide a feed of Fabmoments to the public.

M1: Support of the documentation process in fablabs.

Documentation is a key to creation of value.

As repeatedly discussed in sections 1.3, 2.2.2, 2.2.3, 2.2.6 and 3.2 the gain in online sharing of digital designs lies in the feedback the designer gets. Fablabs are a place of creation, and mostly digital designs are used. We need to encourage documentation of these events.

M2: Support sharing of documentation the Internet

Sharing the creation of value keeps communities alive.

To extend the requirement M1 from section 3.3 we now need to put the just created documentation into a publicly available place. The community, which maybe supported the creation of the design should gain the created value, too. Therefore the sharing of the documentation is a main

requirement of this system. It would encourage new users to contribute as they gain experience. This exchange keeps communities alive and improves the collective knowledge over the whole group.

M3: Help organizing internal processes of fablabs

A clear and easy way to manage reoccurring processes is needed.

Administrators of a lab need to know multiple details about their visitors beforehand. In order to reduce the energy put in management and raise the amount of time spent with the visitors, a scheduling system is clearly a benefit. Instead of using external services with costly accounts involved we need to provide a portal that can hook up the user with the local fablab without any barrier in between. The goal is to accomplish this by implementing a scheduling service inspired by the discussed services in section 2.3.

M4: Offer a usable system

User centered design to raise the quality of the service.

As Norman described in Norman [2002], a user interface should be designed user-centered. The fact that there is a computer, operating system and a web browser should stay behind the primary function of the application. Ubiquitous computing as coined by Weiser [1991] supports this idea at large. Web-based applications already are seen as entities that no longer require the description that they are found in the Internet and need a web browser to be shown on a computer. Usability is a requirement to secure the quality of this service. A website that is hard to use will not have any impact on the workflow of any user since it will most likely not be used.

3.4 System Architecture

FabCenter needs to be accessible to a broad and active user base that already communicates via the Internet. With the decisions mentioned earlier in this chapter we also need to

integrate web-based services, therefore FabCenter has to be a web-based application, too.

3.4.1 Technology

In order to implement a web platform we decided to use the Apache[2] web server to host the whole implementation and deliver the dynamic HTML-content (Hypertext Markup Language). Apache was chosen since it is a reliable, wide spread and highly supported technology, which proofed itself in uncountable productive systems worldwide. With the same argument we also settled with PHP[3] as the primary programming language to implement the dynamic content. Both are free available and easy to deploy on almost every computer system.

To benefit from the Model-View-Controller pattern as initial described in a paper by Krasner and Pope [1988], we base our main implementation on the PHP-framework Codeignitor[4] . This framework provides internal access to a relational database, prepared structure for MVC, capabilities for the REST-technology (Representational State Transfer) to connect to the API's of other services like Thingiverse that will comfort the implementation of the system. REST is demonstrated by Battle and Benson [2008] as a bridging technology between web platforms.

To extend the usage of the MVC-pattern, we also modified the Codeignitor-framework with the Smarty template engine[5] to have a cleaner separation between design and code.

MySQL[6] will form the foundation of the back end in function of a relational database since it is also highly supported, widespread, and deployable across multiple hardware- and software-platforms. MySQL is a server software to provide access to relation databases via SQL (Struc-

[2] http://httpd.apache.org
[3] http://php.net
[4] http://ellislab.com/codeigniter
[5] http://www.smarty.net
[6] http://www.mysql.com

tured Query Language). SQL is a development out the original language SEQUEL (A structured English query language) introduced by Chamberlin and Boyce [1974] and later renamed due to copyright issues. SQL provides a language to request complex datasets out of relational databases.

To provide a back end for the schedule of a fablab and a source for photos, we decided to make use of Google's calendar[7] and photo service (Picasa Web[8]). This is also an back end to which the local fablab has got own access, to eventually integrate with other applications or use cases.

To have the layout of the views defined in a more consistent and well formed way, we user the Twitter Bootstrap[9] CSS- (Cascading Style Sheets) and JavaScript-library including some dependencies and own extensions. CSS is explained in detail by Briggs et al. [2004] and used as a efficient tool to divide content and representation of web pages. JavaScript is a client-side scripting language integrated in many web browsers. Through the usage of a JavaScript-engine, websites are able to manipulate the content of the rendered page without involvement of the delivering server. A good introduction and description of the technology can be found in Flanagan [1998].

Overall a overview of the used technologies and their interaction can be seen in figure 3.4. The user's browser sends its request to the Apache web server, which calls the PHP-framework with the contents of the request. Our main controller inside the framework will invoke one of the implemented controllers that will gather all needed information from the data models and will call actions to manipulate the model if needed. After that, all gathered information is put into the template engine along with the template in charge of generating the needed view.

[7] https://support.google.com/calendar/answer/2465776?hl=en
[8] http://picasaweb.google.com/
[9] http://twitter.github.io/bootstrap/

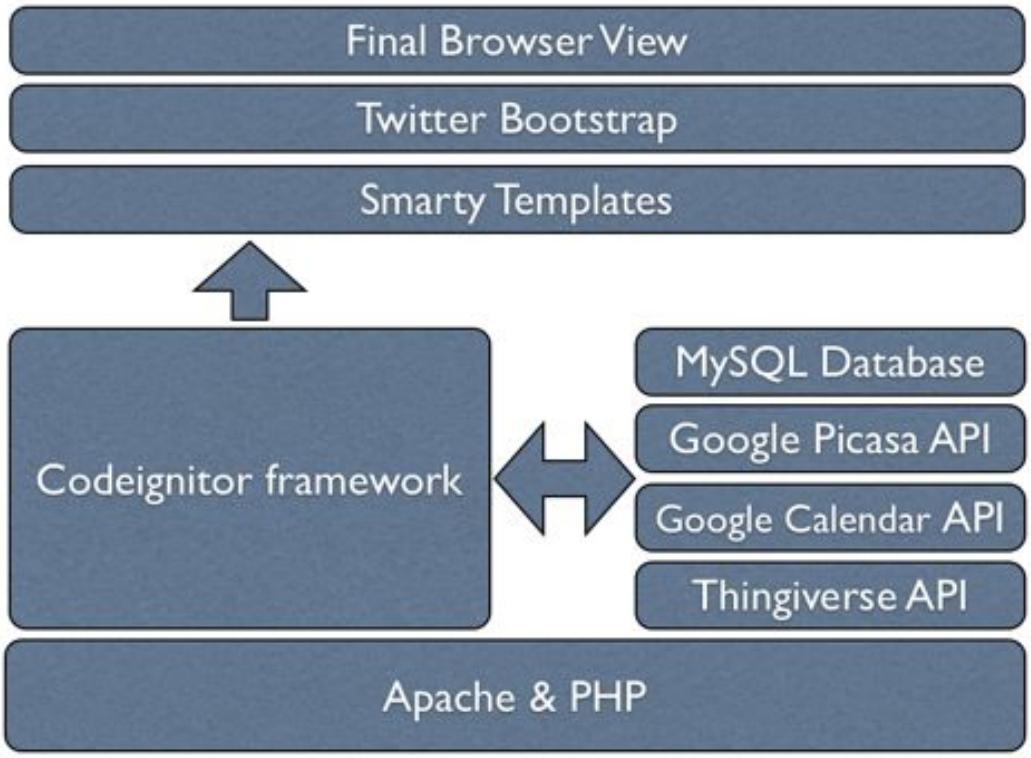


Figure 3.4: Overview of the general system architecture

3.5 Implementation

3.5.1 Login and Authentication

User login and authentication via oAuth2.

To facilitate login and authentication with the Thingiverse API, we need to implement the oAuth2[10] authentication mechanism, which is sketched in figure 3.5.

Basic mechanism of oAuth2.

At first the user wants to login, and sends the corresponding request to our system. We redirect the user's browser to the login-site of the remote service along with our registered application-ID. After the User logged in and granted the needed access to our application ID, the Remote Service redirects the user's browser back to a registered URL of FabCenter, where we now receive a temporary code. To proof to the remote service that we are the registered appli-

[10] http://oauth.net/2/

cation and that we sent the user to login we now provide the remote service with the temporary code and the application ID. As a response we receive a token that enables us to act on behalf of the user toward the remote service as long as this token is valid and we can proof that we are the same registered service as before. The proof itself is a cryptographic challenge response problem with asymmetrical keys. On registering the application with the remote service, a unique application ID and a unique secret key is produced, such that only the applications infrastructure knows the secret. In the whole process the user never had to show the application his or her login credentials, and the application did never show its secret to anybody involved in the process.

All requests between the three parties were made with the REST-technology, which basically consists of different HTTP-requests (Hypertext Transfer Protocol) to the other server. In general this HTTP-requests are bound to HTTPS (HTTP Secure). HTTP is introduced by Berners-Lee [1989], who also implemented the first working client-server system based on HTTP. HTTPS extends HTTP with an encryption layer to prevent man-in-the-middle attacks on communication between HTTP-client and -server.

The data format itself is mostly the text-based format JSON (JavaScript Object Notation, also adapted by many other languages) that enables us to easily exchange concrete data structure between servers. JSON is presented by Crockford [2006] as a lightweight data exchange format.

A very basic request to a REST-API using the cURL[11] library may look like this:

Most API-requests to remote services are done by cURL.

```
$this->curl->create(
$this->tvURL.''search$query''
.''?access_token='' .$this->User->tvToken);
$buffer = $this->curl->execute();
$result = json_decode($buffer);
```

As seen in this code example the provided infrastructure

[11] http://curl.haxx.se

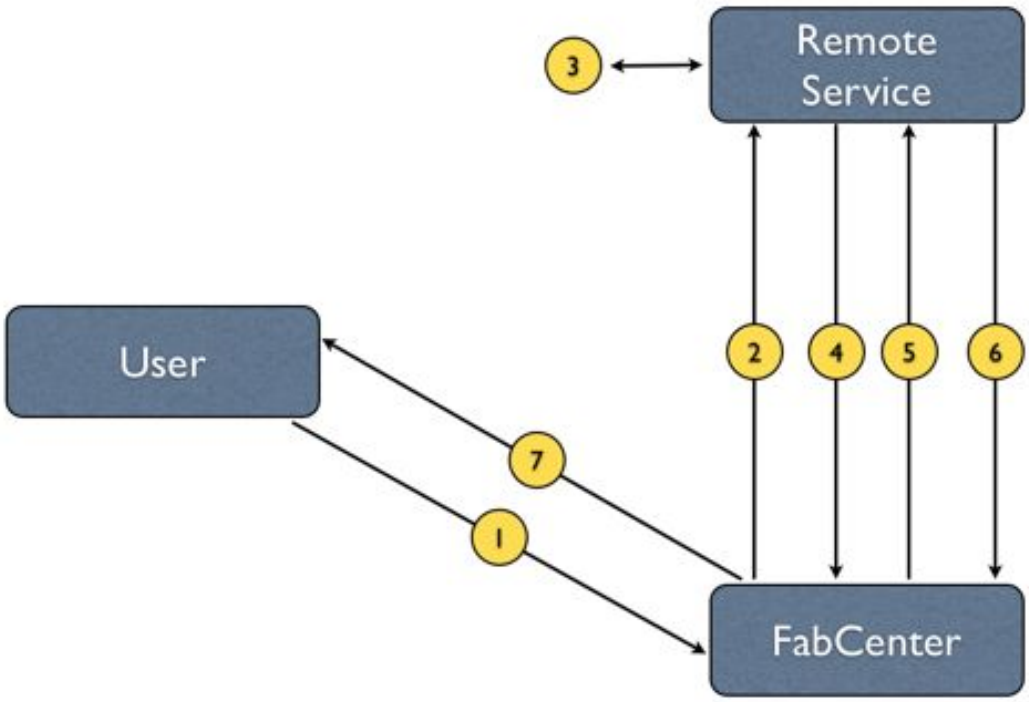


Figure 3.5: Mechanics of the oAuth2 authentication.
(1) The user sends the login request to FabCenter
(2) The browser gets redirected to the remote service
(3) User logs in into remote service
(4) Remote service redirects the user with temporary code the the FabCenter website
(5) FabCenter authenticates with own application key and the received code
(6) Remote service answers with access token
(7) User is logged in, FabCenter may act on users behalf toward the remote service

of the framework and an added third-party package make such requests relative easy.

3.5.2 Database Structure

MySQL provides access to relational data.

To manage the data that is not provided by the remote services or stored into them, we needed to define a set of relations to be able to save, manipulate and retrieve data stored in a MySQL-server. To remember a user, e.g., we use the user-table or relation to store an ID of the type integer, the

user name from Thingiverse, his e-mail address provided by Thingiverse, Information about the current token to access the API, timestamps of the date the user was created (first visit) and logged in the last time, and last but not least the internal ID of the fablab a user visits as default, and current.

The e-mail address of the user is needed to be saved in FabCenter, since we may send the user an e-mail while we do not have a valid token to read it again from Thingiverse. The Thingiverse API Terms of Service[12] states "You shall not: [...] Use the Thingiverse API to spam, collect personal data or otherwise harass users.". According to this we are allowed to save the data, since have no intention to just collect the personal data, since we need it for internal processes, and under no circumstance the e-mail address is shown to other persons than necessary on an need-to-know basis.

Part of the data in the mentioned table can be changed in the profile-view of the user. The current fablab indicates where the user wants to visit now, the default one is the fallback, or indicates that the user is a staff at his default fablab, although the user wants to visit a different institution.

The overall structure is presented in figure 3.6.

As we proceed with the description of the implementation we will refer back to the figure 3.6.

3.6 Web Application FabCenter

Next we go through the different sections of the implemented web application, and describe the functionality.

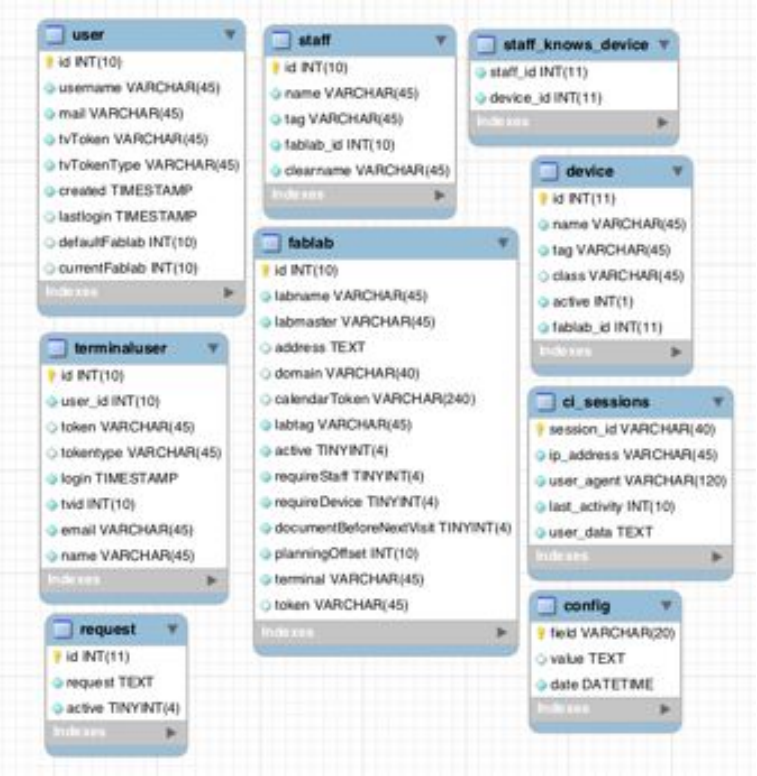


Figure 3.6: SQL-definitions of the used tables

Login screen

The first page the user sees is the welcome-page (see figure 3.7). A short introduction to the system is made, and a prominent button to login is placed. Also the user has got his first encounter with the primary menu bar at the top of the page.

Since we have several pages that depend on fast response times by the remote services it may occur that the user will experience that a page will not load instantly after clicking a button or link.

In order to communicate that the system is still working and will come back to the user in a short time, we implemented an overlay with an waiting or loading-animation

[12] http://www.thingiverse.com/developers/api-legal

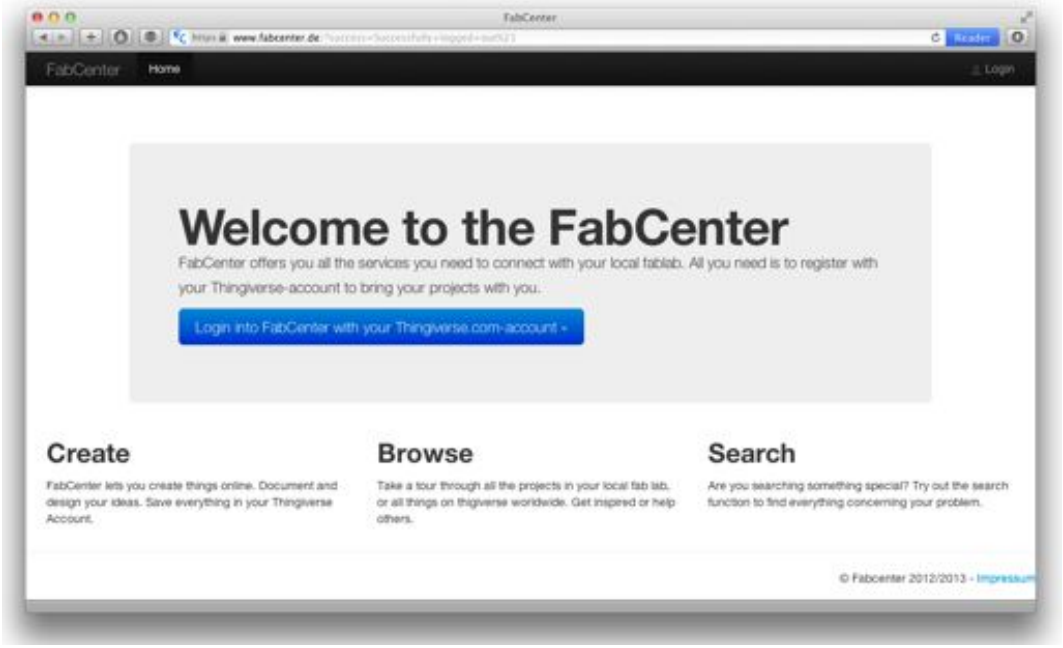


Figure 3.7: Welcome screen of FabCenter

(see figure 3.8 for detail). This overlay will be made visible by CSS and JavaScript right after the mentioned user-interaction. As long as the browser will not render the new page, this animation will signal the user to be patient. The benefit of this mechanism was not explicit covered by questions in the later user study (see 4.1), but we were able to observe that any participants showed signs of awkwardness or impatience as long as the animation was shown.

My Profile

After a successful login with the oAuth-method described in section 3.5.1 a first time user will be prompted to choose his fablab to visit. After that the user is directed to his profile, where the personal settings can be reviewed (see figure 3.9). Currently the system needs to know which fablab is the default fablab, and which one to visit beside that. As already mentioned in section 3.5.2 this default-information lets the system know where the user is located in case of the user being a staff member in perhaps multiple fablabs.

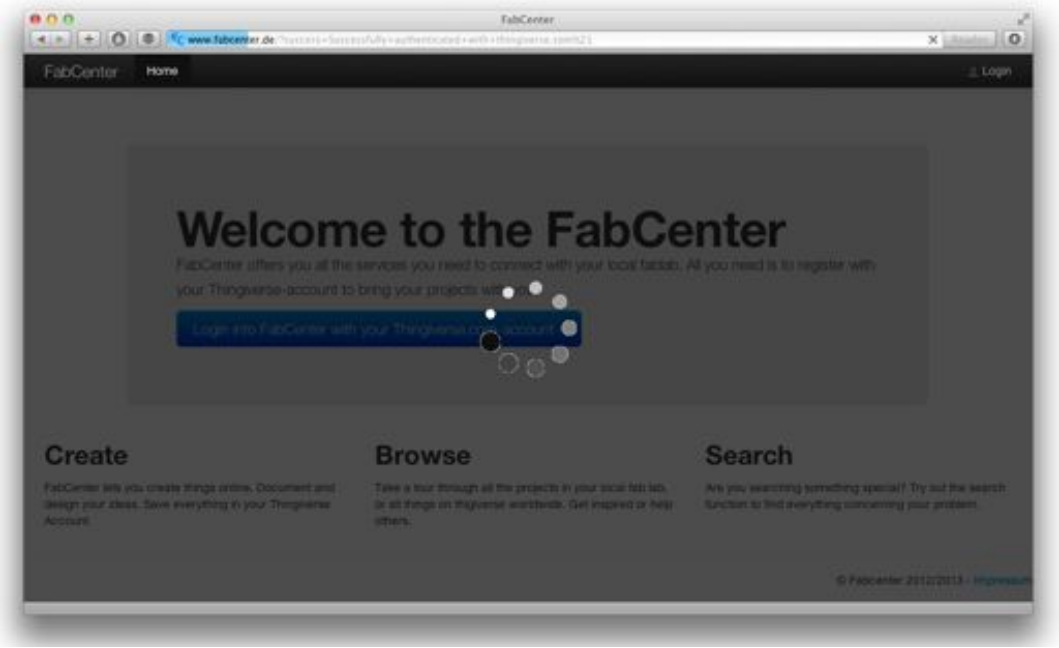


Figure 3.8: Loading animation of FabCenter

Then the system would know which lab to use for the "My Lab" section (see subsection "My Lab" later in this section).

Projects

In general the user will visit the site in preparation for his visit to a fablab. We want the user to select one or more projects when scheduling a visit, so we also mus provide a opportunity to browse the own projects, the ones he "liked" on the Thingiverse platform, and other lists provided by the Thingiverse API.

To facilitate this, we implemented a view to show different lists of projects, and a view to get an overview of one specific project. In figure 3.10 an exemplary view of the most popular things on Thingiverse can be seen. Similar lists are available for each category and shown in a secondary horizontal menu. In addition to the categorized lists we also provide a search over the projects on Thingiverse (see figure 3.11). The user can enter a search request and get a list

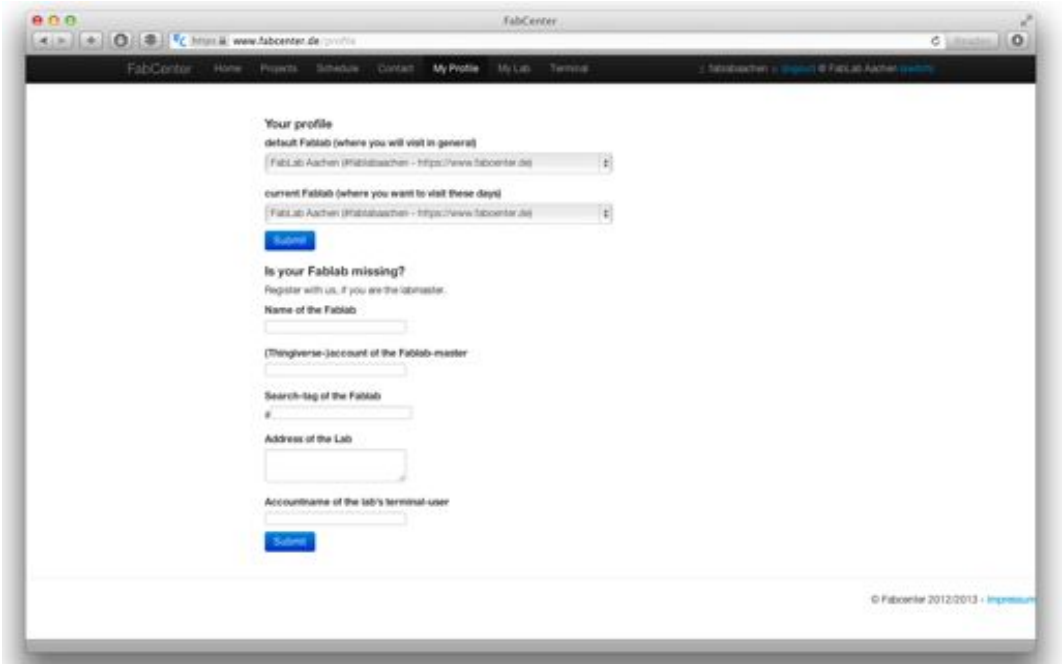


Figure 3.9: User's profile setup section

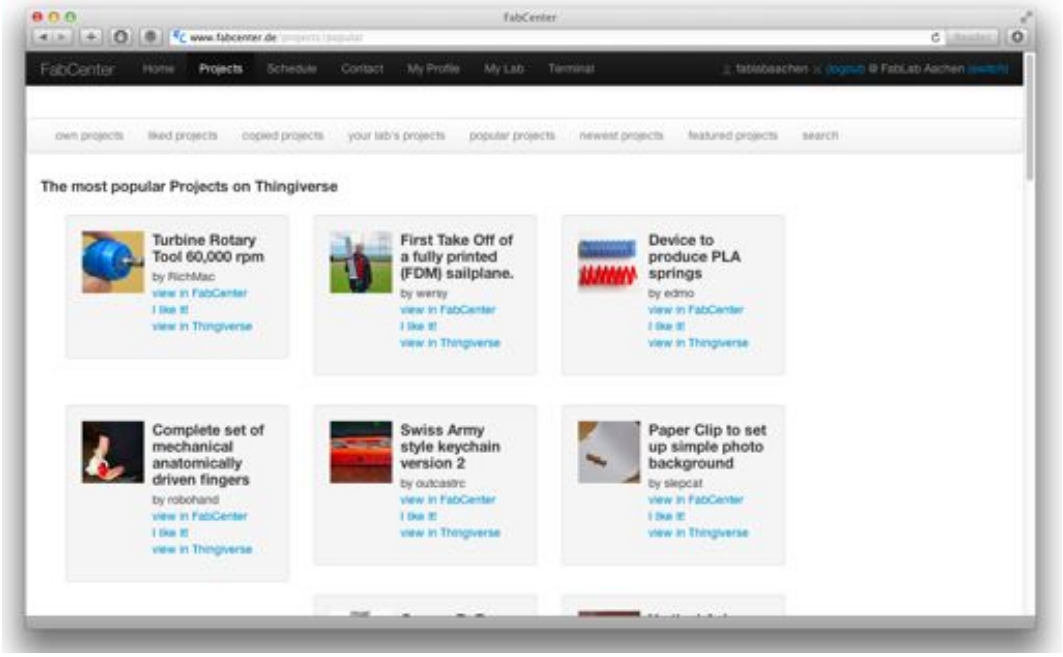


Figure 3.10: List of projects in FabCenter

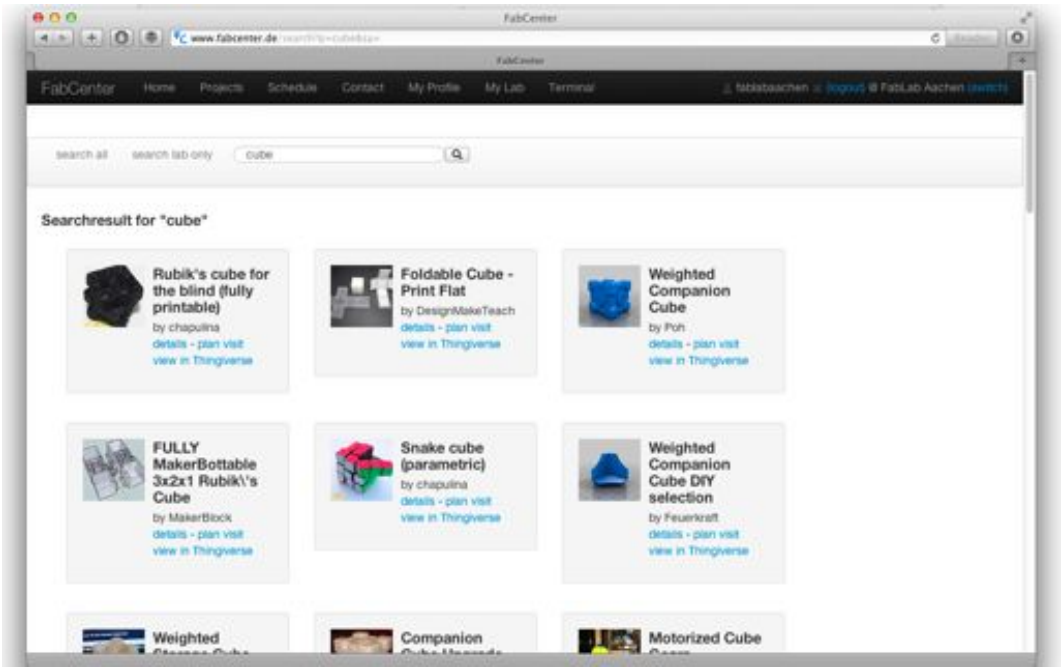


Figure 3.11: The search section in FabCenter

of all matching projects sent back from the Thingiverse API.

After finding a project of interest the user is able to view the details of it inside FabCenter or alternatively open the project's original homepage on the Thingiverse platform in a blank browser window. Figure 3.12 gives an overview of the project-representation inside of FabCenter. The Page is divided into three columns.

In the middle we can see a display for the different pictures listed on the right hand side. This pictures are connected to the project and may be actual photos of the project, previews from the source files of the project or photos of copies, other users have made. By clicking on one thumbnail, the display in the middle will switch to that picture and show a bigger version of it. To completely enlarge the picture, the user can click on the display and get an full size representation in a full screen overlay made with the use of CSS and JavaScript like the loading-animation before. Right under the display the user can find the description of the project, and the provided instructions to assemble or produce the project's content. On the left hand side there is

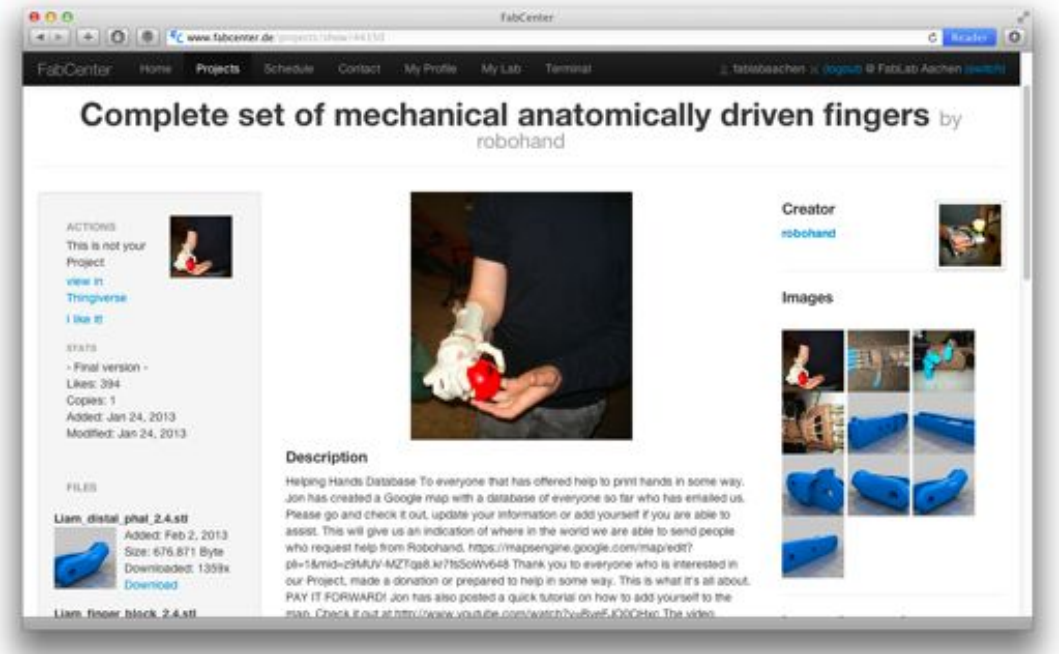


Figure 3.12: Project view in FabCenter

a short section about the projects with thumbnail, statistics from Thingiverse and some actions that can be taken by the user. These actions depend on owning the project. The user is able to mark foreign projects as "liked" and open them in Thingiverse. On own projects the actions are to finalize the project (if it is a work in progress on Thingiverse), make it public (if it is still private and invisible to others) and to plan a visit to a fablab with it (with having the projects preselected in the form of the first step of scheduling). Underneath the actions and statistics the user has access to every file belonging to the project. Since the user most likely will use this download only in the location of the fablab, and probably on one of the computers there we decided to rename all files requested for download in order to help the administrators with the file keeping on the computers. Every file is named after the user downloading it, the project ID on Thingiverse and the origin filename including the file extension.

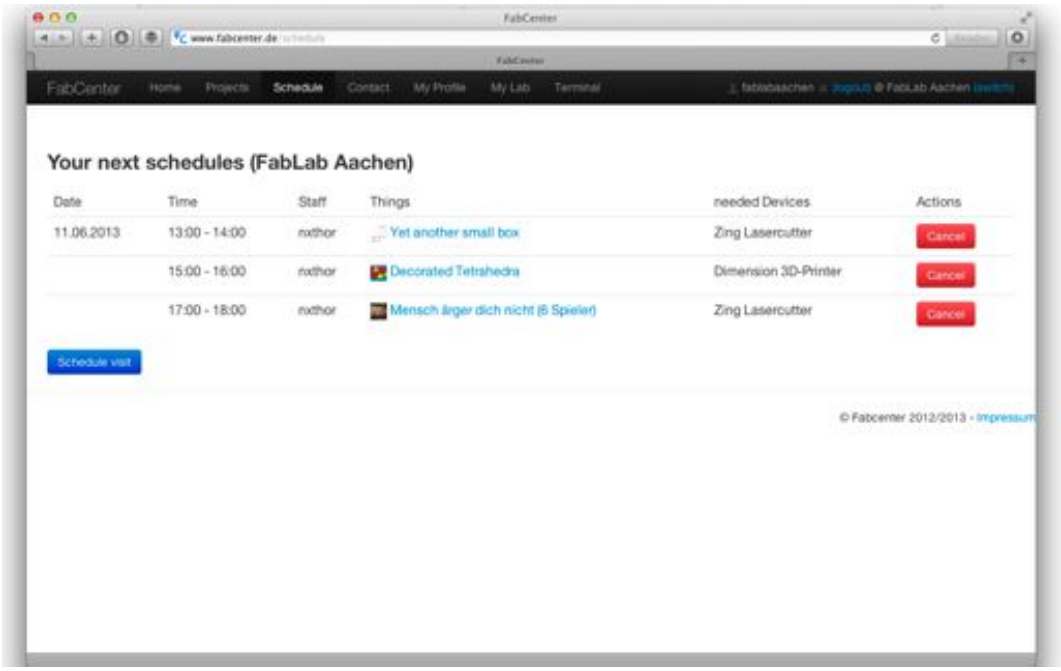


Figure 3.13: User's view of own scheduled visits

Schedule

In order to schedule a visit that we just mentioned the user also has got an entry in the primary menu to see his own already scheduled visits and has got the option to plan a new visit. The basic schedule is shown in figure 3.13. Moving over to figure 3.14 we can see the first step out of two of the scheduling dialog.

At first the user is asked to select the projects with which the user wants to visit the lab, and which device of the fablabs to book. Finalizing this step the user is able to add an comment to the request to have a channel to ask questions or to inform the lab's administrator about some issues regarding material or capabilities of the devices.

In the second step the user is shown the possible free time slots that the user is able to book. The user may select the most fitting slots, and complete the request with the option to review the comment from the last step. In case of preferences on the time slots, the lab's administrator would need

Figure 3.14: First step of scheduling a visit

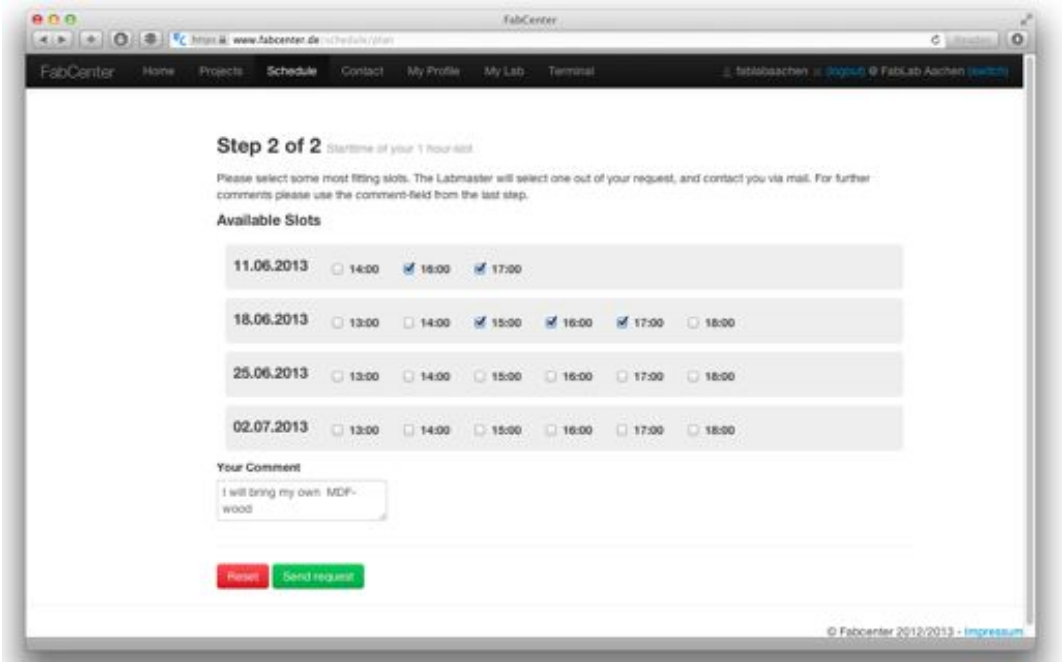


Figure 3.15: Second step of scheduling a visit

to know about it or similar issues. To identify all possible time slots for this representation the system takes the following conditions into account:

- We assumed that a user might need around one hour machine-time. Reason for this is an example of booking by the hour at the fablab in Luxembourg[13] , own experience and less complexity, since minutes are no longer an issue to deal with.
- To the given hour at least one staff member has to be present and awaiting visitors. Depending on the mode of operation in the fablab staff members may also be present, but cannot take care of visitors. The system differs this state in "public" and "private".
- The staff member has to be available and may not already be busy with another visitor.
- The requested device has to be available and may not be booked already.
- The present staff member has to know the device and has to be able to use it.
- The booking has to be planed a defined number of hours ahead, to give the lab's administrator the time to prepare and react the requested schedules.
- The booking window is reduced to one month in the future to stay flexible in the organization of the lab.
- No other event in the calendar dedicated to save the schedule-information may be coexisting during the time slot. If "the lab" has a vacation day or a holiday occurs, this would prevent booking.

To support other operation modes of fablabs, the lab's administrator may setup the preferences of the lab to ignore the overbooking of devices or "public" staff members. Also the booking windows can be configured (see figure 3.16).

This way we are able to break down the scheduling problem to a list of manageable options from which the user may choose some.

[13] http://fablablux.org/booking/

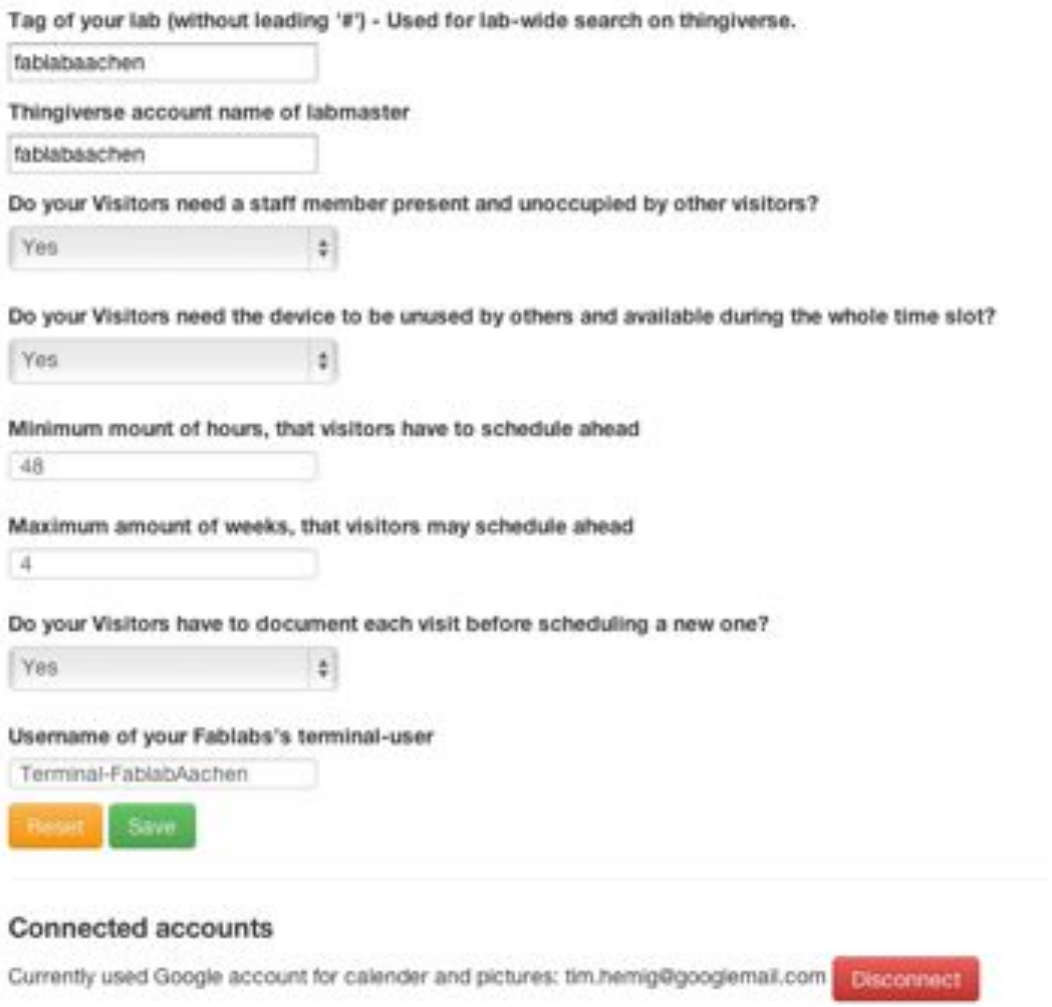


Figure 3.16: Detail view of the lab's setup

In the end of this process the request of the user is saved into the database (see figure 3.6) as a serialized PHP-object and the lab's administrator is informed via e-mail about this ongoing. The user will get feedback on the next shown page about the success of saving the request and the next steps. Now the lab's administrator needs to take action and has to use the system to accept one of the requested time slots. As a result the user and the lab's administrator will get e-mails with the details of the finally scheduled visit. The view that enables the administrator this last step will be discussed in a later subsection as we further progress through the system.

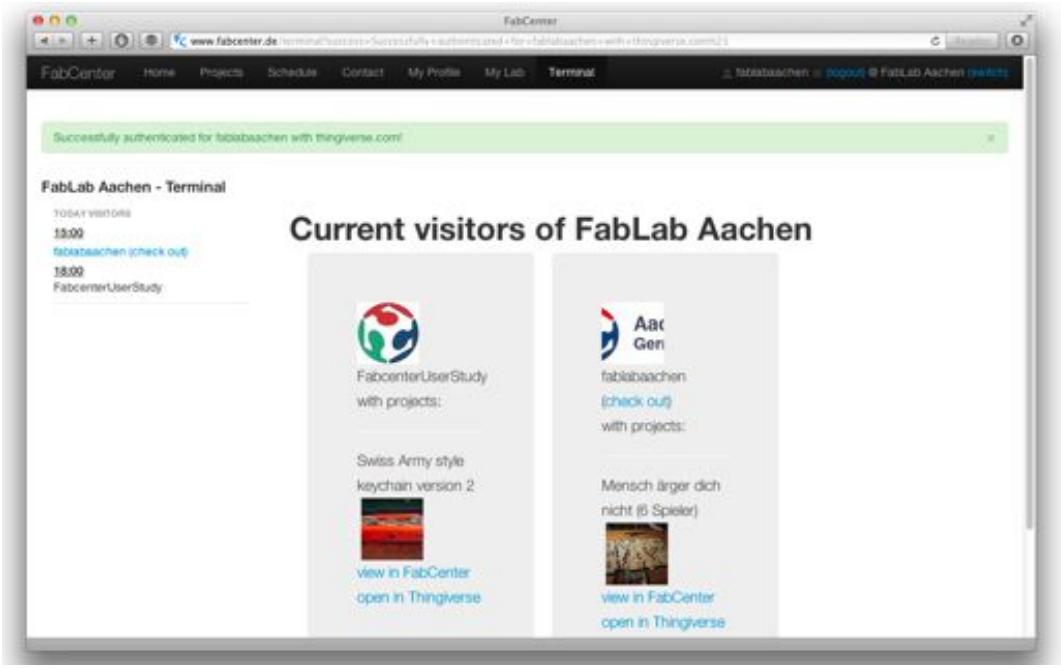


Figure 3.17: Terminal view of FabCenter showing the current present visitors

Terminal

One of the main intentions of the FabCenter system is the support of users and administrators of fablabs documenting their experiences. The terminal section guides the user through the visit. The user is supposed to check in at the beginning of the visit and check out before leaving the lab. Of course the labs procedures have to support this feature since the web application depends on the necessary participation.

The terminal view shows a list of the today's visitors in the lab. After checking in each user will be shown in the middle of the view as a current checked in visitor with the projects the user wants to work on that day. This view also provides the user with the check out action, if the user is already logged in. Figure 3.17 gives an overview of the layout.

Redesign of the terminal due to technical problems with the authentication process

The original design was destined to have a terminal-user or no user at all logged in into the system itself, but have the terminal maintain the checked in users. This would

have supported having a single terminal-computer inside the lab's rooms, to which users can walk up to and interact with the system. Using multiple browser instances or multiple clients was no problem. Due to the session management and the oAuth2 mechanism maintaining the users in one session would be disrupted by logging out and in again to Thingiverse in order to check a new user. As soon as the new user has to login with Thingiverse to validate the login into the FabCenter, the currently logged in user would be kicked out of the browsers session with Thingiverse. The user, which has to log out of Thingiverse also destroys the valid token that the system was using to get the information about projects or to post information back to the system. Users would have to repeatedly enter their credentials in order to work with one single point of interaction for example at the entrance to the lab. Since this concept got changed very late in the development of the system, there are still the terms of "login" or "logout" and "check in" or "check out" in the user interface. In the further development of the system after this thesis this issue will be resolved through merging this terms into the same functionality. The terminal would show the same capabilities but on a login the terminal would automatically check in the user if a visit is schedule for this day. On leaving the workstation and logging out of the system the FabCenter would trigger the check out of the system, too. This mode forbids that multiple users may use the same instance of a web browser application and enforces the usage of either multiple instances of the same browser application, different web browser applications per user, or different hardware clients for each user.

Having mentioned the process of checking out we now want to describe the process of checking out itself. After sending the request to check out, the system shows a pool of pictures to the user including a form to enter some comments or descriptions about the pictures. These pictures are fetched from the fablab's photo gallery inside the service of Google Picasa Web. The settings enable the administrator to choose an album out of the Google Account that is already connected to the lab since it is used as the back end for the schedule as mentioned in section 3.4.1. Each picture is shown with its filename and a list of check boxes, which the user can activate to assign the picture to one of

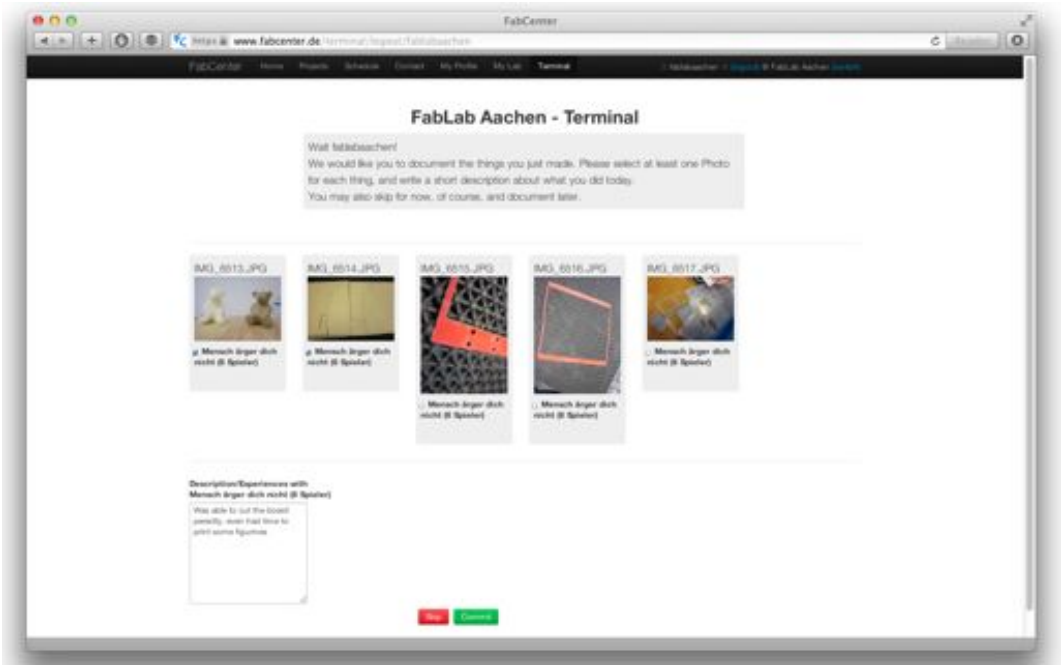


Figure 3.18

the projects the user visited with. Each check box is identified by the name of the project, as is the text box for description or comment underneath, too.

The user may finish this form and commit the data into the project, or skip the process for now, what will prevent any documentation. If the user commits the data, the images and the text will be added to the gallery and description in case of an own project, or added as an "copy" to Thingiverse. Now other users browsing this project have the opportunity to see this information and consider it in their own review of the project.

My Lab (for lab administrators)

With providing the services for users there also comes the need to manage the processes behind the scenes and setup the concrete environment for a fablab inside the FabCenter.

First there is the schedule view for lab administrators as

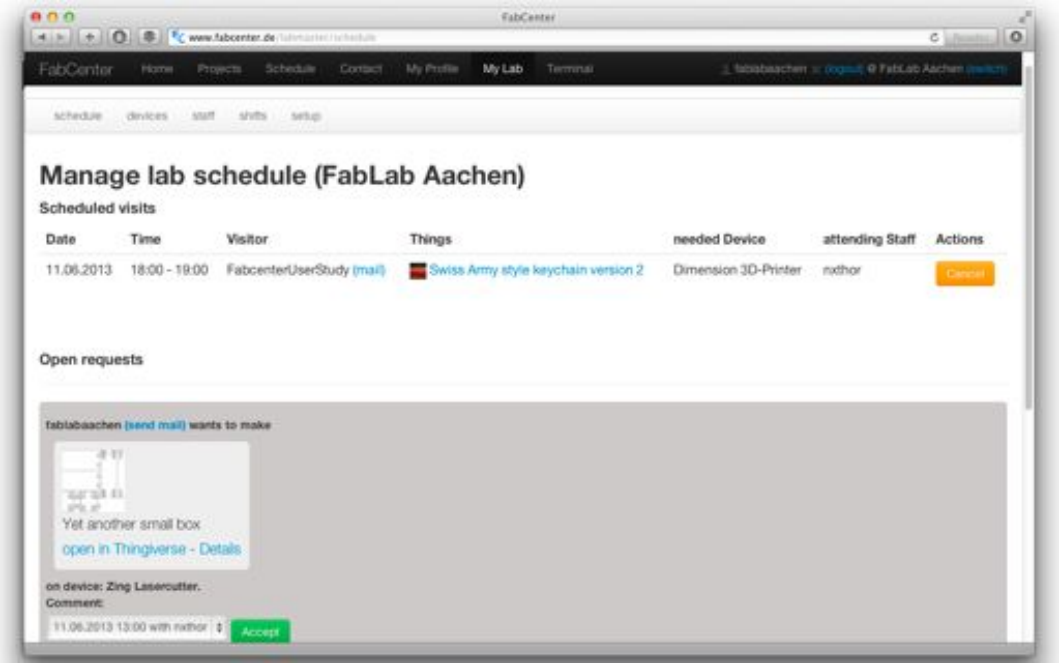


Figure 3.19

shown in figure 3.19. Like the user before, the administrator can see all upcoming visits to the own fablab, review the information, cancel, reactivate or even delete one of the entries. Also the incoming requests are shown here if they exist. At the bottom of the figure 3.19 we can see such a request. The lab administrator is provided with the user's name and additionally the e-mail address in form of a clickable link. The projects, which the user selected, are shown in a compact overview with their thumbnail, name and the links to open it inside FabCenter or Thingiverse. Underneath the administrator learns about the needed device, the user's comment on the request, and is presented a selection box to choose one of the selected time slots. The administrator can get information by sending an e-mail to the user by clicking on the corresponding link if the situation is not clear enough. This link will tell the operating system to open a new e-mail with the already inserted address of the user. After accepting one of this options all parties will get an e-mail from the system about this finalized schedule. In case of a cancellation initiated from any of the parties the visit can alway be rescheduled, or just be reactivated as it was. At the top of the figure 3.19 we see a secondary menu

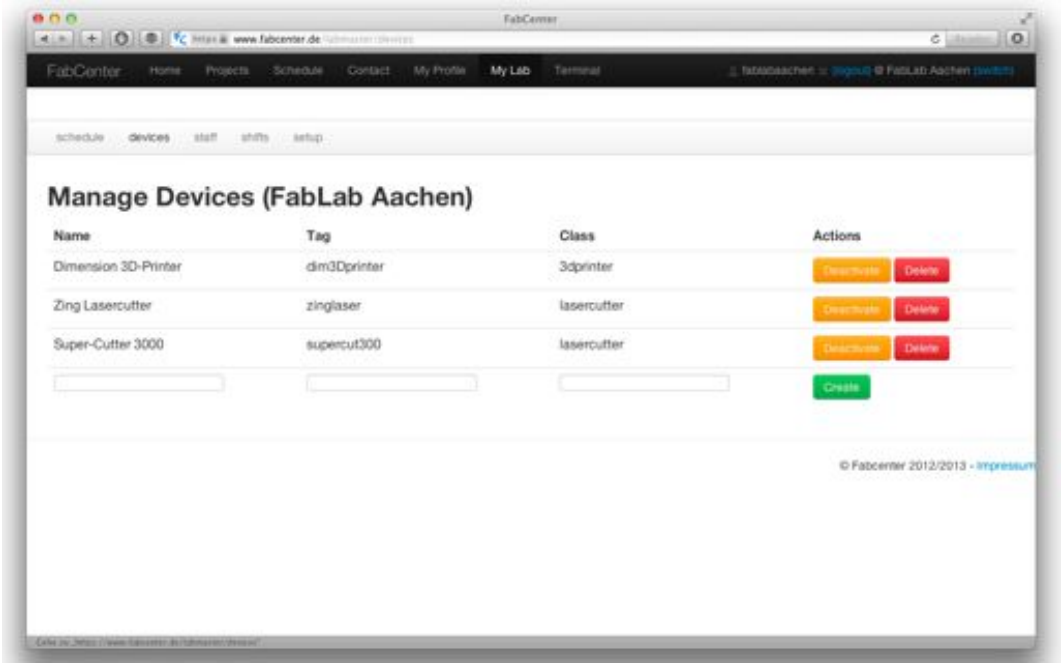


Figure 3.20

that guides the administrator to the other sections in the lab's section.

In the subsection "Devices" the administrator can manage all the lab's devices (see figure 3.20 for an overview). A list shows all currently available devices with their name, a tag for identification in the back end, the class of devices it is in and the set of actions that can be performed on each device. There is the possibility to deactivate or activate a device, to even delete a device from the list and one can add a new device to the list using the form in the last row of the table. If active, these devices are shown in the user's form in step one of the scheduling process. By reusing a tag of a deleted device, the new device may overtake the function of the old device in already schedule visits, since each visit is saved with the device's tag into the back end calendar.

After the devices are managed, the next subsection is about the list of staff members (see figure 3.21). Each added staff member is defined by a Thingiverse account, a clear name for representation, a tag for internal identification, and the list of devices the staff member may operate. This list is

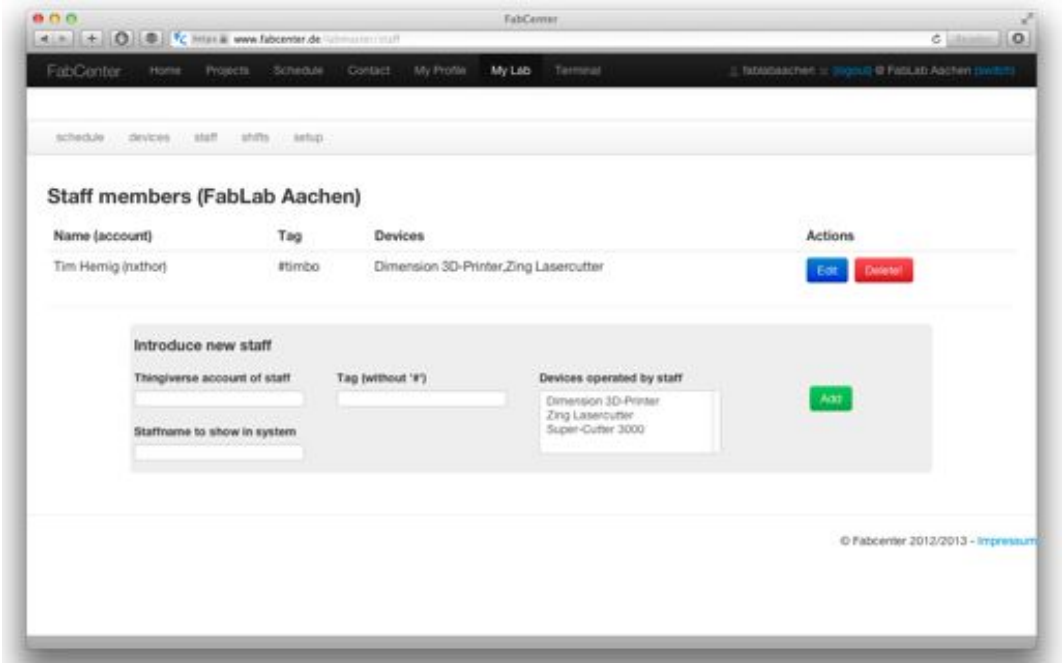


Figure 3.21

editable and each entry can also be deleted. Since the staff is assigned to shifts there is no need to deactivate a staff temporarily, the lab's administrator is able to facilitate this effect in the next subsection of the fablab's section.

All members of staff may be assigned to shifts in the fablab, which maybe public or private depending on the staff being there to take care of visitors or only for internal purpose. At the top of figure 3.22 we see a table of the reoccurring shifts ("General Shifts"). Each shift here has a start date and time, a end time and a weekday on which this shift is repeated on. The fist day may be a different weekday, but after that every selected weekday is also taken into account as a shift with the same daytimes for the staff to get on and off the shift.

Each shift is represented in the back end as an event with special tags in the description to save the setting like which device is operate-able and if the shift is public or not and active or not. Since the shifts are saved as events in the fablab's Google Calendar account, we cannot prevent interaction with these events from the outside. Therefore it

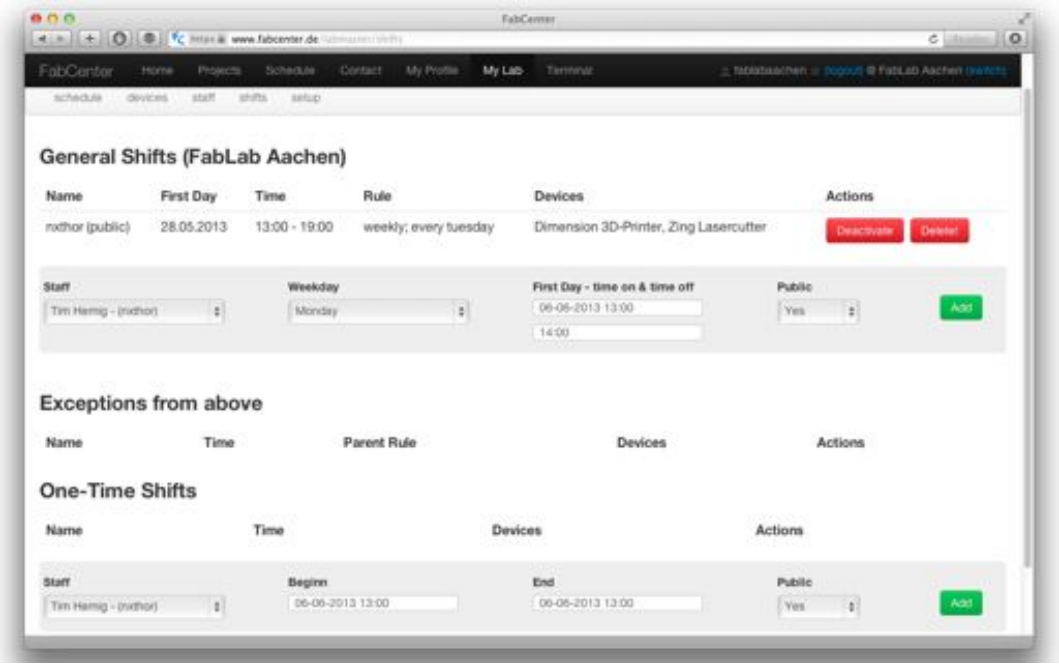


Figure 3.22

is possible that there are individual changes to some of the reoccurring events. All this exceptions are listed in the second table on this view if they are present in the back end. In general, moving shifts or visits to other points in time through a different way of access will not destroy the consistency of the schedule. All Data needed to interpret an event is coded in the event itself.

At the bottom of the view we see a list of all "One-Time Shifts". These are shifts that are not foreseen to reoccur. In this case the selection of the weekday to reoccur on is left out and start and end are defined by a full date and daytime information. In figure 3.22 the last two tables are empty in order to provide a compact overview. Each table would fill analog to the first one.

The last figure (see figure 3.23) in this subsection is about the last subsection in the lab administrator's section "My Lab" in the menu. The setup provides a list of setting to be edited to enable the lab to model the mode of operation. Also some informal settings can be made like the name, address and tag of the lab. This tag will be also be used to

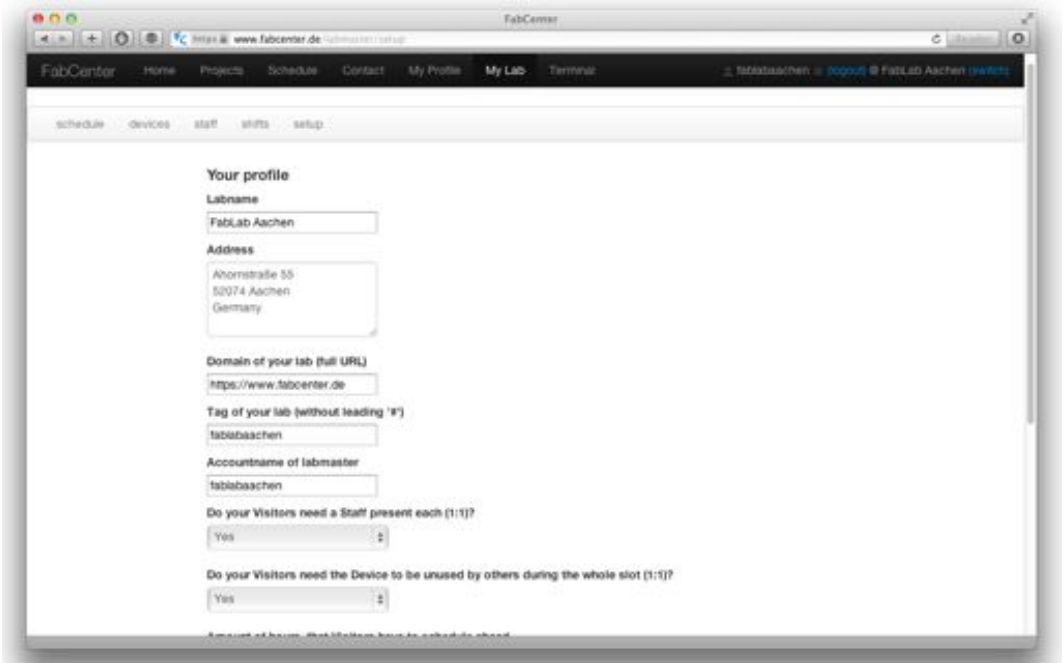


Figure 3.23

find projects connected to the lab on Thingiverse. Here the administrator can control the parameters of the search for available time slots in the scheduling process like whether each visitor will mark an available staff busy or not.

Fabmoment-feed

Feed of Fabmoments realized by dedicated controller

Finally one controller was implemented to provide a RSS-feed (see example in figure 3.24) as described in section 2.2.1.

All publicly available project data connected to one fablab indicated by the request is gathered, and translated into the RSS-format by a template. No access token of any user is used in this process. This creates a URL, which is callable by any system outside FabCenter. The administrator can see this URL in the lab's setup, as figure 3.25 shows.

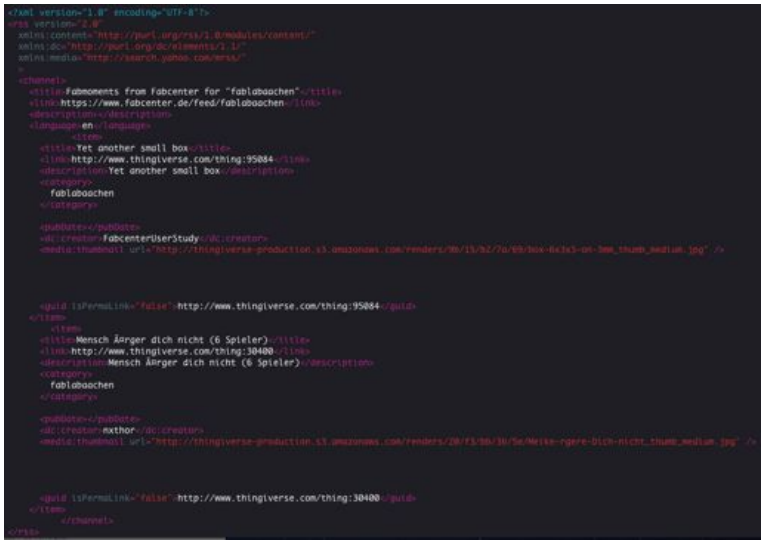


Figure 3.24: Example of a generated RSS-feed providing Fabmoments

Figure 3.25: URL shown to access Fabmoments via RSS-feed

3.7 Summary of Own Work

In this chapter we have set up our requirements for the system, have shown the early design phase of the system, discussed the issues of the redesign of the system and presented the used technology and the implementation of the FabCenter. To show that the requirements were met we will discuss the evaluation of the system in the next chapter.

Chapter 4

Evaluation

This chapter will cover the evaluation of the final implementation as described in chapter 3. We conducted two basic user studies with a few tasks for the users to execute with the help of FabCenter. One study covers the user-side of FabCenter and the other one covers the administrators-side. The two groups are disjunctive; anyone has participated in both studies. In the user-side study are 14 participants, and 10 in the administrator-side study. Overall we have 24 participants in two disjunctive groups to evaluate the usability of FabCenter in two different contexts. The tasks cover the general use cases the system does provide. Both studies include individual questions related to the tasks and the system in general. The Participant first got a rough explanation of the context and goals of the system. After that they were asked to perform the tasks without looking around inside the system first. The poller made notes about observations. Most participants on both sides of the study are students at the university in a technical related subject like engineering or technical communication and computer science.

24 participants in total.

Both studies contain 10 questions to gather data for a usability scale according to "System Usability Scale" by Brooke [1996]. Out of this data we will discus the quantitative part of the user study in section 4.1, present the results of the usability scale in section 4.2. Closing this chapter we will show in section 4.3 that FabCenter meets the require-

ments set in section 3.3.

The questionnaires can be found as figures F.1 to F.4 and F.5 to F.8 in appendix F. The questions are numbered and we will reference in most occasions to the numbers instead of quoting the whole question.

4.1 Quantitative User Study

First we want to roughly describe the participants. The mostly technical background of the groups was mentioned before. As figure 4.1 shows, the participants on the user-side are in the early twenties in a range from 20 to 30 years old.

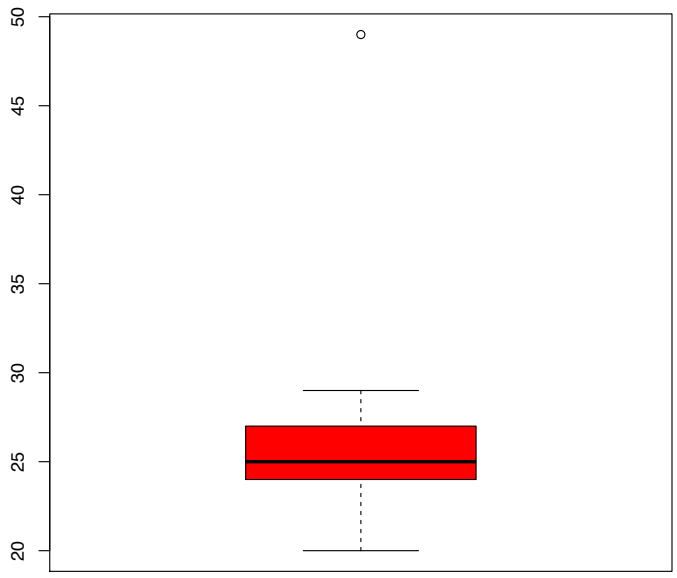


Figure 4.1: Age distribution of the participants in the user's study

Figure 4.2 shows that 86% of the participants in that group

are male, and the other 14% female.

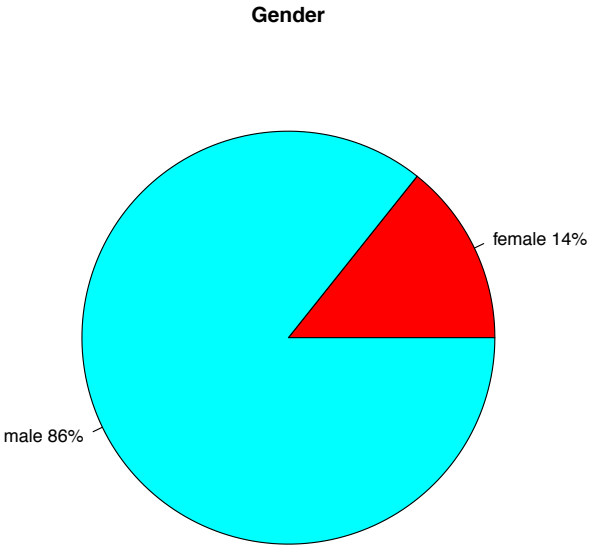


Figure 4.2: Gender distribution of the participants in the user's study

In the administrator-targeted study the participants are to 70% male and 30% female (see figure 4.4) and the age ranges in general from 23 to 27 with the median around 24 years (see figure 4.3.

In both studies the first three questions asked the participant for their familiarity with the topic of Personal Fabrication (Q a), fablabs (Q b) and Thingiverse (Q c). The range of "unknown" to "very familiar" is mapped into the values of one to five.

As figures 4.5 and 4.6 show, the participants on the user side are a quite familiar with Personal Fabrication and fablabs but the majority is less familiar with Thingiverse. On the administrator-side the familiarity is overall a bit lower, but Thingiverse is better known in this group.

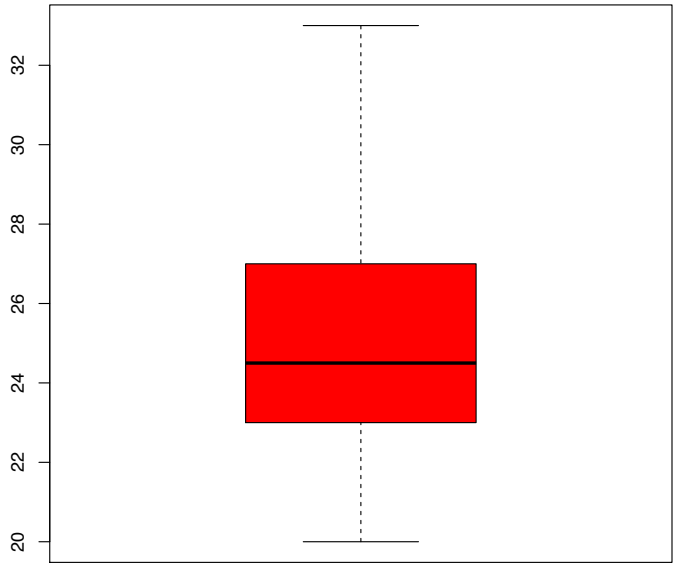


Figure 4.3: Age distribution of the participants in the administrator's study

All following questions were answered in a range from one to five, where one corespondents to "disagree" and five to "agree". Three would be the middle of the range and mean "undecided".

4.1.1 Task Related

Users can work with the system easily.

Figure 4.7 shows the results of the questions concerning the tasks performed by the participants. All users could easily log in (Q 01), find their projects (Q 02) and schedule a visit (Q 03). The following tasks using the terminal of FabCenter are in general agreed to be easy (Q 04). Sometimes the users showed confusion about the different terms of "login" vs. "check in". These terms will be merged as discussed in section 3.6, and the issue can be resolved in future version of FabCenter. The Users were able to successfully work with

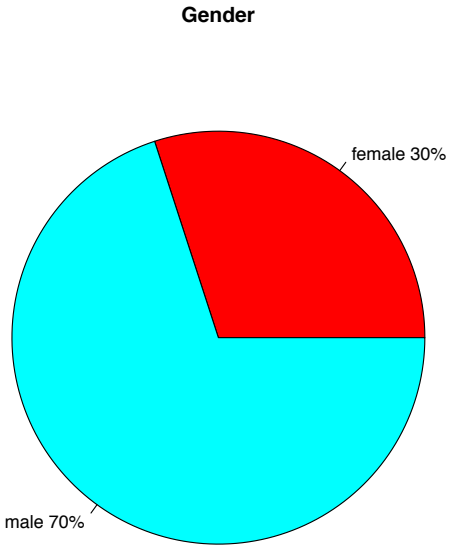


Figure 4.4: Gender distribution of the participants in the administrator's study

the desired file (Q 05-07). The majority thinks that documenting at the end of the visit is reasonable and that they would use the system again for this task (Q 08-09).

In figure 4.8 we see the results of the questions about the administrator-side of the tasks. We learn that logging in (Q 01), accepting users requests (Q 05) and canceling visits (Q 06) are clear seen as short and easy to perform tasks. Finding the lab's setup (Q 02) got sightly lower results. In most cases users found their own settings, before realizing that the lab' settings were meant. To raise the usability in future versions of FabCenter we should perhaps think about merging all settings into one section, and dividing them into subsections on a deeper level in the system's hierarchy. Once having seen the "My Lab"-section all management tasks (Q 03-6) were easily accomplished, and agreed to be easy (Q 04), simple (Q 05) and short (Q 06).

Easy administration of a lab.

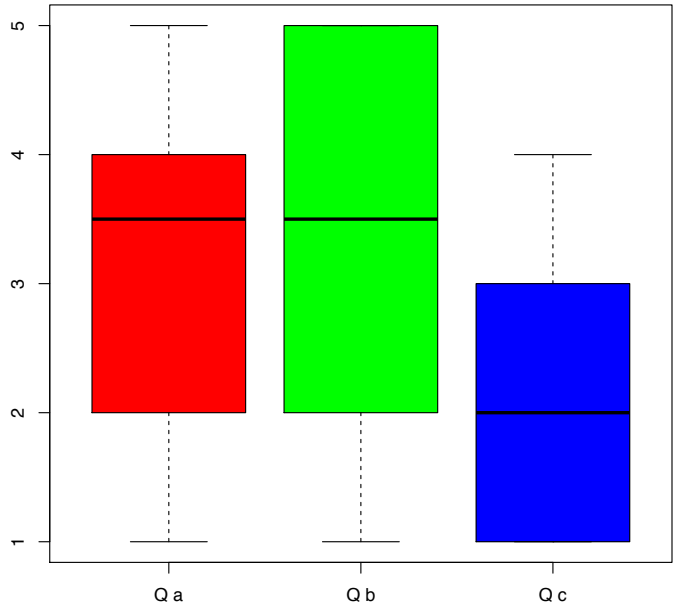


Figure 4.5: Box-and-whisker plot of the familiarity-questions on the user side

4.1.2 System Related

This section covers the part of both studies that is connected to the impression the system made at all. For the user-side figure 4.9 describes that the forms were good to use (Q 13), the documenting properties of the system does add value to the public (Q 14) and the user thinks that the system is overall satisfying the need for usability (Q 15). With slightly lower results, but still agreed by the majority are the statements about the system giving feedback to the interaction (Q 10). Showing how to reach a goal is fulfilled (Q 11), also most users see the value in documenting not only for the public, but also for themselves (Q 12).

On the administrator-side of this part of the study we see in figure 4.10 that managing the core features is accomplished in a very clear way (Q 09). The feedback is noticed (Q 07)

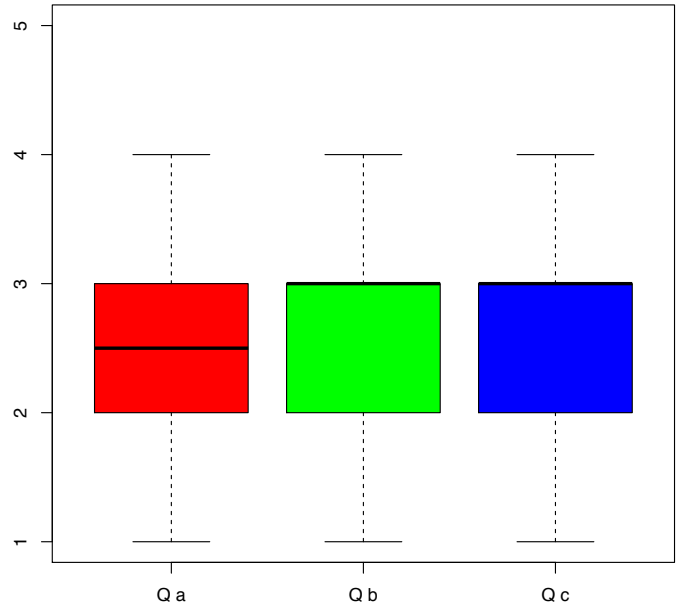


Figure 4.6: Box-and-whisker plot of the familiarity-questions on the administrator-side

and the system shows how to reach the goals of a task (Q 08). Overall the user is satisfied with the easy usage of the system.

4.2 SUS

The last part of the studies is designed according to the System Usability Scale by Brooke [1996]. On the user-side of the study this concerns the questions from Q 16 to Q 25 and on the administrator-side from Q 11 to Q 20.

Next to the box-and-whisker plot in figure 4.11 and 4.12 we also are able to calculate a SUS score, as a value out of one hundred to roughly compare the usability with other systems. Also comparable values for the comparison of us-

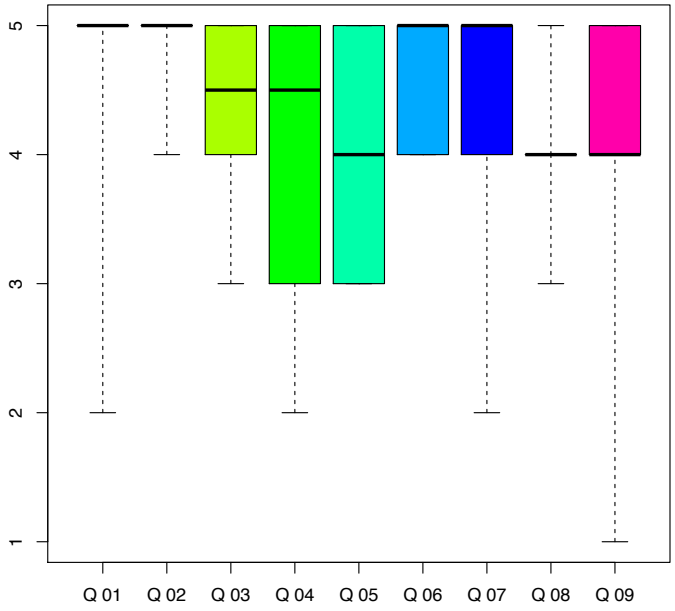


Figure 4.7: Box-and-whisker plot of the task-questions on the user-side

ability and learnability of the system are calculated by this method. In the appendix the figures F.9 and F.10 show the single participant's scores and the summary of each study. The final scores are also shown in table 4.1 and 4.2. The scores show that FabCenter has got a good (see section 5.1 of Bangor et al. [2008]) usability on the user-side, and a almost superior value on the administrator-side.

Users	SUS Score	Usability	Learnability
Mean	78,93	79,24	77,68
Standard deviation	17,94	17,62	23,60

Table 4.1: SUS results for the user-side study

The relative weakness of the user-side can be identified in figure 4.11. Q 19 and Q 20 cover the well integration of the functions and the need for technical support using the

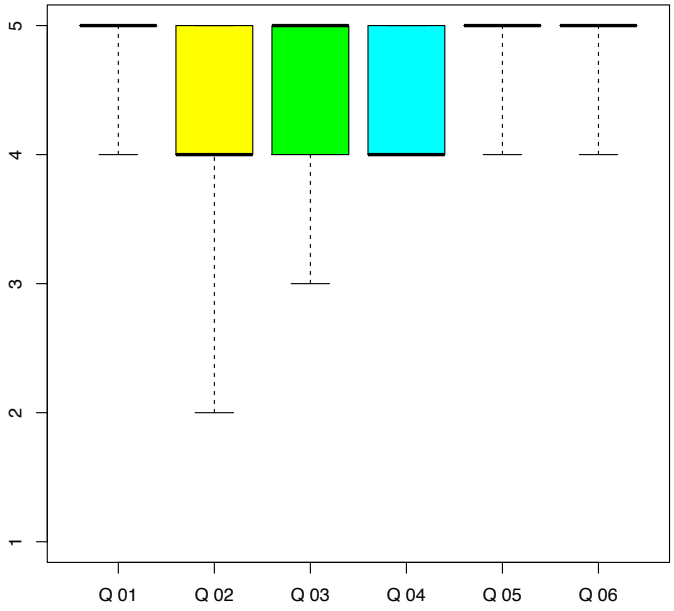


Figure 4.8: Box-and-whisker plot of the task-questions on the administrator-side

system. The observed confusion in section 4.1.1 may be the source for this high deviation.

4.3 Requirement Analysis

In this chapter we acquired results describing the state of FabCenter by evaluating the system. We now can compare the accomplishments to the requirements set in section 3.3 and show that all requirements were met.

All requirements were met.

We will discuss the single elements of the two lists:

- M1&M2: The support of the documentation process and sharing the results with the Internet is imple-

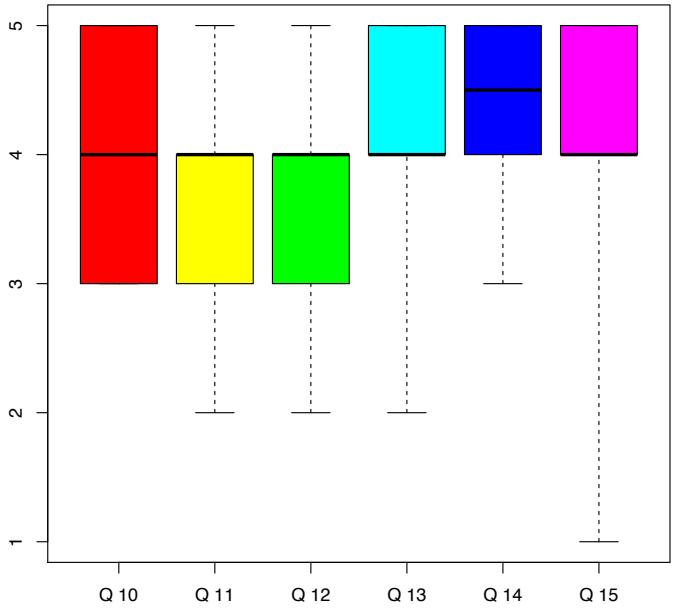


Figure 4.9: Box-and-whisker plot of the system-questions on the user-side

mented into the system, and the user values its function (See section 4.1.2). Section 3.6 described the documenting function, and section 4.1.1 proved that the user was able to perform this task.

- M3: All results in section 4.1 and 4.2 show that the presented implementation in section 3.6 show that organizing internal processes of fablabs is supported by the system and the user can easily perform tasks in this part of the system.
- M4: FabCenter offers a usable system. The System Usability Scale used in section 4.2 and the results from the system related questions in section 4.1 validate this accomplishment.

- R1: The presented implementation of section 3.6 enables the user to browse his projects from Thingi-

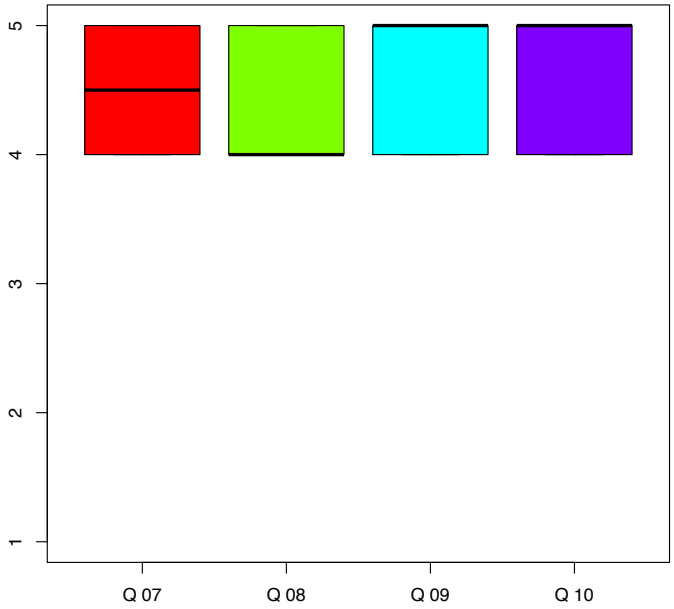


Figure 4.10: Box-and-whisker plot of the system-questions on the administrator-side

verse. An internal view is provided, and the project information is used to schedule visits. Also the documentation is put back into Thingiverse. The requirement of Thingiverse-integration is met.

- R2: As mentioned above, the scheduling is part of the functionality of FabCenter, and was evaluated to be usable.

- R3: The "check out"-functionality presented in section 3.6 guides the user into documenting his visit. The evaluation of the system in section 4.1.1 and 4.1.2 shows that the user is able to perform this task and values the process.

- R4: Each fablab is able to register with FabCenter, and the system supports different fablabs as shown in the implementation of the user's profile in section 3.6.

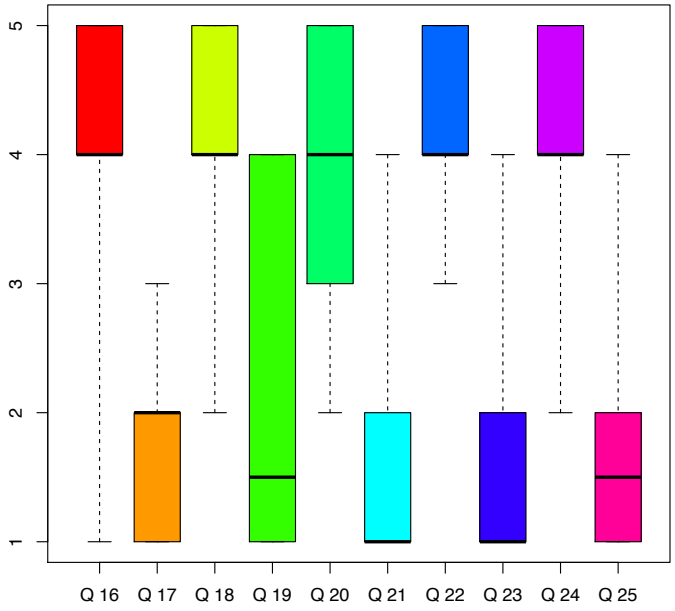


Figure 4.11: Box-and-whisker plot of the SUS-questions on the user-side

- R5: The tasks covered by the quantitative user study in section 4.1 show that the administrator of a fablab is able to manage staff and devices.
- R6: Section 3.6 shows that a feed of all the lab's projects are provided in form of a RSS-feed following the model of Fabmoments as described in section 2.2.1.

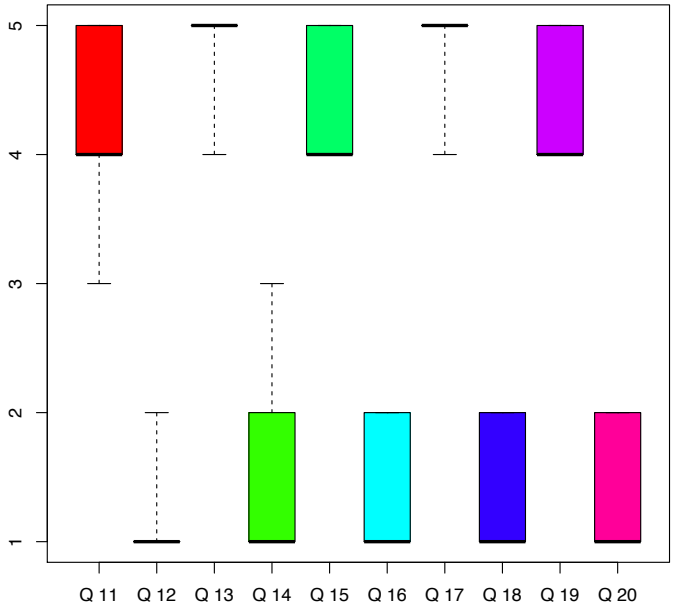


Figure 4.12: Box-and-whisker plot of the SUS-questions on the administrator-side

Administrators	SUS Score	Usability	Learnability
Mean	90,00	89,69	91,25
Standard deviation	4,08	4,18	10,29

Table 4.2: SUS results for the administrator-side study

Chapter 5

Summary and future work

5.1 Summary and contributions

In this thesis we learned about the current possible ways to share and present digital designs over the Internet. We identified different kinds of fablabs regarding the operation modes and management of visiting users.

In FabCenter we created a easy to use tool to share documentation and support administrators of fablabs

After changes in the landscape of Personal Fabrication by gaining access to a huge preexisting user base, we had to analyze the platform and redesign it to react on the latest developments. We designed a system to support users and administrators in the task of organizing, creating and sharing documentation in the context of fablabs. The System was intended to provide existing sources of projects with a back-channel for documentation. Also the System needed to support the administrators of fablabs to manage the lab's schedule with the visitors of the lab.

Redesign of FabCenter was needed

We set up requirements out of the learned lessons of the related work and implemented a web-based application using current Internet technologies. This application provides a mechanism to get the users of a fablab to remember and execute the step of publishing the documentation just created. Also users are able to easily schedule visits with their

FabCenter was implemented based on current technology

local fablab. This is accomplished by reducing the barriers of this process through the help of a functional and easy to use website.

User studies showed success of the implementation

The accomplishment of meeting the requirements was verified by a user study including a System Usability Scale to compare the quality of the system. The system proved to be usable and functional.

FabCenter enriches the quality and quantity of documentation

This thesis shows that FabCenter is able to contribute to the community of Personal Fabrication by giving opportunity to feed back created knowledge and experience while building prototypes out of digital designs.

5.2 Future work

Extending FabCenter would improve the quality of the system

Despite the accomplishments, FabCenter still has got minor issues, as we learned in section 4.1 and 4.2. Some of the known issues can be resolved in a few iterations of development.

System extendable to support multiple services.

Also, we concentrated on supporting one API of a remote service in each case, where we needed an enabling platform to get the users projects from, or the connection to the user's and administrator's infrastructure. Adding more support for different third party APIs would certainly enlarge the potential user base and the acceptance by the users. Developers would be able to create services that could be also used by FabCenter to enrich the networking and sharing of documentation through the Internet.

Internal processes with a long runtime may be optimized

Some procedures, which retrieve a lot of atomic data set from the different remote services, have a potential to be optimized in the further development of the system. Calls to the APIs are a general bottleneck of most algorithms in FabCenter.

More iterations of development on the system would raise the quality of the user experience

As mentioned in the discussion of the "Terminal" interface in section 3.6 the terms of login or logout and check in or check out need to be merged properly. Another approach would be to overcome the technical difficulties we

encountered in that matter. This requires the API of the involve services to evolve, too. Also the user interface could be evolved by a more creative design to improve the look and feel of the system.

Longterm study would show more insights

The conducted user studies were limited in time, such that a longterm study with FabCenter in operation over a duration of six or more months would raise more information on the actual performance of the system. Also the correlation between Thingiverse and FabCenter are a topic for a longterm study. Will users be motivated to create a Thingiverse account, or will the acceptance rate significantly be raised by an internal account-system or rolling out more integration of other services?

Fusion with Fabiji to provide a more complete service

With Fabiji we presented a documentation creating tool in section 2.2.2. FabCenter would gain completeness if a system to create the documentation to be shared is strongly connected if not fully integrated into the system.

Appendix A

Fablab Charter

The Fab Charter

What is a fab lab?

Fab labs are a global network of local labs, enabling invention by providing access to tools for digital fabrication

What's in a fab lab?

Fab labs share an evolving inventory of core capabilities to make (almost) anything, allowing people and projects to be shared

What does the fab lab network provide?

Operational, educational, technical, financial, and logistical assistance beyond what's available within one lab

Who can use a fab lab?

Fab labs are available as a community resource, offering open access for individuals as well as scheduled access for programs

What are your responsibilities?

safety: not hurting people or machines
operations: assisting with cleaning, maintaining, and improving the lab
knowledge: contributing to documentation and instruction

Who owns fab lab inventions?

Designs and processes developed in fab labs can be protected and sold however an inventor chooses, but should remain available for individuals to use and learn from

How can businesses use a fab lab?

Commercial activities can be prototyped and incubated in a fab lab, but they must not conflict with other uses, they should grow beyond rather than within the lab, and they are expected to benefit the inventors, labs, and networks that contribute to their success

draft: October 20, 2012

Figure A.1: Fablab charter, draft from 2012

Appendix B

Open Hardware

CERN OPEN HARDWARE LICENCE v1.1

Preamble

Through this CERN Open Hardware Licence ("CERN OHL") version 1.1, the Organization wishes to disseminate its hardware designs (as published on http://www.ohwr.org/) as widely as possible, and generally to foster collaboration among public research hardware designers.

The CERN OHL is copyright of CERN. Anyone is welcome to use the CERN OHL, in unmodified form only, for the distribution of his own Open Hardware designs. Any other right is reserved.

1. Definitions

In this Licence, the following terms have the following meanings:

"Licence" means this CERN OHL.

"Documentation" means schematic diagrams, designs, circuit or circuit board layouts, mechanical drawings, flow charts and descriptive text, and other explanatory material that is explicitly stated as being made available under the conditions of this Licence. The Documentation may be in any medium, including but not limited to computer files and representations on paper, film, or any other media.

"Product" means either an entire, or any part of a, device built using the Documentation or the modified Documentation.

"Licensee" means any natural or legal person exercising rights under this Licence.

"Licensor" means any natural or legal person that creates or modifies Documentation and subsequently communicates to the public and/ or distributes the resulting Documentation under the terms and conditions of this Licence.

A Licensee may at the same time be a Licensor, and vice versa.

2. Applicability

2.1 This Licence governs the use, copying, modification, communication to the public and distribution of the Documentation, and the manufacture and distribution of Products. By exercising any right granted under this Licence, the Licensee irrevocably accepts these terms and conditions.

2.2 This Licence is granted by the Licensor directly to the Licensee, and shall apply worldwide and without limitation in time. The Licensee may assign his licence rights or grant sub-licences.

2.3 This Licence does not apply to software, firmware, or code loaded into programmable devices which may be used in conjunction with the Documentation, the modified Documentation or with Products. The use of such software, firmware, or code is subject to the applicable licence terms and conditions.

Figure B.1: Page one of three of the CERN Open Hardware License v1.1

3. Copying, modification, communication to the public and distribution of the Documentation

3.1 The Licensee shall keep intact all copyright and trademarks notices and all notices that refer to this Licence and to the disclaimer of warranties that is included in the Documentation. He shall include a copy thereof in every copy of the Documentation or, as the case may be, modified Documentation, that he communicates to the public or distributes.

3.2 The Licensee may use, copy, communicate to the public and distribute verbatim copies of the Documentation, in any medium, subject to the requirements specified in section 3.1.

3.3 The Licensee may modify the Documentation or any portion thereof. The Licensee may communicate to the public and distribute the modified Documentation (thereby in addition to being a Licensee also becoming a Licensor), always provided that he shall:

a. comply with section 3.1;
b. cause the modified Documentation to carry prominent notices stating that the Licensee has modified the Documentation, with the date and details of the modifications;
c. license the modified Documentation under the terms and conditions of this Licence or, where applicable, a later version of this Licence as may be issued by CERN; and
d. send a copy of the modified Documentation to all Licensors that contributed to the parts of the Documentation that were modified, as well as to any other Licensor who has requested to receive a copy of the modified Documentation and has provided a means of contact with the Documentation.

3.4 The Licence includes a licence to those patents or registered designs that are held by the Licensor, to the extent necessary to make use of the rights granted under this Licence. The scope of this section 3.4 shall be strictly limited to the parts of the Documentation or modified Documentation created by the Licensor.

4. Manufacture and distribution of Products

4.1 The Licensee may manufacture or distribute Products always provided that the Licensee distributes to each recipient of such Products a copy of the Documentation or modified Documentation, as applicable, and complies with section 3.

4.2 The Licensee is invited to inform in writing any Licensor who has indicated its wish to receive this information about the type, quantity and dates of production of Products the Licensee has (had) manufactured.

5. Warranty and liability

5.1 DISCLAIMER – The Documentation and any modified Documentation are provided "as is" and any express or implied warranties, including, but not limited to, implied warranties of merchantability, of satisfactory quality, and fitness for a particular purpose or use are disclaimed in respect of the Documentation, the modified Documentation or any Product. The Licensor makes no representation that the Documentation, modified Documentation, or any Product, does or will not infringe any patent, copyright, trade secret or other proprietary right. The entire risk as to the use, quality, and performance of a Product shall be with the Licensee and not the Licensor. This disclaimer of warranty is an essential part of this Licence and a

Figure B.2: Page two of three of the CERN Open Hardware License v1.1

condition for the grant of any rights granted under this Licence. The Licensee warrants that it does not act in a consumer capacity.

5.2 LIMITATION OF LIABILITY – The Licensor shall have no liability for direct, indirect, special, incidental, consequential, exemplary, punitive or other damages of any character including, without limitation, procurement of substitute goods or services, loss of use, data or profits, or business interruption, however caused and on any theory of contract, warranty, tort (including negligence), product liability or otherwise, arising in any way in relation to the Documentation, modified Documentation and/or the use, manufacture or distribution of a Product, even if advised of the possibility of such damages, and the Licensee shall hold the Licensor(s) free and harmless from any liability, costs, damages, fees and expenses, including claims by third parties, in relation to such use.

6. General

6.1 The rights granted under this Licence do not imply or represent any transfer or assignment of intellectual property rights to the Licensee.

6.2 The Licensee shall not use or make reference to any of the names, acronyms, images or logos under which the Licensor is known, save in so far as required to comply with section 3. Any such permitted use or reference shall be factual and shall in no event suggest any kind of endorsement by the Licensor or its personnel of the modified Documentation or any Product, or any kind of implication by the Licensor or its personnel in the preparation of the modified Documentation or Product.

6.3 CERN may publish updated versions of this Licence which retain the same general provisions as this version, but differ in detail so far this is required and reasonable. New versions will be published with a unique version number.

6.4 This Licence shall terminate with immediate effect, upon written notice and without involvement of a court if the Licensee fails to comply with any of its terms and conditions, or if the Licensee initiates legal action against Licensor in relation to this Licence. Section 5 shall continue to apply.

6.5 Except as may be otherwise agreed with the Intergovernmental Organization, any dispute with respect to this Licence involving an Intergovernmental Organization shall, by virtue of the latter's Intergovernmental status, be settled by international arbitration. The arbitration proceedings shall be held at the place where the Intergovernmental Organization has its seat. The arbitral award shall be final and binding upon the parties, who hereby expressly agree to renounce any form of appeal or revision.

Figure B.3: Page three of three of the CERN Open Hardware License v1.1

Appendix C

Initial Questionnaire

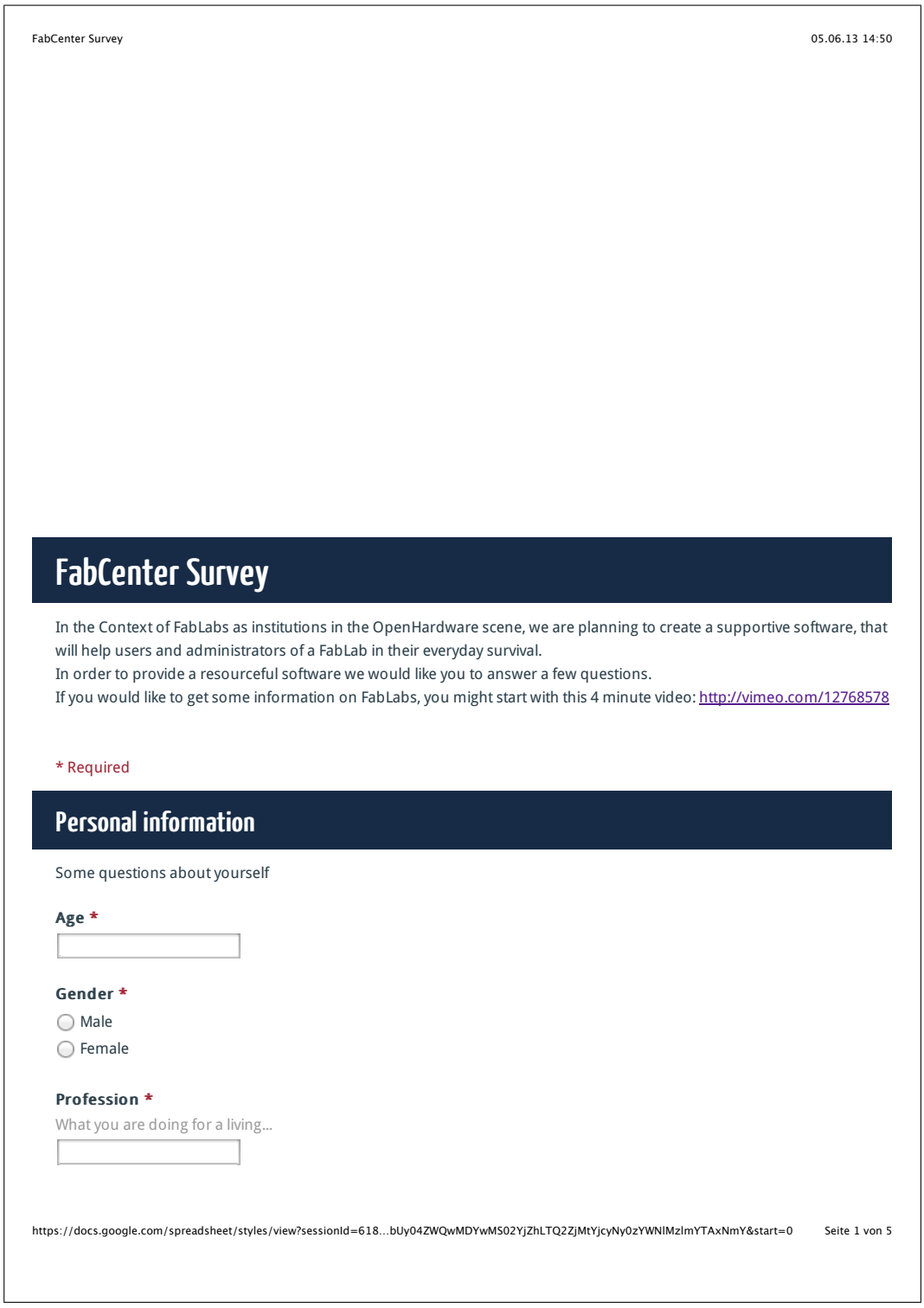


Figure C.1: Initial Questionnaire Page one of five

FabCenter Survey 05.06.13 14:50

I have an idea about what a FabLab does. *

- [ ] yes
- [ ] no

About your projects

Are you a DIY-person?
Please pick from 1 to 5 depending on how correct the statement is to you.
'3' would mean, that you are undecided.

I have projects where I build physical objects that are not standard shaped *

You build things, but the parts won't show up in any store.

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

I use my own products, they endure and have a purpose. *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

I use platforms like thingiverse.com to share my projects. *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

I use platforms like thingiverse just to have my project-files in the cloud. *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

I like to put my project plan into the web to have others comment on it, or use it for their own work. *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

You and the FabLab

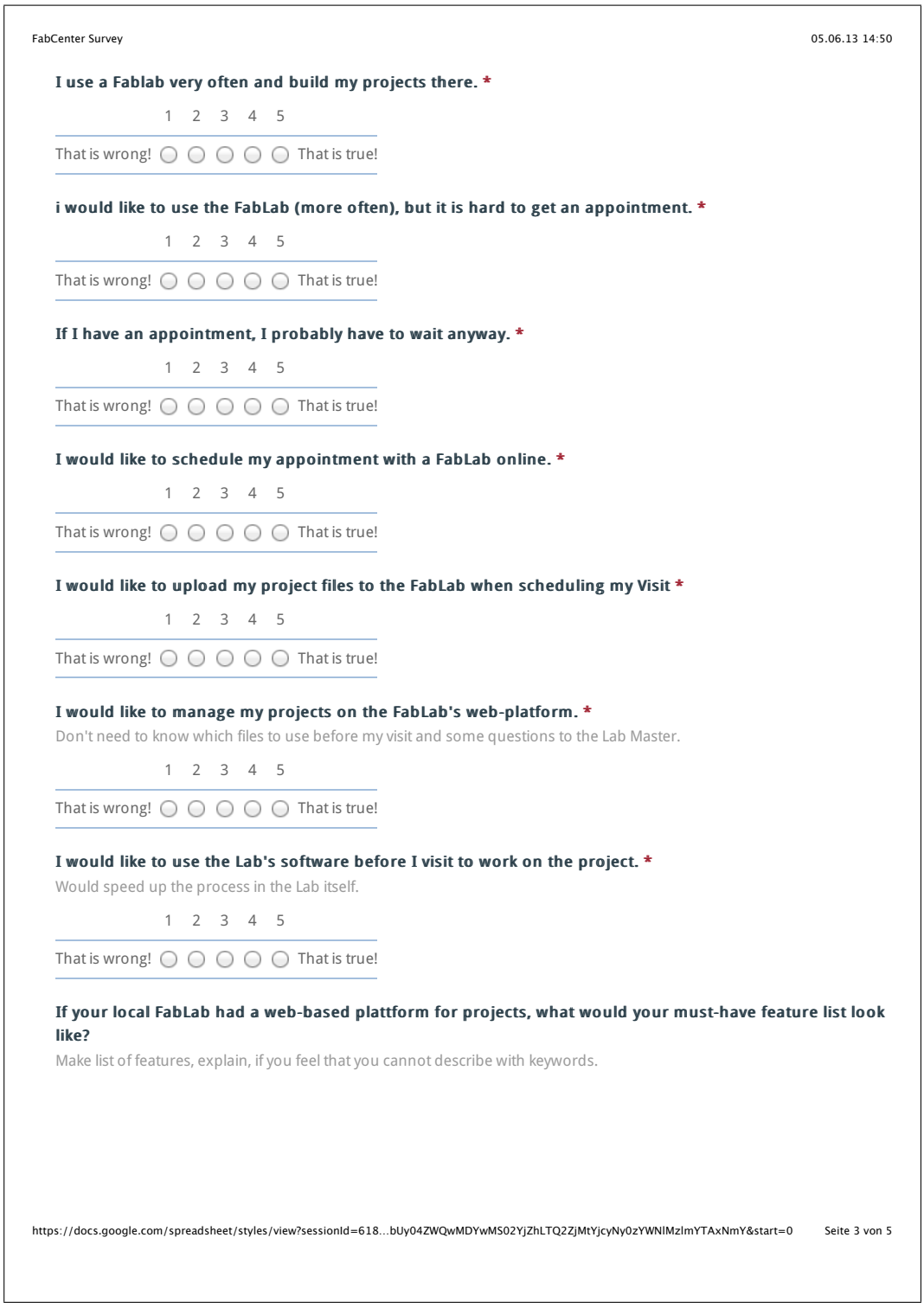
I would use a web platform like thingiverse of my local FabLab if they had one of their own. *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

https://docs.google.com/spreadsheet/styles/view?sessionId=618...bUy04ZWQwMDYwMS02YjZhLTQ2ZjMtYjcyNy0zYWNlMzlmYTAxNmY&start=0 Seite 2 von 5

Figure C.2: Initial Questionnaire Page two of five



FabCenter Survey 05.06.13 14:50

I use a Fablab very often and build my projects there. *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

i would like to use the FabLab (more often), but it is hard to get an appointment. *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

If I have an appointment, I probably have to wait anyway. *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

I would like to schedule my appointment with a FabLab online. *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

I would like to upload my project files to the FabLab when scheduling my Visit *

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

I would like to manage my projects on the FabLab's web-platform. *

Don't need to know which files to use before my visit and some questions to the Lab Master.

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

I would like to use the Lab's software before I visit to work on the project. *

Would speed up the process in the Lab itself.

1 2 3 4 5

That is wrong! ○ ○ ○ ○ ○ That is true!

If your local FabLab had a web-based plattform for projects, what would your must-have feature list look like?

Make list of features, explain, if you feel that you cannot describe with keywords.

https://docs.google.com/spreadsheet/styles/view?sessionId=618...bUy04ZWQwMDYwMS02YjZhLTQ2ZjMtYjcyNy0zYWNlMzlmYTAxNmY&start=0 Seite 3 von 5

Figure C.3: Initial Questionnaire Page three of five

I do run my own FabLab and have visitors at open Lab days. *

I am a FabLab Master

- yes
- no

About FabLab Masters

You run your own FabLab? So we would like to ask some additional questions.

We have open days for visitors to use our Lab for their projects. *

1 2 3 4 5

That is wrong! That is true!

Our schedule is full all the time and visitors have to wait, because the needed device is in use. *

1 2 3 4 5

That is wrong! That is true!

Visitors have to leave undone, because neither he/she or the Lab had the necessary quantity of the materi in stock. *

"We are out of wood."

1 2 3 4 5

That is wrong! That is true!

Visitors have problems to get started on their visit. They need help to get with their files to the right application on the right computer in order to use the right device *

"Where do I go? Which computer should I use?"

1 2 3 4 5

That is wrong! That is true!

Visitors often ask about the features of our software before they visit. If only they could play around with beforehand. *

"Ah, that's what the software is able to do!"

Figure C.4: Initial Questionnaire Page four of five

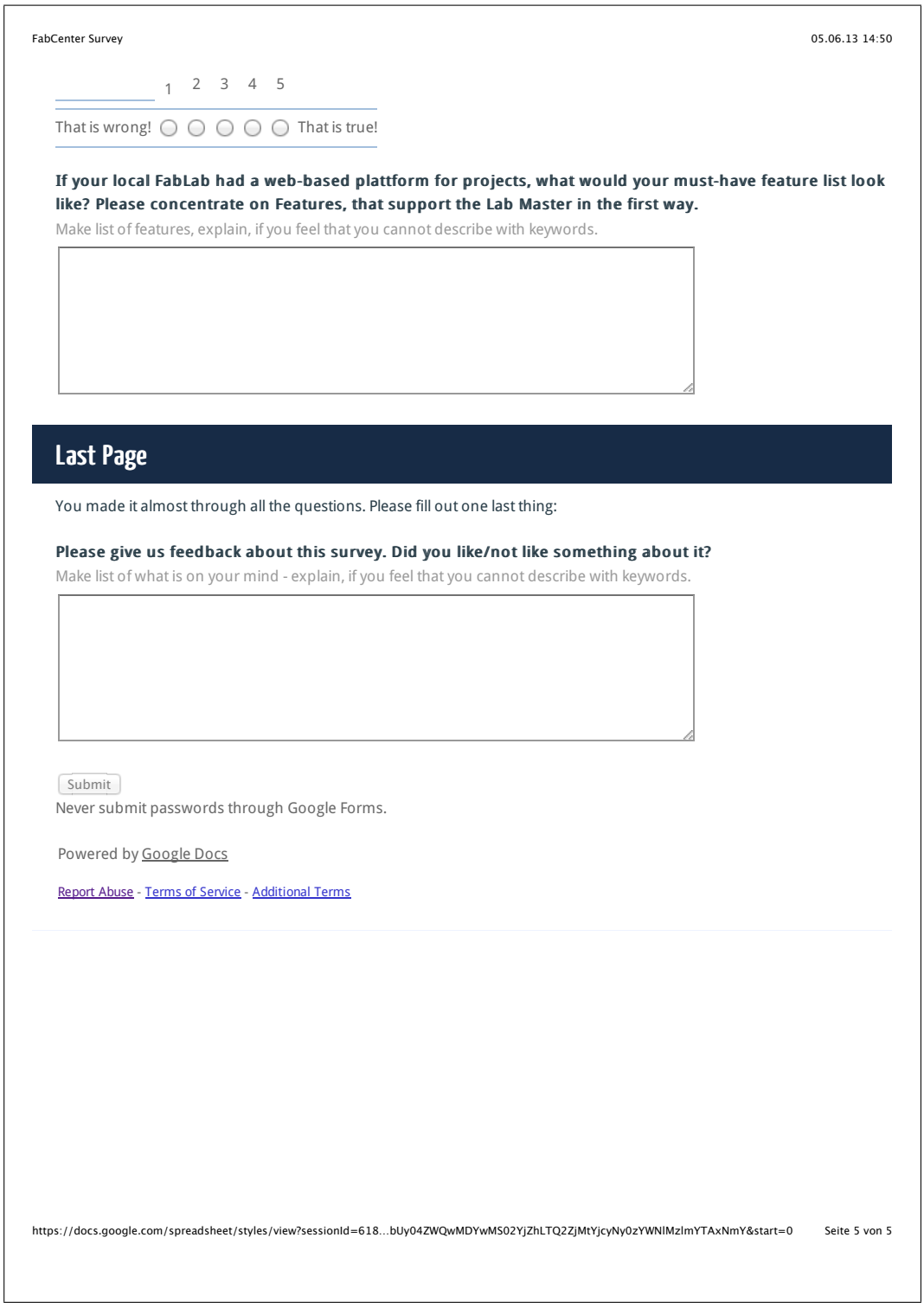
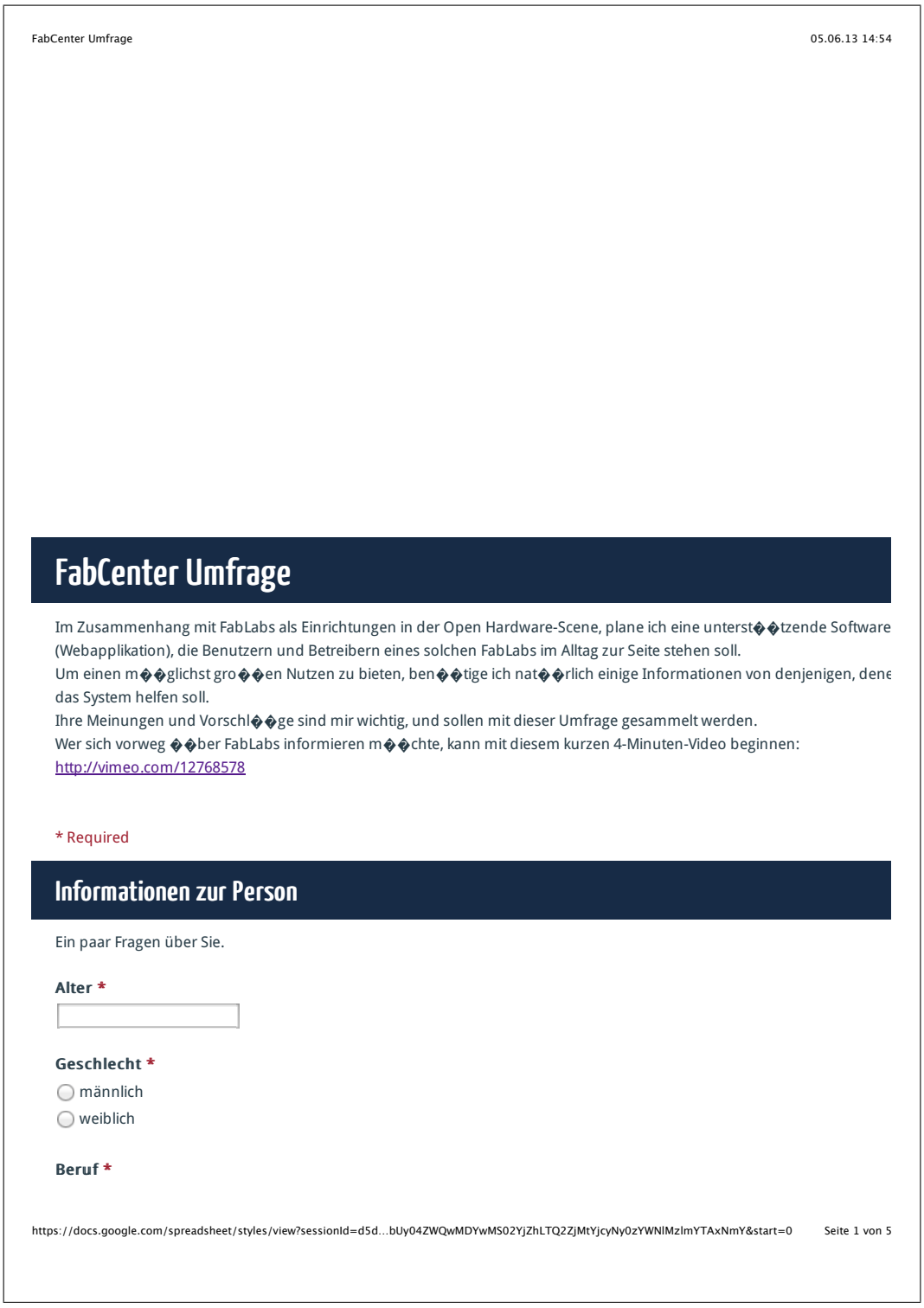


Figure C.5: Initial Questionnaire Page five of five



FabCenter Umfrage 05.06.13 14:54

FabCenter Umfrage

Im Zusammenhang mit FabLabs als Einrichtungen in der Open Hardware-Scene, plane ich eine unterst��tzende Software (Webapplikation), die Benutzern und Betreibern eines solchen FabLabs im Alltag zur Seite stehen soll.
Um einen m��glichst gro��en Nutzen zu bieten, ben��tige ich nat��rlich einige Informationen von denjenigen, dene das System helfen soll.
Ihre Meinungen und Vorschl��ge sind mir wichtig, und sollen mit dieser Umfrage gesammelt werden.
Wer sich vorweg ��ber FabLabs informieren m��chte, kann mit diesem kurzen 4-Minuten-Video beginnen:
http://vimeo.com/12768578

* Required

Informationen zur Person

Ein paar Fragen über Sie.

Alter *

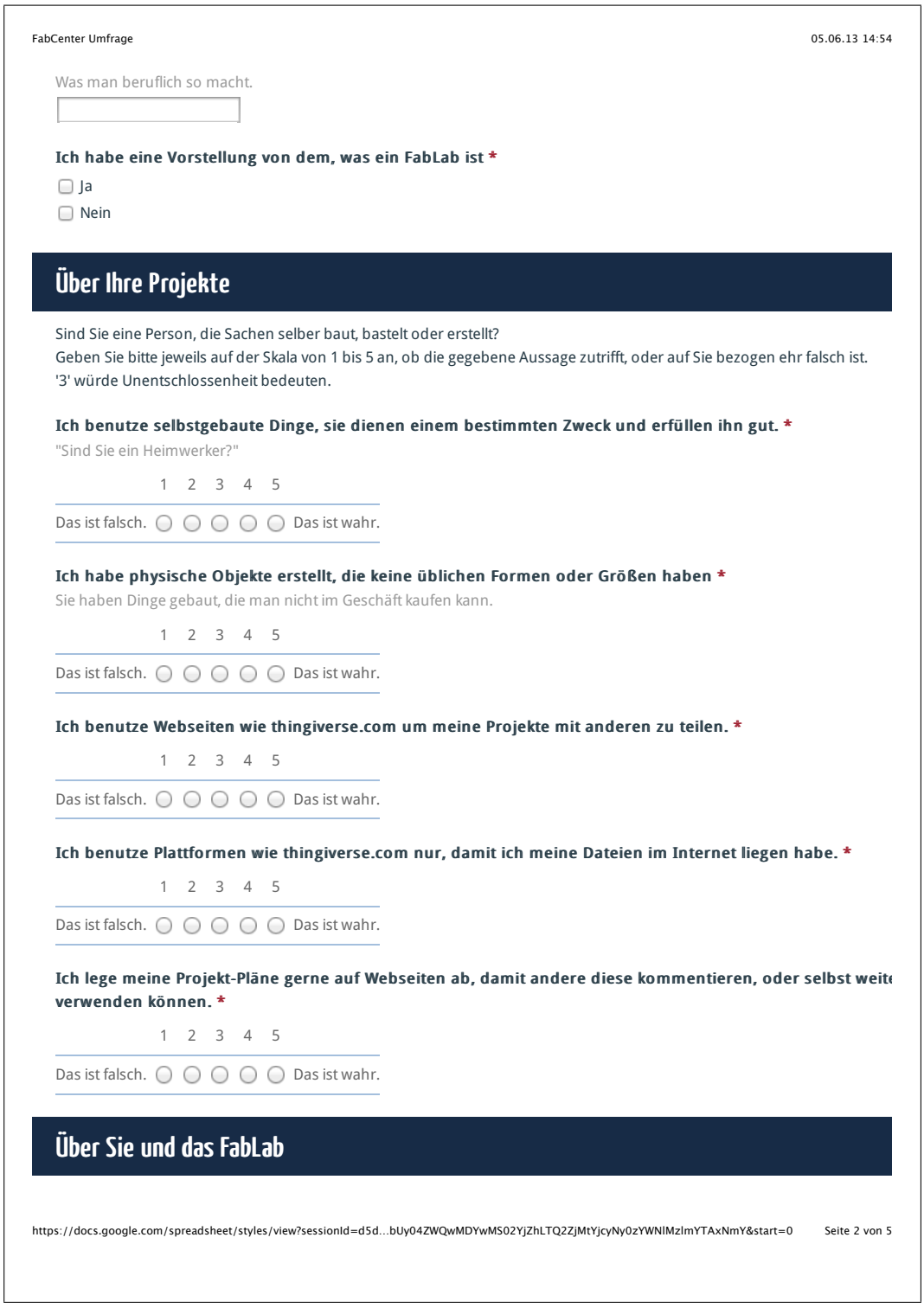
Geschlecht *

- ○ männlich
- ○ weiblich

Beruf *

https://docs.google.com/spreadsheet/styles/view?sessionId=d5d...bUy04ZWQwMDYwMS02YjZhLTQ2ZjMtYjcyNy0zYWNlMzlmYTAxNmY&start=0 Seite 1 von 5

Figure C.6: Initial Questionnaire Page one of five



FabCenter Umfrage 05.06.13 14:54

Was man beruflich so macht.

Ich habe eine Vorstellung von dem, was ein FabLab ist *

- [ ] Ja
- [ ] Nein

Über Ihre Projekte

Sind Sie eine Person, die Sachen selber baut, bastelt oder erstellt?
Geben Sie bitte jeweils auf der Skala von 1 bis 5 an, ob die gegebene Aussage zutrifft, oder auf Sie bezogen ehr falsch ist.
'3' würde Unentschlossenheit bedeuten.

Ich benutze selbstgebaute Dinge, sie dienen einem bestimmten Zweck und erfüllen ihn gut. *
"Sind Sie ein Heimwerker?"

	1	2	3	4	5	
Das ist falsch.	○	○	○	○	○	Das ist wahr.

Ich habe physische Objekte erstellt, die keine üblichen Formen oder Größen haben *
Sie haben Dinge gebaut, die man nicht im Geschäft kaufen kann.

	1	2	3	4	5	
Das ist falsch.	○	○	○	○	○	Das ist wahr.

Ich benutze Webseiten wie thingiverse.com um meine Projekte mit anderen zu teilen. *

	1	2	3	4	5	
Das ist falsch.	○	○	○	○	○	Das ist wahr.

Ich benutze Plattformen wie thingiverse.com nur, damit ich meine Dateien im Internet liegen habe. *

	1	2	3	4	5	
Das ist falsch.	○	○	○	○	○	Das ist wahr.

Ich lege meine Projekt-Pläne gerne auf Webseiten ab, damit andere diese kommentieren, oder selbst weite verwenden können. *

	1	2	3	4	5	
Das ist falsch.	○	○	○	○	○	Das ist wahr.

Über Sie und das FabLab

https://docs.google.com/spreadsheet/styles/view?sessionId=d5d...bUy04ZWQwMDYwMS02YjZhLTQ2ZjMtYjcyNy0zYWNlMzlmYTAxNmY&start=0 Seite 2 von 5

Figure C.7: Initial Questionnaire Page two of five

FabCenter Umfrage 05.06.13 14:54

Ich würde eine Plattform wie thingiverse.com nutzen, wenn mein örtliches FabLab eine eigene davon hätt* *

1 2 3 4 5

Das ist falsch. ○ ○ ○ ○ ○ Das ist wahr.

Ich nutze bereits ein FabLab, und verwirkliche meine eigenen Projekte dort. *

1 2 3 4 5

Das ist falsch. ○ ○ ○ ○ ○ Das ist wahr.

Ich würde ein FabLab (öfter) nutzen, aber es ist so schwer einen Termin zu bekommen. *

1 2 3 4 5

Das ist falsch. ○ ○ ○ ○ ○ Das ist wahr.

Falls ich einen Termin habe, muss ich öfters warten. *

1 2 3 4 5

Das ist falsch. ○ ○ ○ ○ ○ Das ist wahr.

Ich möchte gerne meinen Termin im FabLab online buchen. *

1 2 3 4 5

Das ist falsch. ○ ○ ○ ○ ○ Das ist wahr.

Ich würde gerne meine Projekt-Dateien zum FabLab hoch laden, wenn ich einen Termin ausmache. *

1 2 3 4 5

Das ist falsch. ○ ○ ○ ○ ○ Das ist wahr.

Ich würde gerne meine Projekte auf der Webseite des FabLabs organisieren. *

Dann brauche ich nicht vorher schon genau zu wissen, welche Dateien ich beim Termin benötige, und kann mich z.B. noch dem Betreiber absprechen.

1 2 3 4 5

Das ist falsch. ○ ○ ○ ○ ○ Das ist wahr.

Ich würde gerne die Software aus dem FabLab schon vorher benutzen können. *

Dann würde ich vor Ort nicht so lange brauchen.

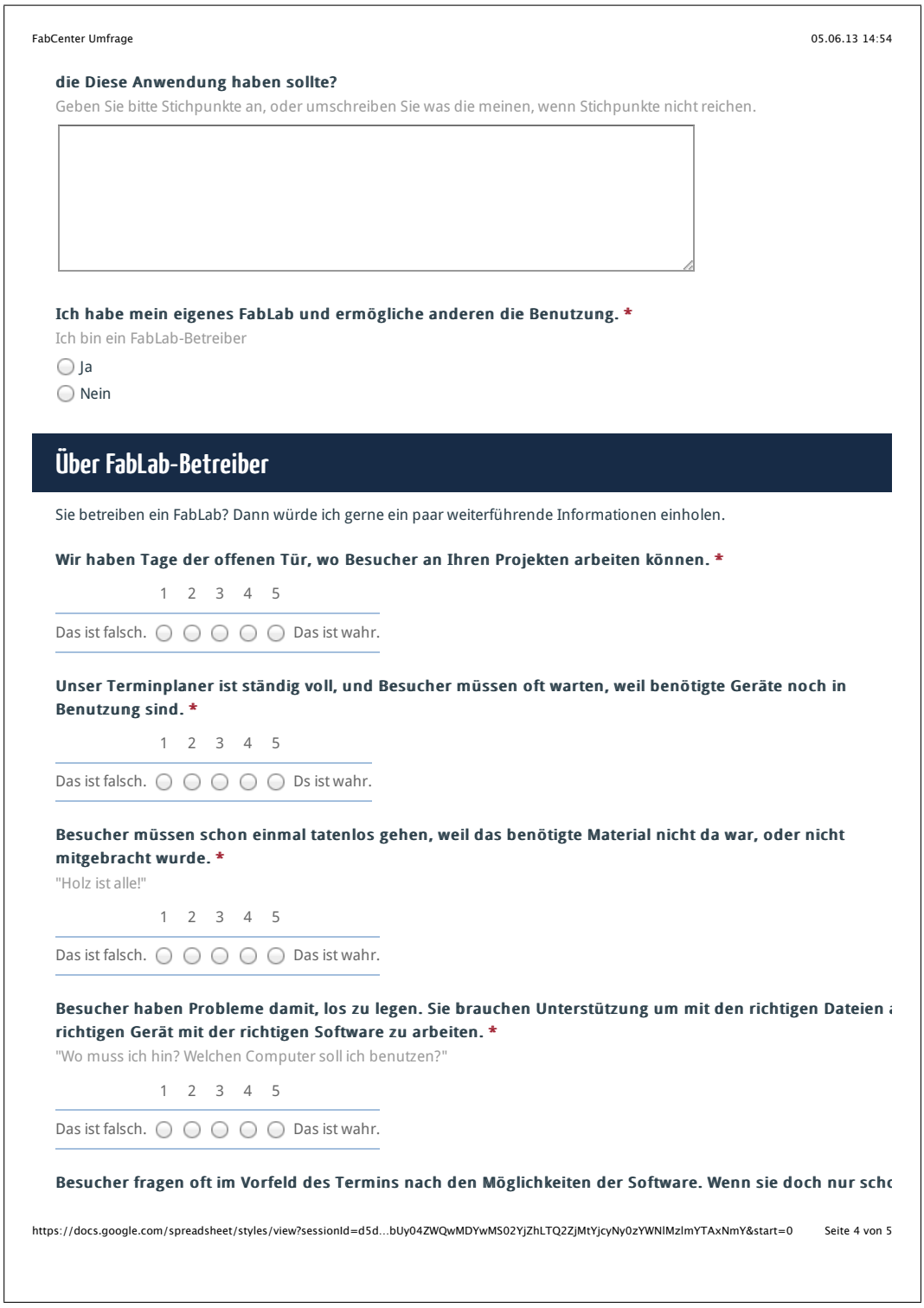
1 2 3 4 5

Das ist falsch. ○ ○ ○ ○ ○ Das ist wahr.

Wenn Ihr lokales FabLab eine Webseite zur Projektorganisation hätte, was wären die wichtigsten Funktion*

https://docs.google.com/spreadsheet/styles/view?sessionId=d5d...bUy04ZWQwMDYwMS02YjZhLTQ2ZjMtYjcyNy0zYWNlMzlmYTAxNmY&start=0 Seite 3 von 5

Figure C.8: Initial Questionnaire Page three of five



FabCenter Umfrage 05.06.13 14:54

die Diese Anwendung haben sollte?

Geben Sie bitte Stichpunkte an, oder umschreiben Sie was die meinen, wenn Stichpunkte nicht reichen.

Ich habe mein eigenes FabLab und ermögliche anderen die Benutzung. *

Ich bin ein FabLab-Betreiber

- Ja
- Nein

Über FabLab-Betreiber

Sie betreiben ein FabLab? Dann würde ich gerne ein paar weiterführende Informationen einholen.

Wir haben Tage der offenen Tür, wo Besucher an Ihren Projekten arbeiten können. *

	1	2	3	4	5	
Das ist falsch.	○	○	○	○	○	Das ist wahr.

Unser Terminplaner ist ständig voll, und Besucher müssen oft warten, weil benötigte Geräte noch in Benutzung sind. *

	1	2	3	4	5	
Das ist falsch.	○	○	○	○	○	Ds ist wahr.

Besucher müssen schon einmal tatenlos gehen, weil das benötigte Material nicht da war, oder nicht mitgebracht wurde. *

"Holz ist alle!"

	1	2	3	4	5	
Das ist falsch.	○	○	○	○	○	Das ist wahr.

Besucher haben Probleme damit, los zu legen. Sie brauchen Unterstützung um mit den richtigen Dateien a richtigen Gerät mit der richtigen Software zu arbeiten. *

"Wo muss ich hin? Welchen Computer soll ich benutzen?"

	1	2	3	4	5	
Das ist falsch.	○	○	○	○	○	Das ist wahr.

Besucher fragen oft im Vorfeld des Termins nach den Möglichkeiten der Software. Wenn sie doch nur scho

https://docs.google.com/spreadsheet/styles/view?sessionId=d5d...bUy04ZWQwMDYwMS02YjZhLTQ2ZjMtYjcyNy0zYWNlMzlmYTAxNmY&start=0 Seite 4 von 5

Figure C.9: Initial Questionnaire Page four of five


FabCenter Umfrage 05.06.13 14:54

vorher mit der Software herum probieren könnten. *

"Ach, so das geht ja mit der Software viel einfacher!"

1 2 3 4 5

Das ist falsch. ○ ○ ○ ○ ○ Das ist wahr.

Wenn Ihr lokales FabLab eine Webseite zur Projektorganisation hätte, was wären die wichtigsten Funktione die Diese Anwendung für Betreiber bereithalten sollte?

Geben Sie bitte Stichpunkte an, oder umschreiben Sie was die meinen, wenn Stichpunkte nicht reichen.

Die letze Seite

Sie haben es fast durch alle Fragen geschafft! Zuletzt würde ich gerne noch eine Sache wissen:

Bitte geben Sie an, ob Ihnen etwas an diese Umfrage gefallen hat, oder nicht. ich bin für jede Kritik offen und dankbar.

Notieren Sie einfach Stichwörter die Ihnen in den Kopf schießen, oder umschreiben den Punkt.

Submit

Never submit passwords through Google Forms.

Powered by Google Docs

Report Abuse - Terms of Service - Additional Terms

https://docs.google.com/spreadsheet/styles/view?sessionId=d5d...bUy04ZWQwMDYwMS02YjZhLTQ2ZjMtYjcyNy0zYWNlMzlmYTAxNmY&start=0 Seite 5 von 5

Figure C.10: Initial Questionnaire Page five of five

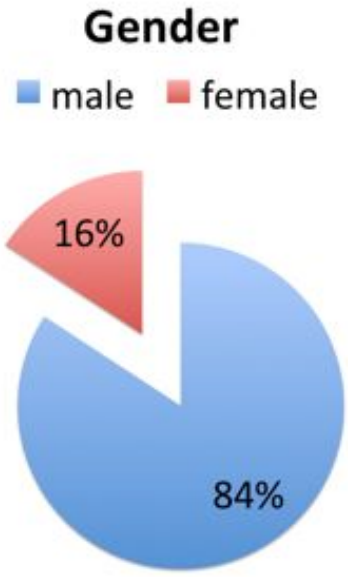


Figure C.11: Gender distribution of participants in questionnaire

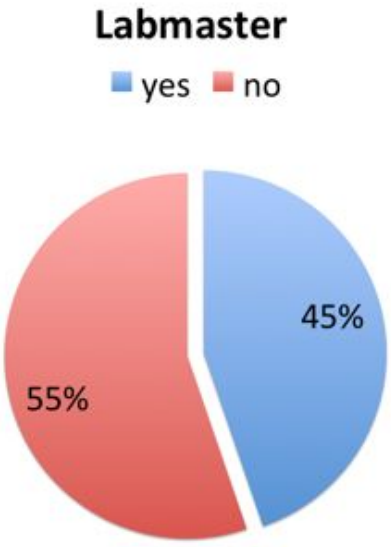


Figure C.12: Rate of lab administrators in questionnaire

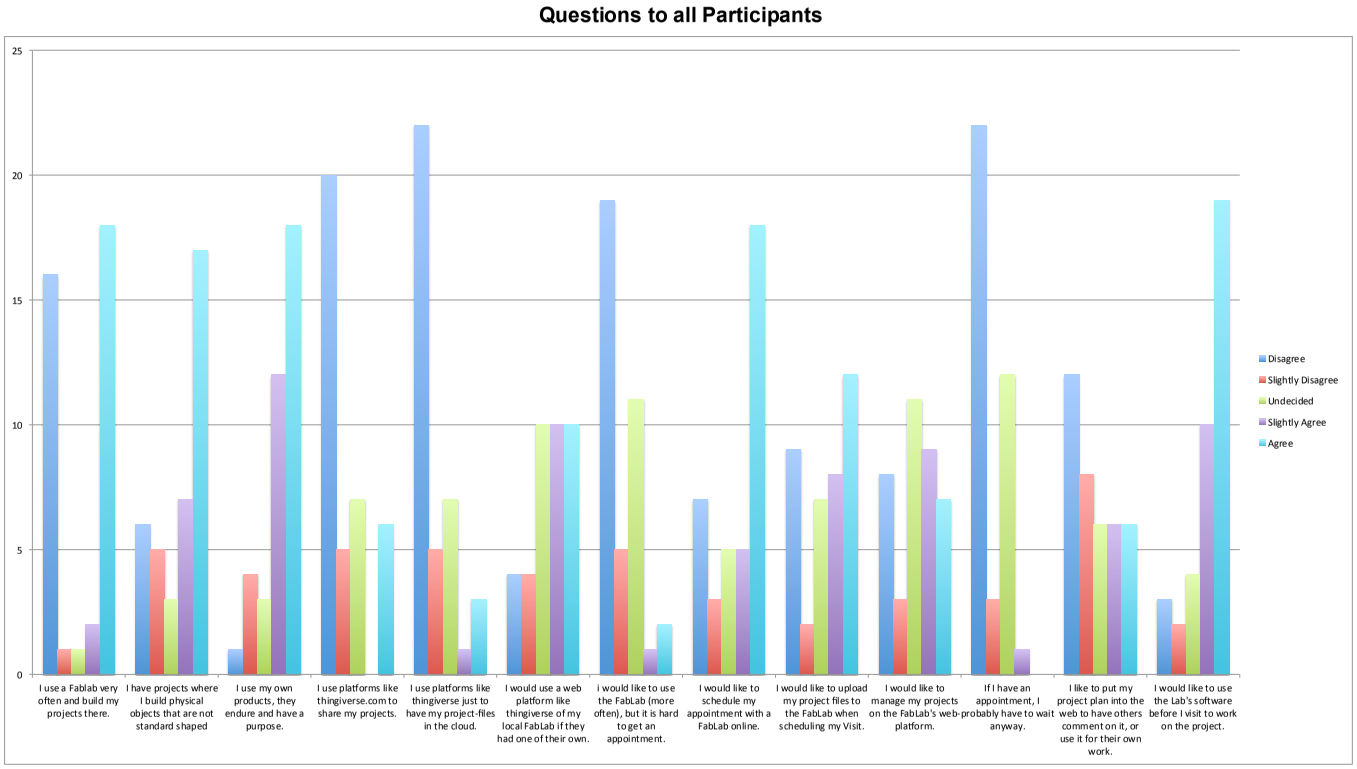


Figure C.13: Questions from initial questionnaire concerning all participants

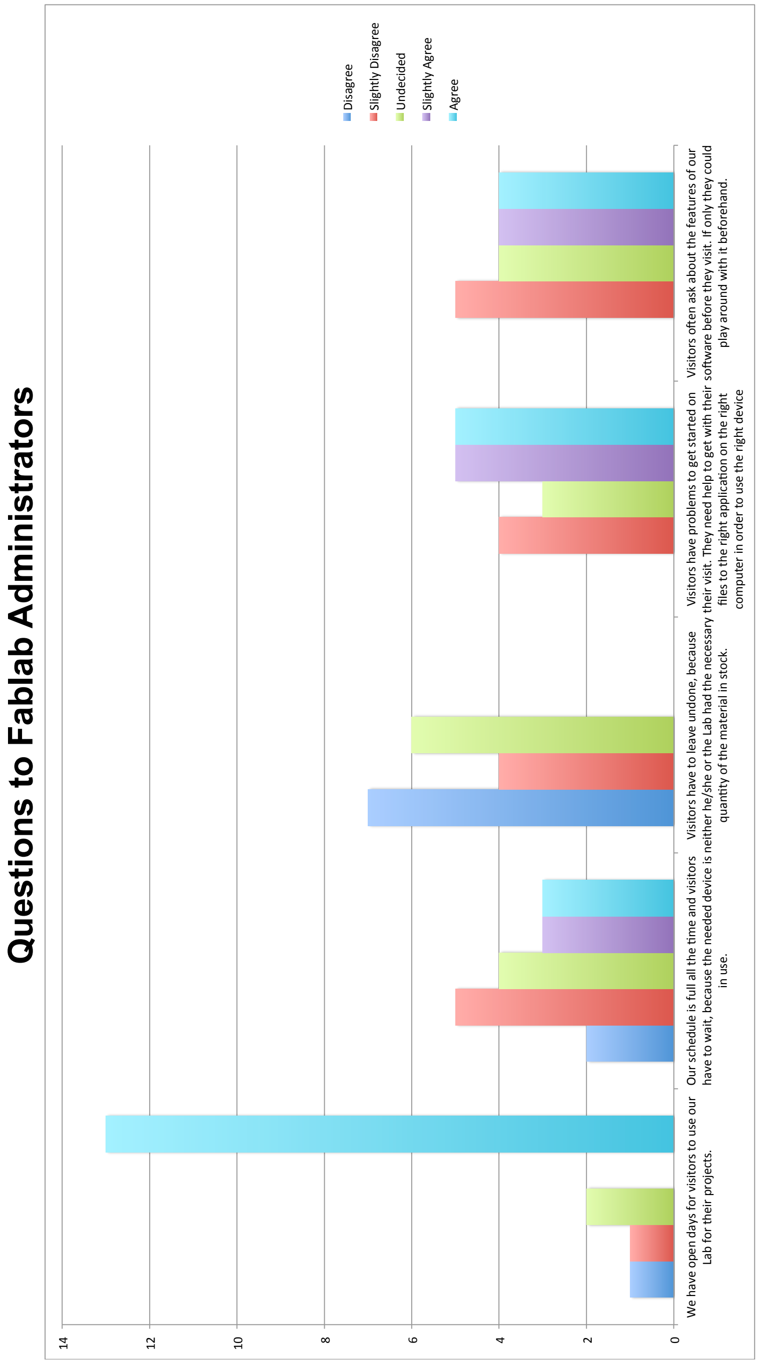


Figure C.14: Questions from initial questionnaire concerning only administrators

If your local FabLab had a web-based plattform for projects, what would your must-have feature list look like?

- speichern, teilen, viewen, versenden (an Geräte)
- unkompliziertes Hochladen von Dateien (ftp oÄ ggf mit sync)
- Teilen von Projekten, verschieben von Rechten an den projekten (wenns jemand übernehmen will)
- Unterstützung von verschiedenen Projekten, Auswahl von Schwerpunkten (klein/groß, Gewicht,...)
- Zeichenptogramn, ordnerverwaltung, tabellenkaljulationstool
- möglichst viel Vorbereitung von zu Hause aus möglich
- gute Anleitung und Hinweise
- vorgegebene Pläne, die ich selbst einfach verändern kann (ohne Einarbeitung in komplizierte Software)
- possibility to save the design files, pictures of the object, instructions how was it done
- possibility to search all the designs
- search with different keywords
- visual categorization (you may necessarily want to use a design for other purposes than what they had been designed)
- possibility to save the designs privately (you may not always want to share your work)
- Terminmanagment, Kostenrechner, geschätzte Dauer
- Dateien, Fotos, Text zur Beschreibung, Lizenz, verwendete Werkzeuge
- Communityfunktion: Ünterstützung von gemeinsamen Projekten
- Dateiablage mit QuickViewer im Browser
- Wiki-/Blog-Page zur öffentlichen Darstellung mit Fotos und beschreibenden Texten
- gute Freigabemöglichkeiten (privat, für Lab-Nutzer, für Öffentlichkeit)
- Versionsverwaltung
- Forks ermöglichen --> Forks zu neuem Projekt umwandeln mit Hinweis "inspired by" oder "based on"
- Integration in thingiverse o.ä. wünschenswert --> privat im Fablab starten, dann Checkbox für "Publish on thingiverse.com" setzen --> Projekt erscheint im Fablab-Thingiverse-Channel
- Photo, plan, possibility of video
- Free as in libre
- Accesible for everybody
- Easy to use
- Good previews
- Open, reliable, free, distributed
- 3D and 2D CAD programs
- Software that is available to common fab lab equipment for self-learning. Example: Partworks for Shopbot CNC Router.
- Professional looking format without too much clutter.
- File sharing
- Time allocation
- Discussion or commenting
- Easy documentation plateforme (or how to upload 3 pictures and a short description plus links) Most people have hard time knowing how to document project. Think about it. Normal people never document what they do in real life so they have no structure when it come to it. It need to be a great UI with easy to answer questions that gives a structure to the docuementation
- A mechanism to sharing across labs, to search across labs (even if they use completely different platforms).
- A way to find other projects depending on materials, machines, processes, machine-material-combinations, design questions, engineering questions, components used, software used
- A way to pull my project from the platform directly to the CAD/machine/IDE I'm using ... ideally across multiple labs (as in showing a video on youtoube on my PC, your phone, their tablet: mill a piece on my shopbot, your Modela, their CNC...)

Figure C.15: Raw collection of wanted features from the users

If your local FabLab had a web-based plattform for projects, what would your must-have feature list look like? Please concentrate on Features, that support the Lab Master in the first way.

- Öffnungszeiten/mögliche Termine eintragen die dann "gebucht" werden können.
- Wordpress-Integration zur Öffentlichkeitsarbeit
- Hervorhebung von Projekten
- Interface führt den User zu "guter" Dokumentation: gute Bilder, gute Texte, sauberes Layout
- clustering, contest, stats, visitor list, daily stats
- Free (libre)
- Preferably self hosted, if possible connected with other sites like Diaspora
- Easy upload (if possible some kind of git repository so I could connect it to SparkleShare for example)
- Stimulating people to share accesible files like .svg instead of .ai for example.
- Based on / extension / remix functionality to build upon other people's projects
- Tags
- Comments
- Guided documentation (multiple fields like short description, bill of materials, build instructions, what went wrong, what can be improved etc).
- See fabmoments of protospace.nl and/or fabpublications of Fablab Amersfoort.
- Nice to have: easy way to upload photos, using for example Eye-fi sd cards.
- All files uploaded should be available for preview and possibly proofing before visits, depening on the visitors desire/needs.
- A sign-in page for members. Logging hours and days is critical to demonstrating the participation in fab labs for funding and grants.
- See previous list.
- Also, organized grouping of photos, CAD drawings, shopping cart?, donation tab to be used for promoting new projects.
- Difficult to say. Someone with a Arduino related project have different needs than someone only using the 3d Printer to print her boyfriend face.. Booking time slots and documenting project is über important for lab staff and users and should be a priority. We currently uses a Forum, a Wiki, our blog and a Gallery for these puropose.
- Calculation / estimation of machine time (that's a MUST MUST MUST) Skill-tracker and e-learning so novice users can be identified, prepared beforehand and signalled to the lab manager / steward on arrival

Figure C.16: Raw collection of wanted features from the administrators

Appendix D

Brainstorming session with experts

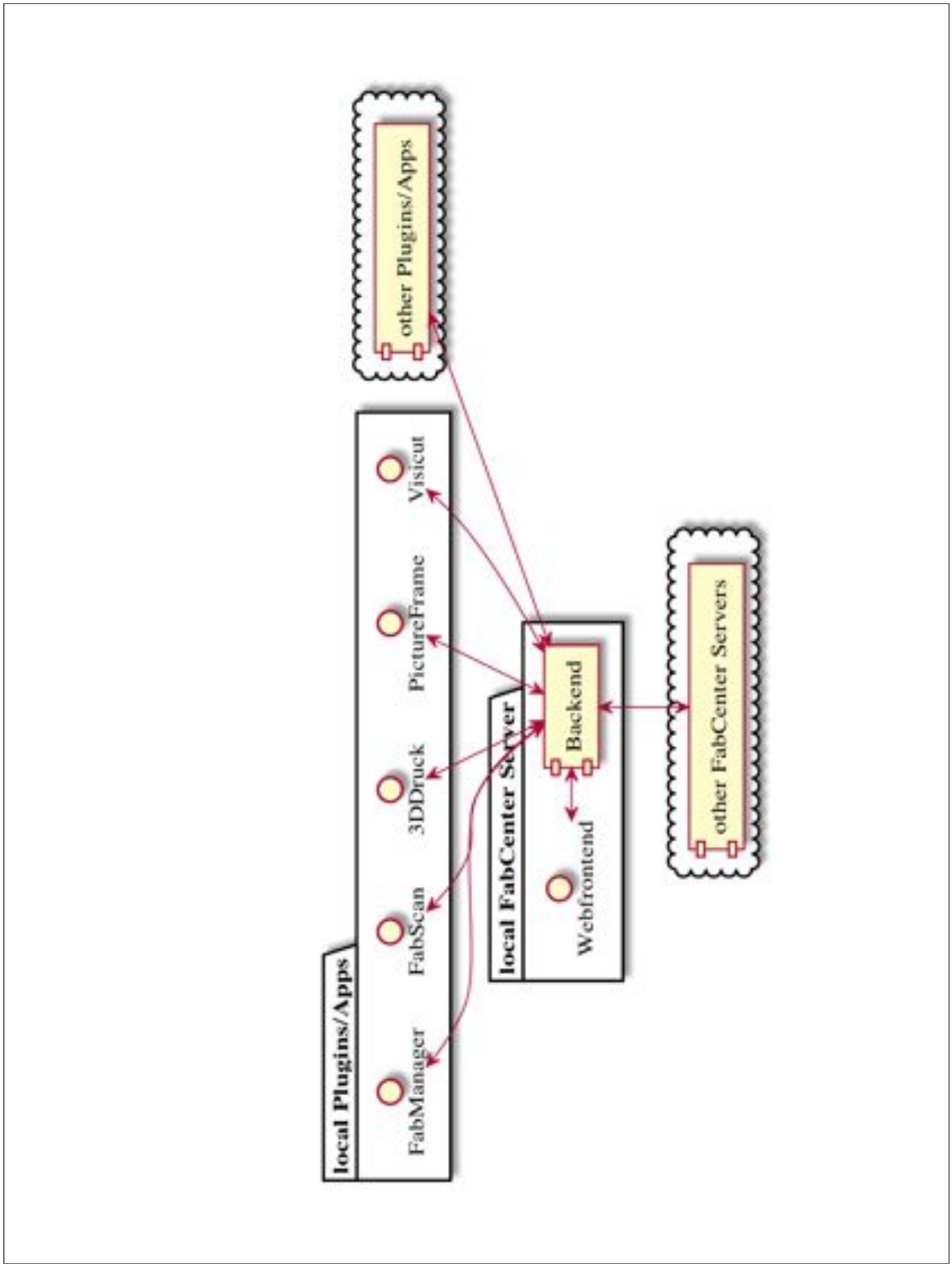


Figure D.1: Brainstorming session - slide one

- Island Solution
  - User may not create public accounts (children) and would be singled out from big and public centralized platforms.
  - Designs may be private, cannot be out of hand.
    Licensing issues prevent users from using platforms like thingiverse (if uploaded there, it is public right from the start to everybody)
- Granular Sharing
  - Design my be shared on a defined level
- Modular/Extensible
  - production-devices or creative apps can be integrated

Figure D.2: Brainstorming session - slide two

Survey

- Most people do not use a fab lab yet. If they do, the are most likely involved with it too.
- Only a few people use thingiverse.com or similar, but most would use a local platform, if provided.
- Scheduling visits seems to be an Aachen-centric issue, but most people would like to schedule online, and upload designs before the visit.
- Wanted Features: sharing across platforms, free (libre), open source, editing online, searching by keyword & properties, documentation with media files & text, given structure for documentation, cost/time calculator, scheduling system.
- Most fab labs have open lab days
- people who want to share and gain feedback via internet, often are fab lab involved

Figure D.3: Brainstorming session - slide three

User story I

View of a new fab lab user

Steve is an engineer and wants to design a gearbox for his remote controlled model car.

He registered with his local fab lab and loaded the first sketches up into his account. The project is private for now, since he wants to work out some thing before showing everything to others.

He decides to share his project with two of his friends, he also gives them the ability to write on his project content.

After some design iterations the gearbox is ready and Steve worked out an instruction list to recreate their success. The project is not shared on a local base, after some feedback from users he decides to make it global available since he is aiming for more feedback and users of his design.

Figure D.4: Brainstorming session - slide four

User story II

View of a longterm fab lab user

Gerald is a hacker and open source hardware enthusiast.
All his projects are public, and he already crated many little things, that he and other users use in their everyday life.

Since he has no fresh ideas for something to create, for now, he likes to look at other‘s designs and get inspired by them.

One project had some unanswered questions/problems in one of the log entries - He has got some ideas to solve this.

He communicated with the owner of the design and now contributes his knowledge, after the owner also gave him access to the content.

Figure D.5: Brainstorming session - slide five

User story III

View of a longterm fab lab user

Max is a lab master of the local fab lab.
To prepare for the open LabDay in two days he looks into his FabCenter account and gets an overview of the visiting users an their designs to fabricate.

He notices that one user will not bring his own material, and knows that this specific material is out of stock in the lab.
After planing a tour to the DIY market, he accepts some schedule-requests for the next weeks.
Also he sends some mails to the visitors in order to fill out some open questions about their visit.

At the open LabDay the local FabCenter Terminal shows all schedules projects and is ready for the visitors to log in and use the machine park.

Figure D.6: Brainstorming session - slide six

User story IV

group of children/teacher

Peter is a teacher in the USA. He and his class of 10-11 aged children take part in a program, where they get to visit the local fab lab to learn about DIY fabrication.

To be able to use the infrastructure they need to have an account in the system, to work on their things and finally print or cut them out.

Because of the Children's Online Privacy Protection Act of 1998, anyone of the children is allowed to have an account in the internet, but luckily the fab lab uses a local solution, such that the children‘s accounts only get permissions to share within the group of children in the local intranet. No data reaches the internet, and everybody may use the system.

Figure D.7: Brainstorming session - slide seven

Appendix E

Early Paper Prototype

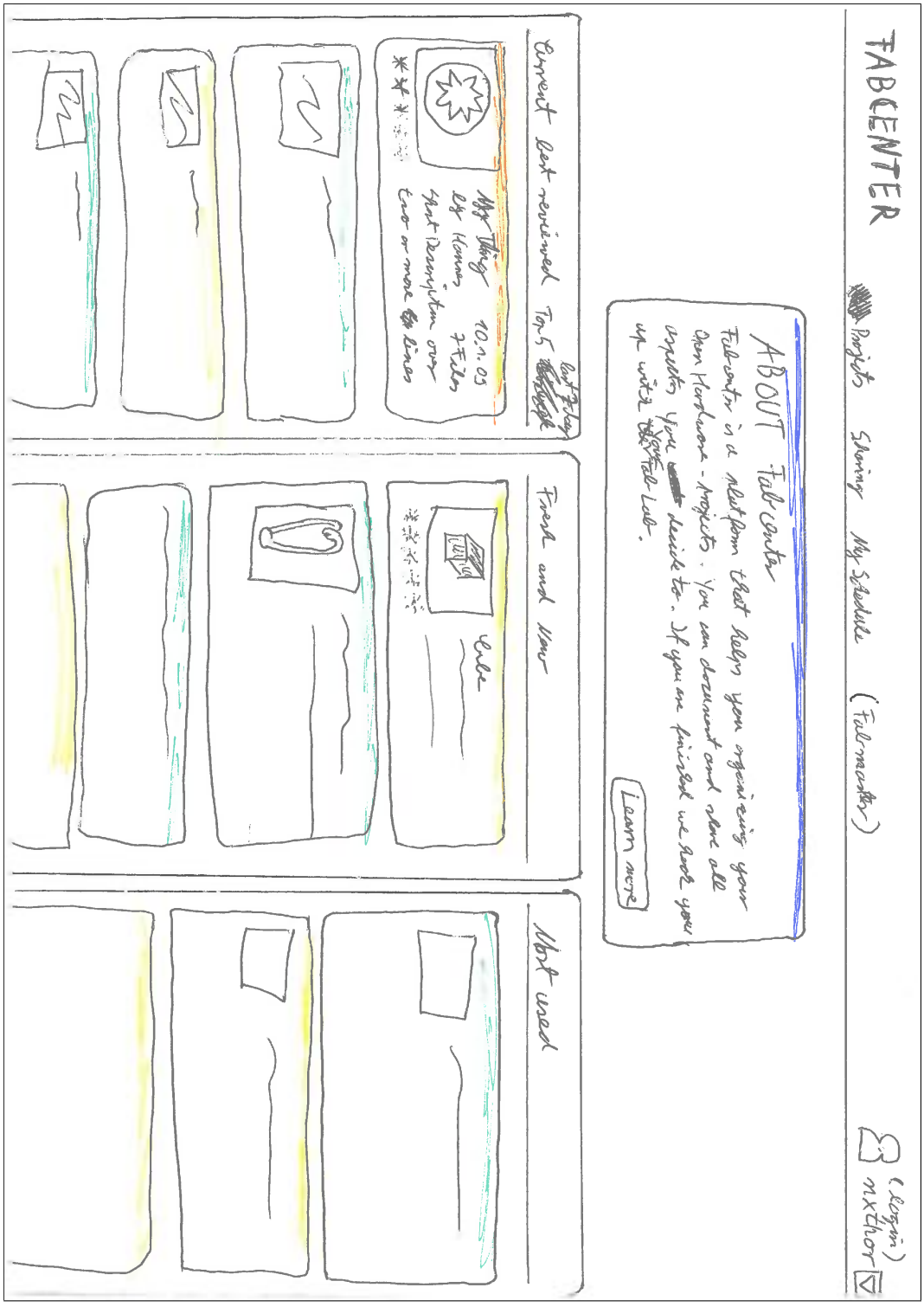


Figure E.1: Welcomescreen from the early paper prototype

Figure E.2: Project overview from the early paper prototype

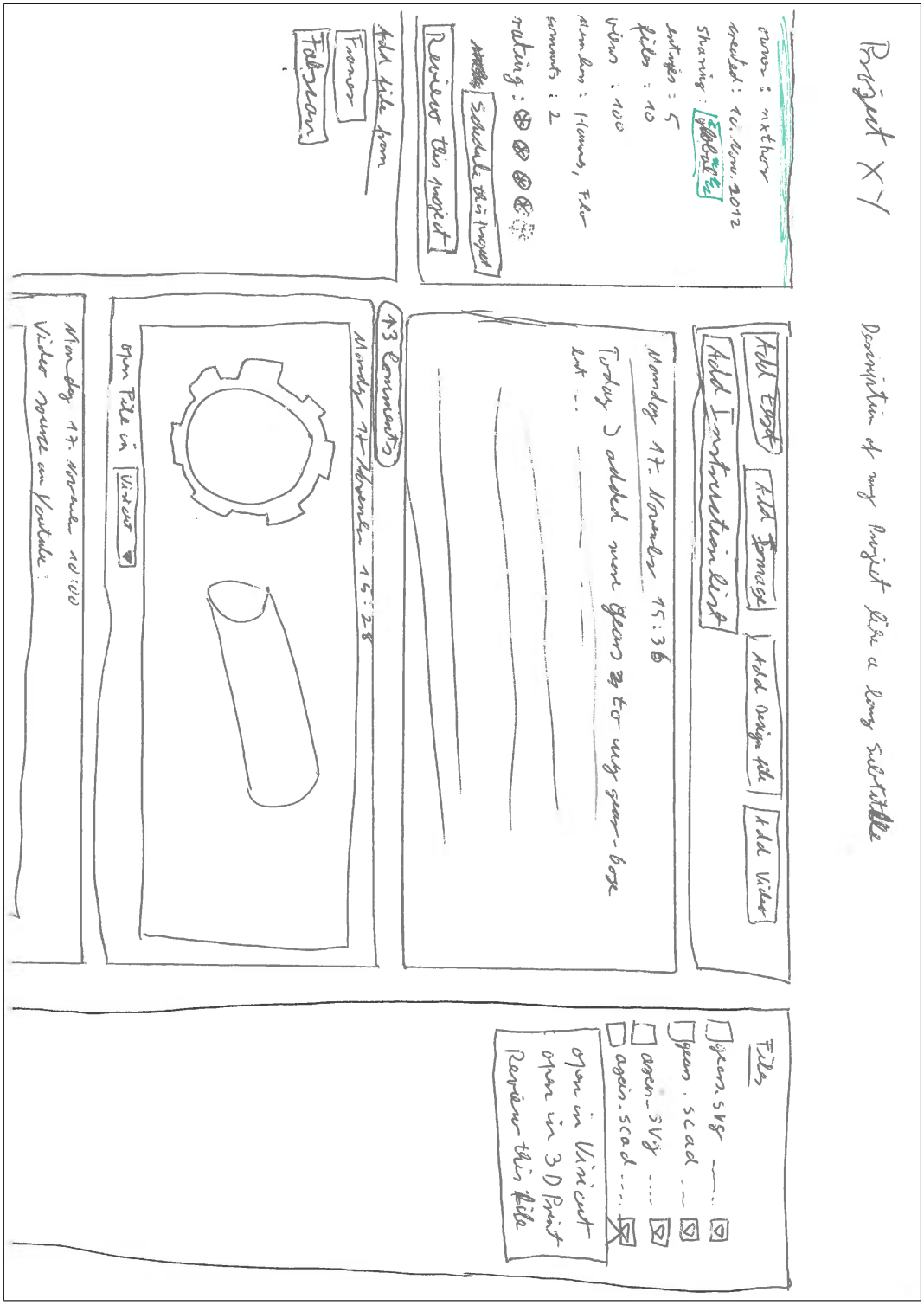


Figure E.3: Project details from the early paper prototype

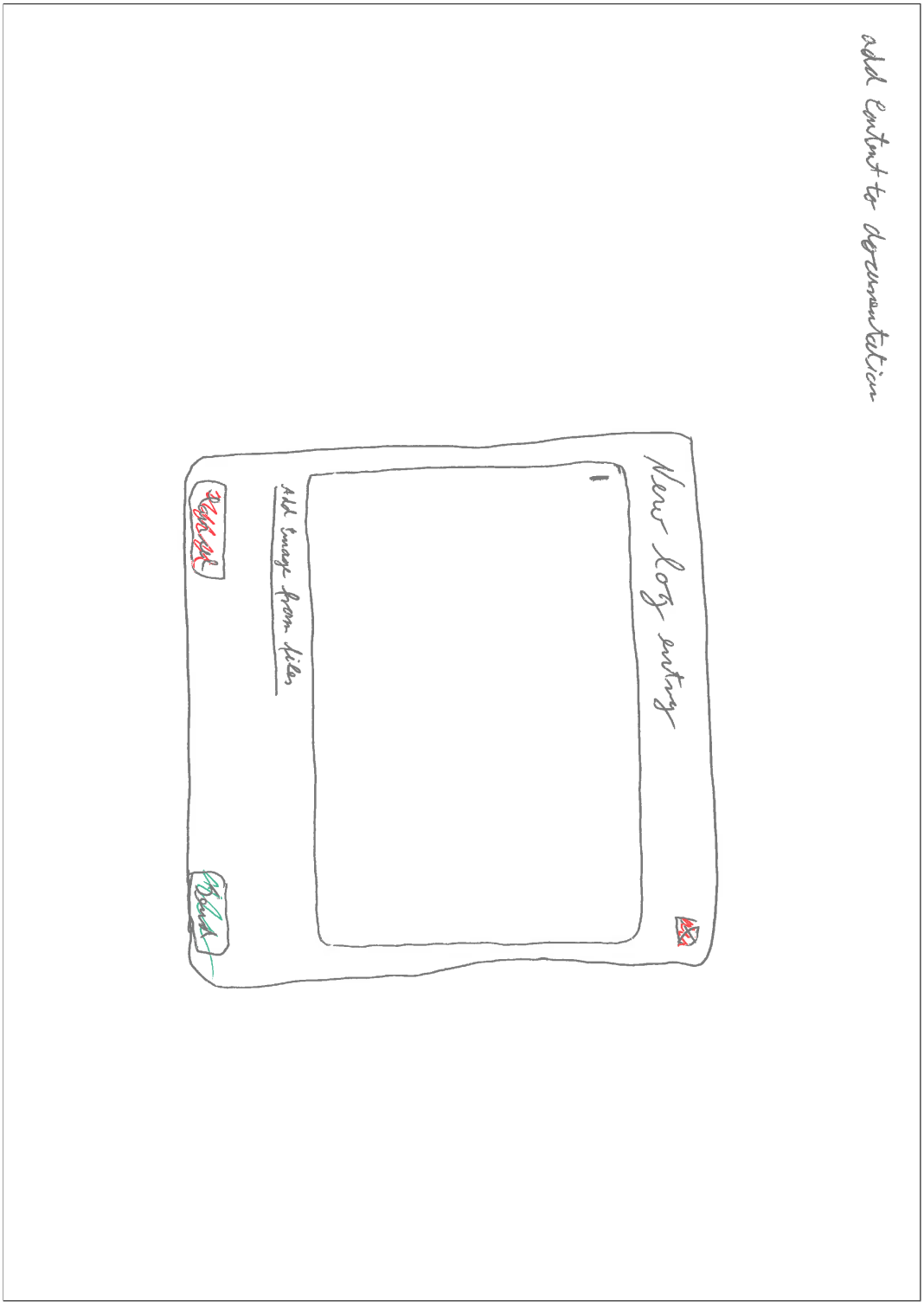


Figure E.4: New text-entry dialog from the early paper prototype

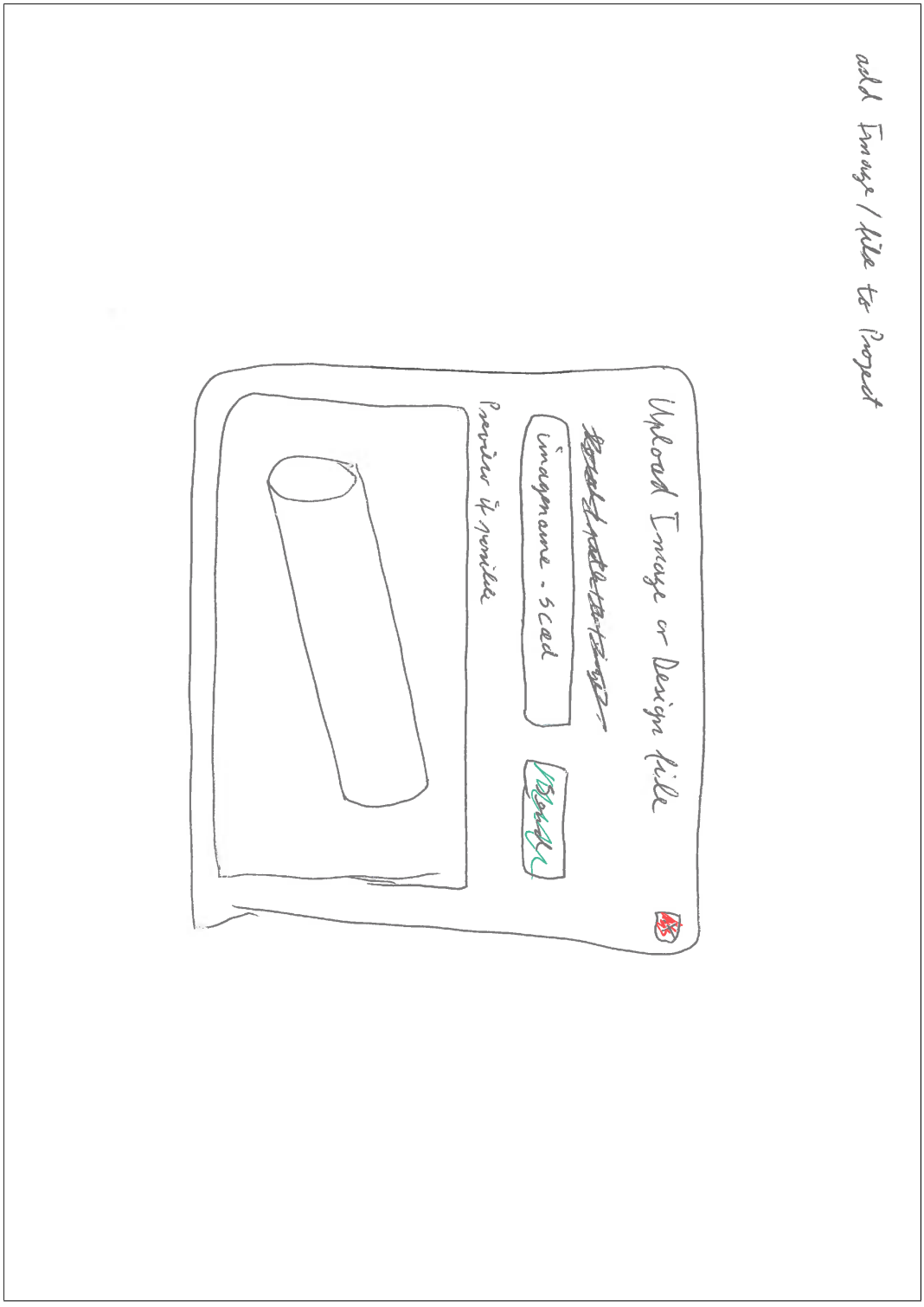


Figure E.5: File upload dialog from the early paper prototype

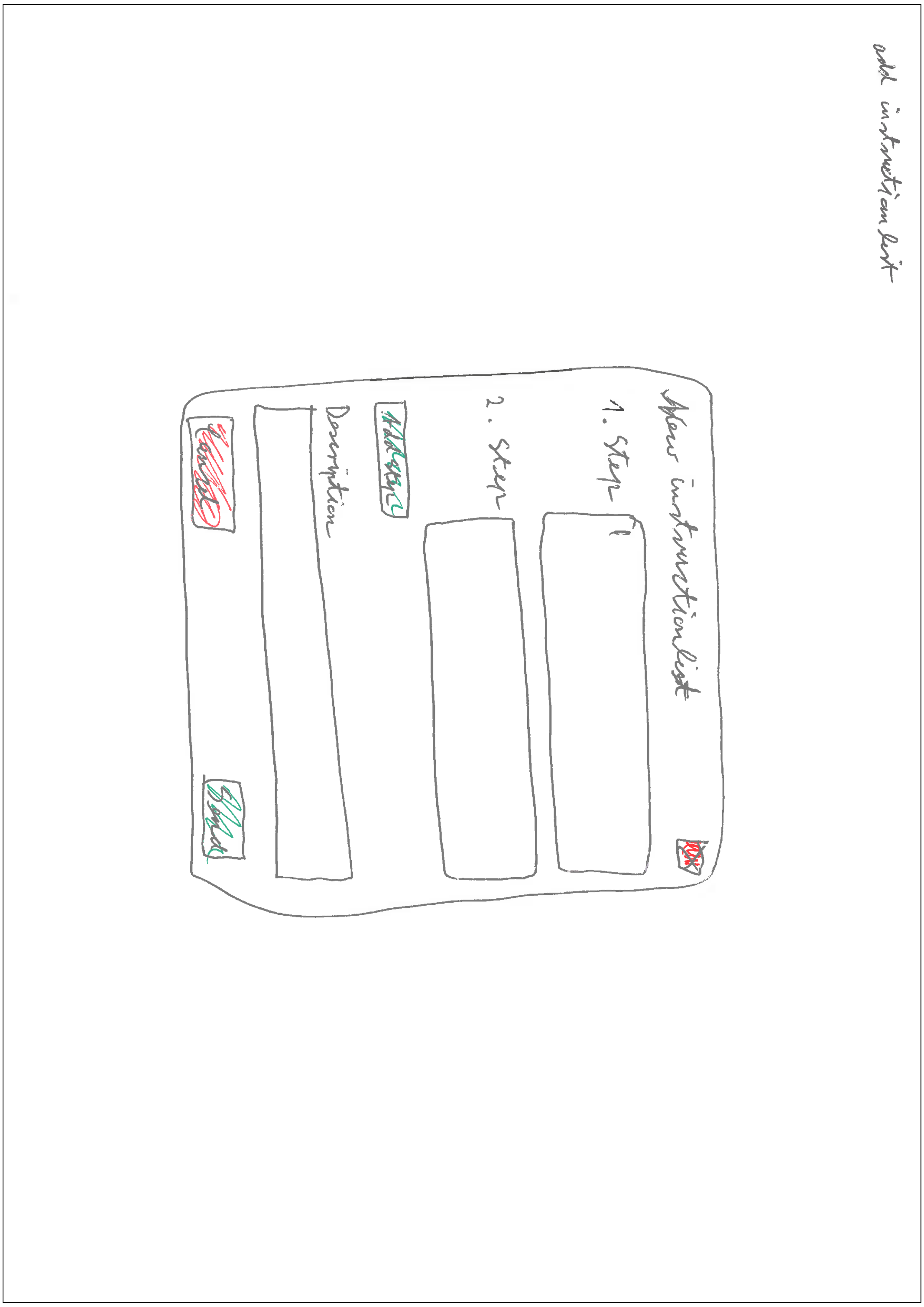


Figure E.6: Add instruction set dialog from the early paper prototype

Figure E.7: Sharing view of own projects from the early paper prototype

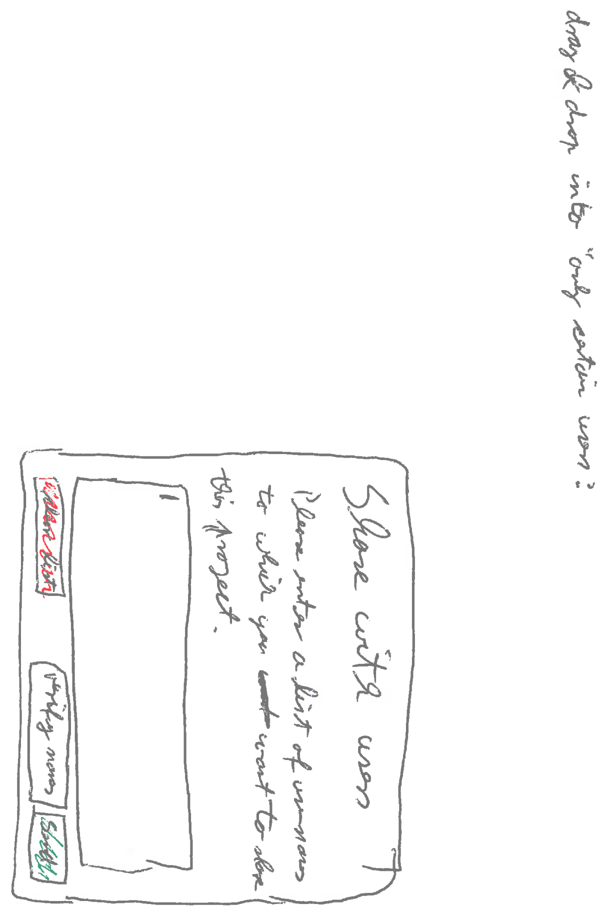


Figure E.8: Sharing dialog from the early paper prototype

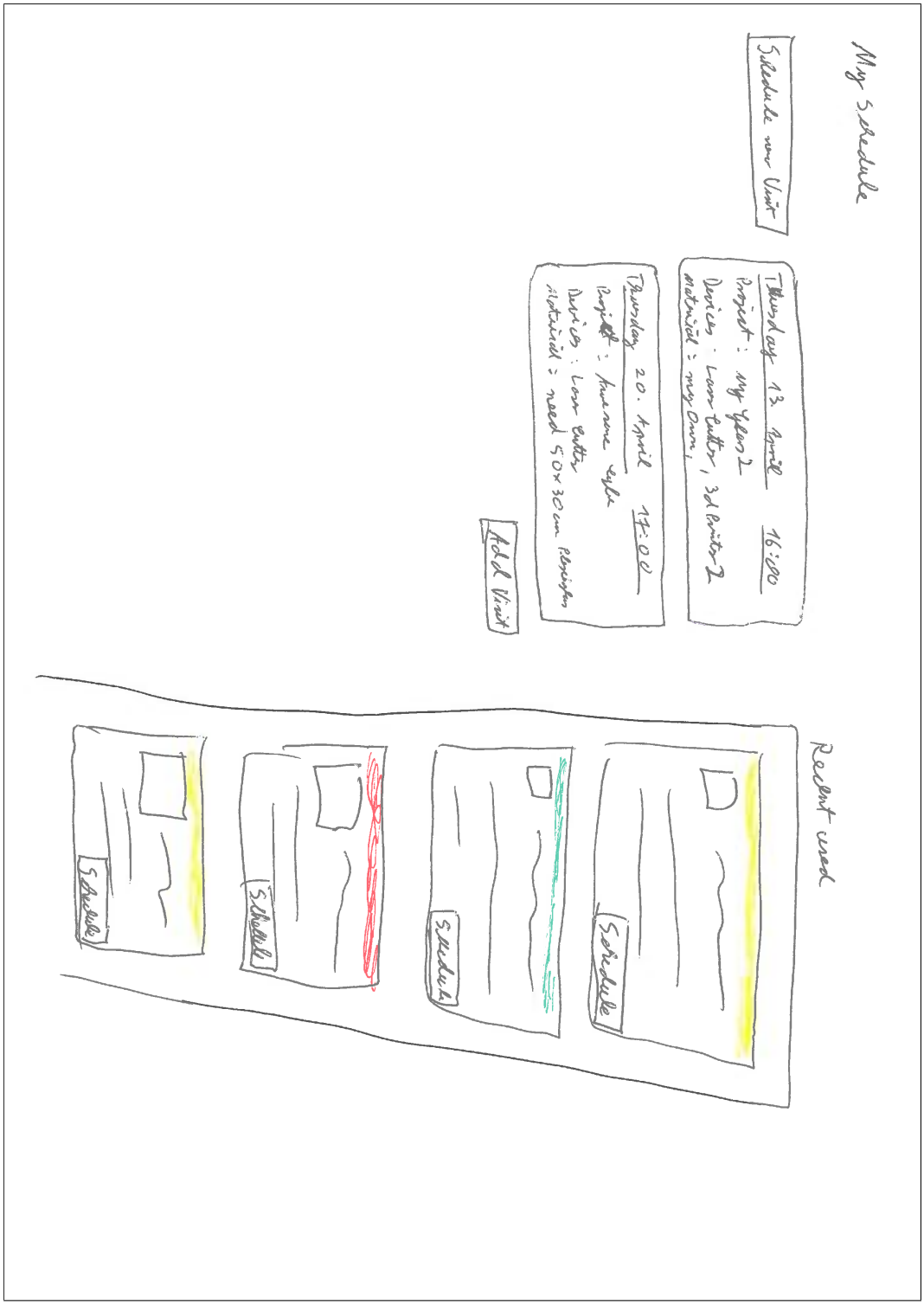


Figure E.9: User's schedule from the early paper prototype

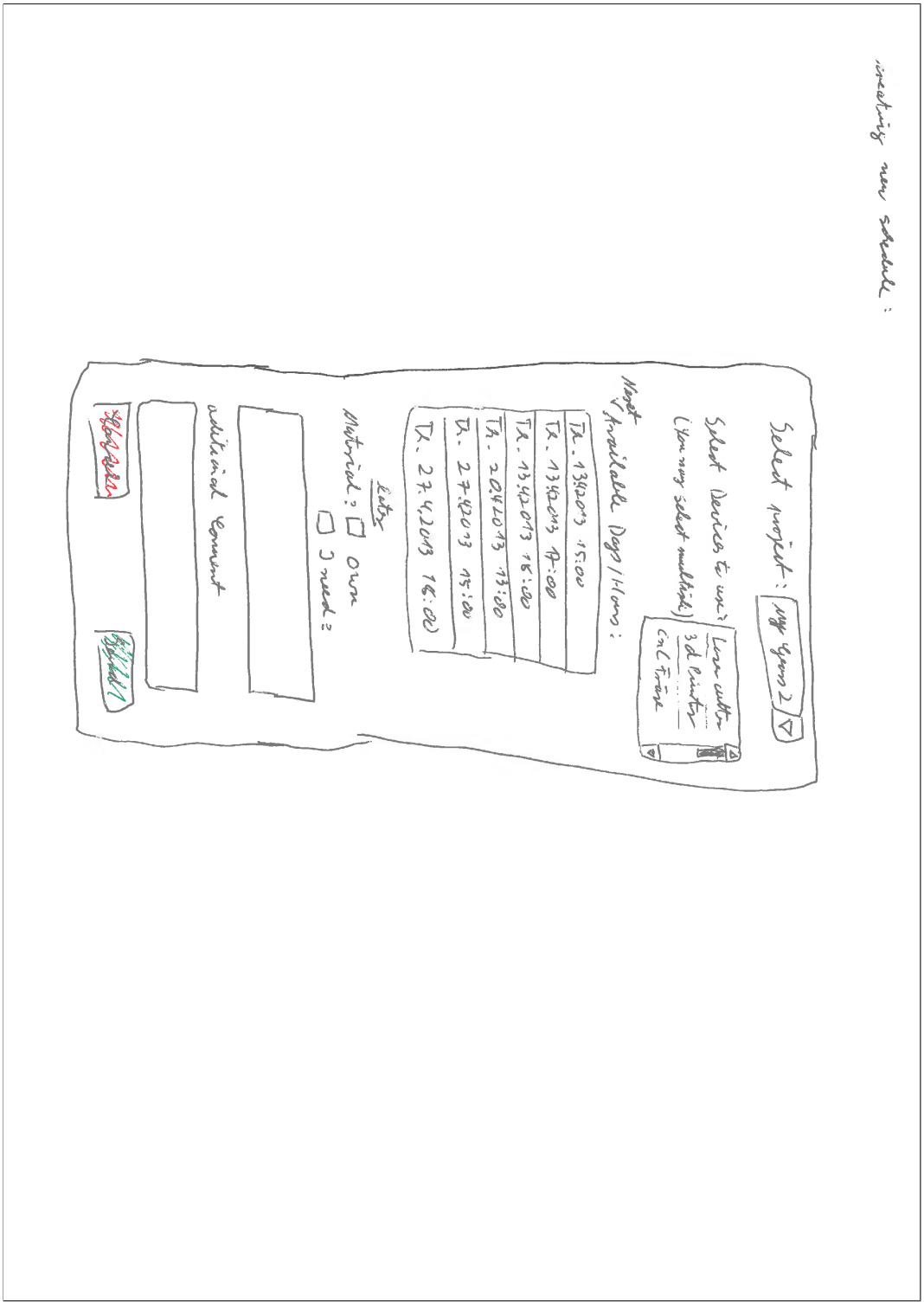


Figure E.10: Creating new schedule dialog from the early paper prototype

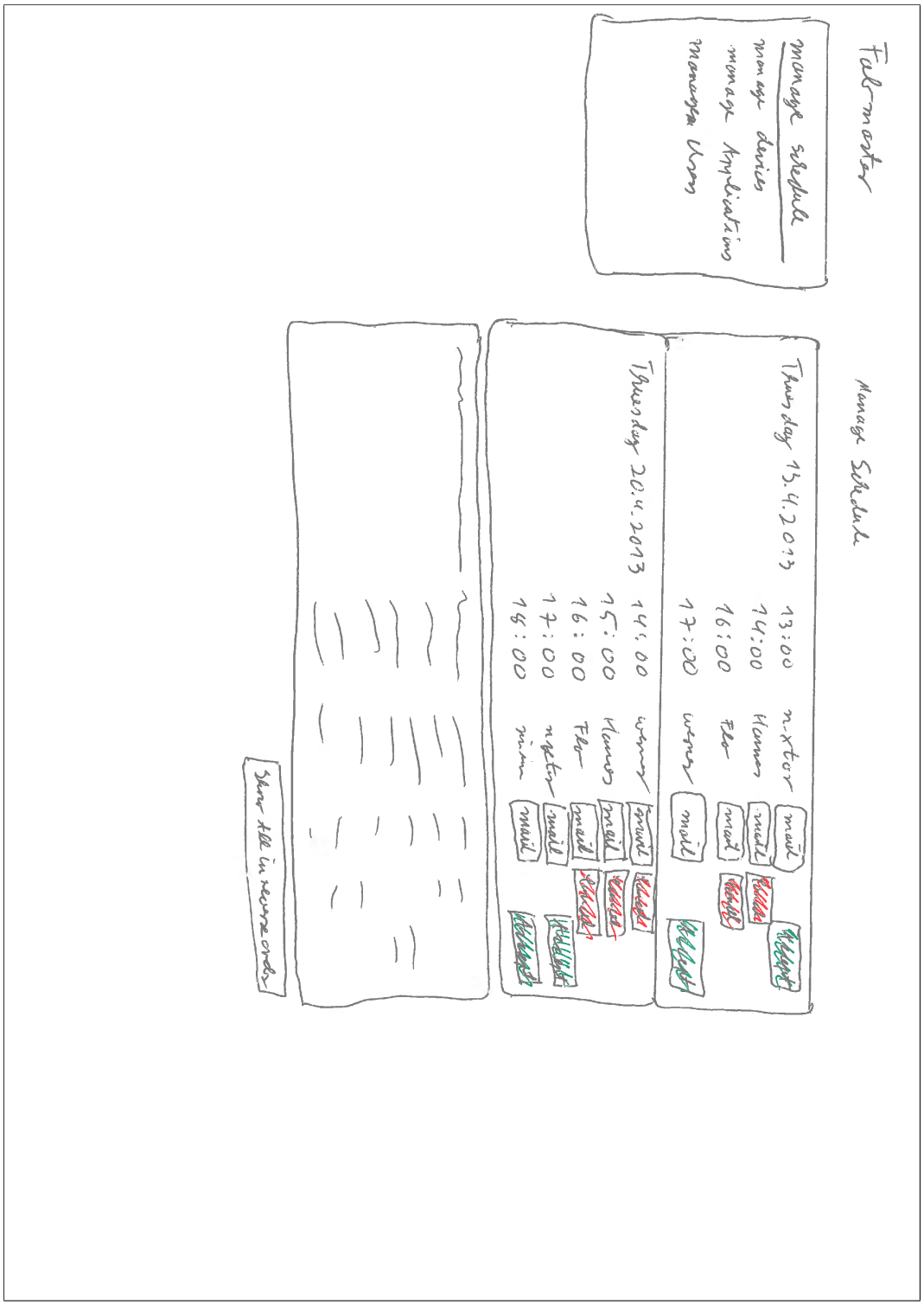


Figure E.11: Labadministrator's schedule management from the early paper prototype

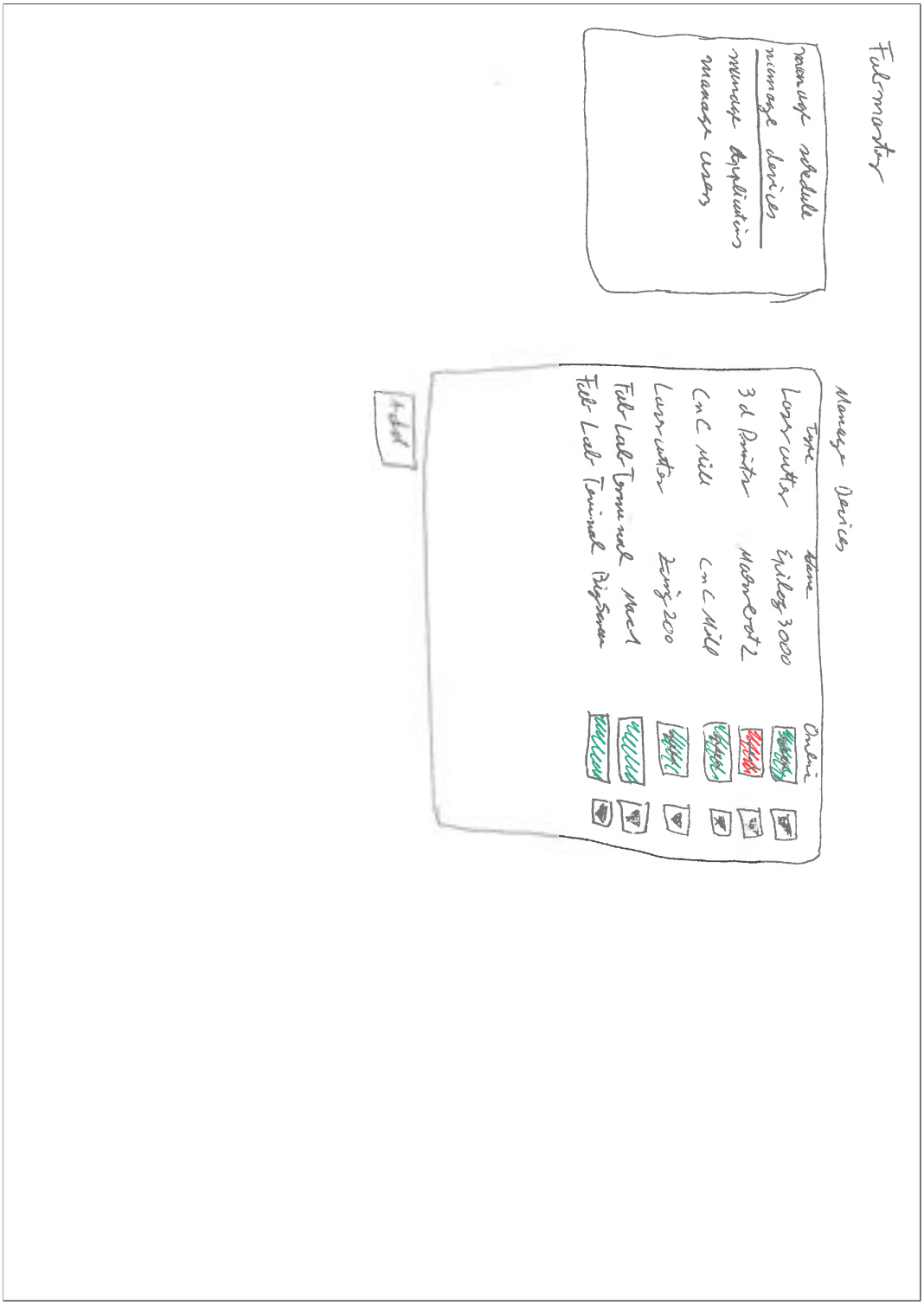


Figure E.12: Labadministrator's device management from the early paper prototype

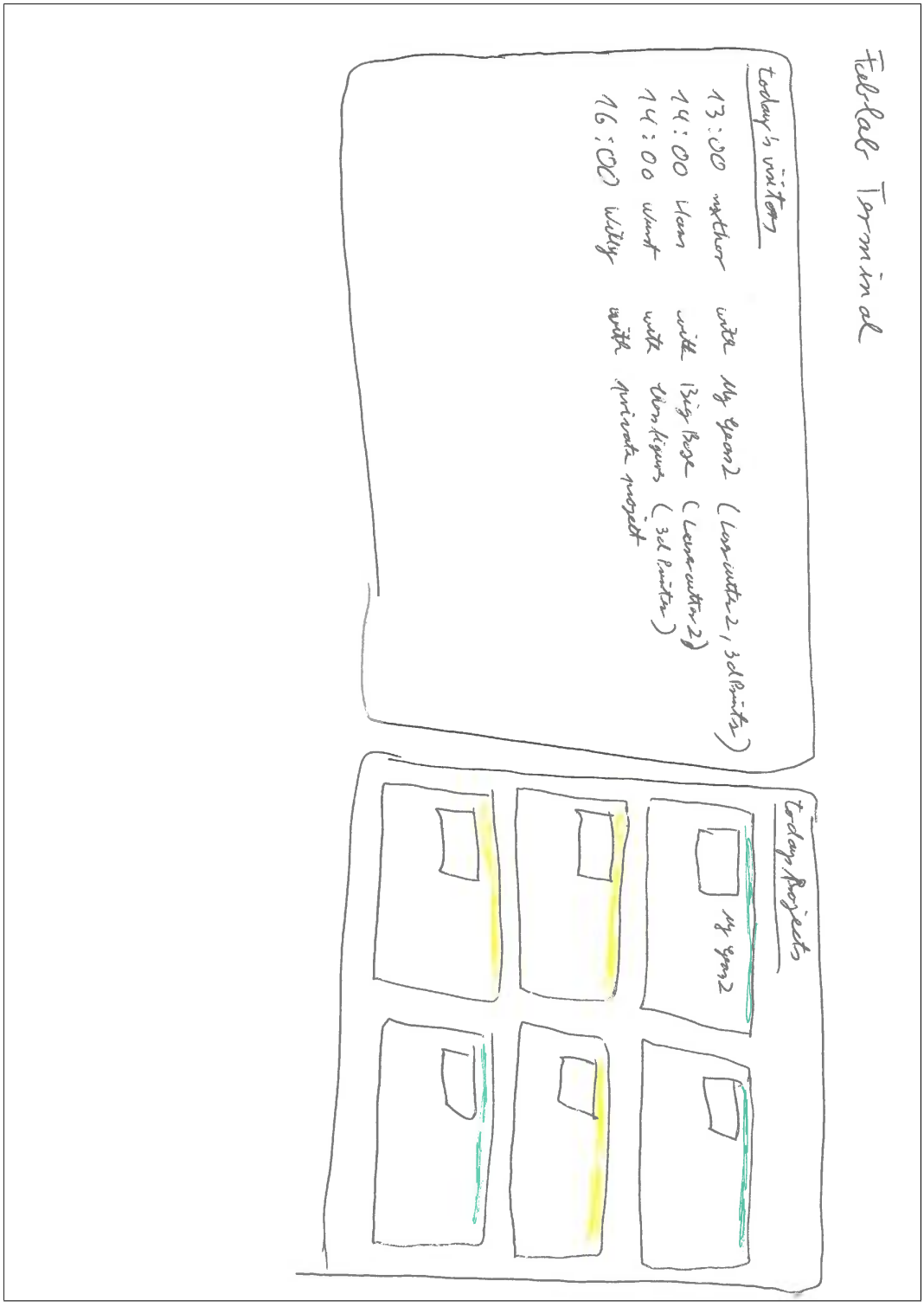


Figure E.13: Fablab's Terminal view from the early paper prototype

Appendix F

Final User Study

Fabcenter - Qualitative Evaluation - Users Participant ID: _____

Fabcenter - Qualitative Evaluation - Users

Form of Consent

I agree, that the data generated in this user study will be stored and analyzed for research within the Fabcenter project. I agree, that this data may be published in anonymized form. Under no circumstances shall this data be published or be otherwise made available to a third party without prior removal of all personally identifiable data.

Aachen, _____ _____
(Date) (Signature)

Initial Questionaire

Gender: [ ] male [ ] female

Age: _____

Profession/field of study: _____

Please rate your familiarity with...

(Q a) Personal Fabrication	unknown	[ ]	[ ]	[ ]	[ ]	[ ]	very familiar
(Q b) Fablabs	unknown	[ ]	[ ]	[ ]	[ ]	[ ]	very familiar
(Q c) Thingiverse.com	unknown	[ ]	[ ]	[ ]	[ ]	[ ]	very familiar

Figure F.1: Page one of the qualitative user study including a section for usablity scale targeting users

Fabcenter - Qualitative Evaluation - Users Participant ID: _____

Task 1:	Login into Fabcenter	Notes:
Task 2:	Get a list of your projects	Notes:
Task 3:	Schedule a visit with your project	Notes:
Task 4:	Checkin to Fabcenter‘s Terminal	Notes:
Task 5:	Download the svg-file of your project	Notes:
Task 6:	Open the svg-file with Visicut	Notes:
Task 7:	Lasercut the file	Notes:
Task 8:	Logout of the Terminal and follow the instructions	Notes:

Figure F.2: Page two of the qualitative user study including a section for usablity scale targeting users

Fabcenter - Qualitative Evaluation - Users Participant ID: _____

Fabcenter - Qualitative Evaluation - Users

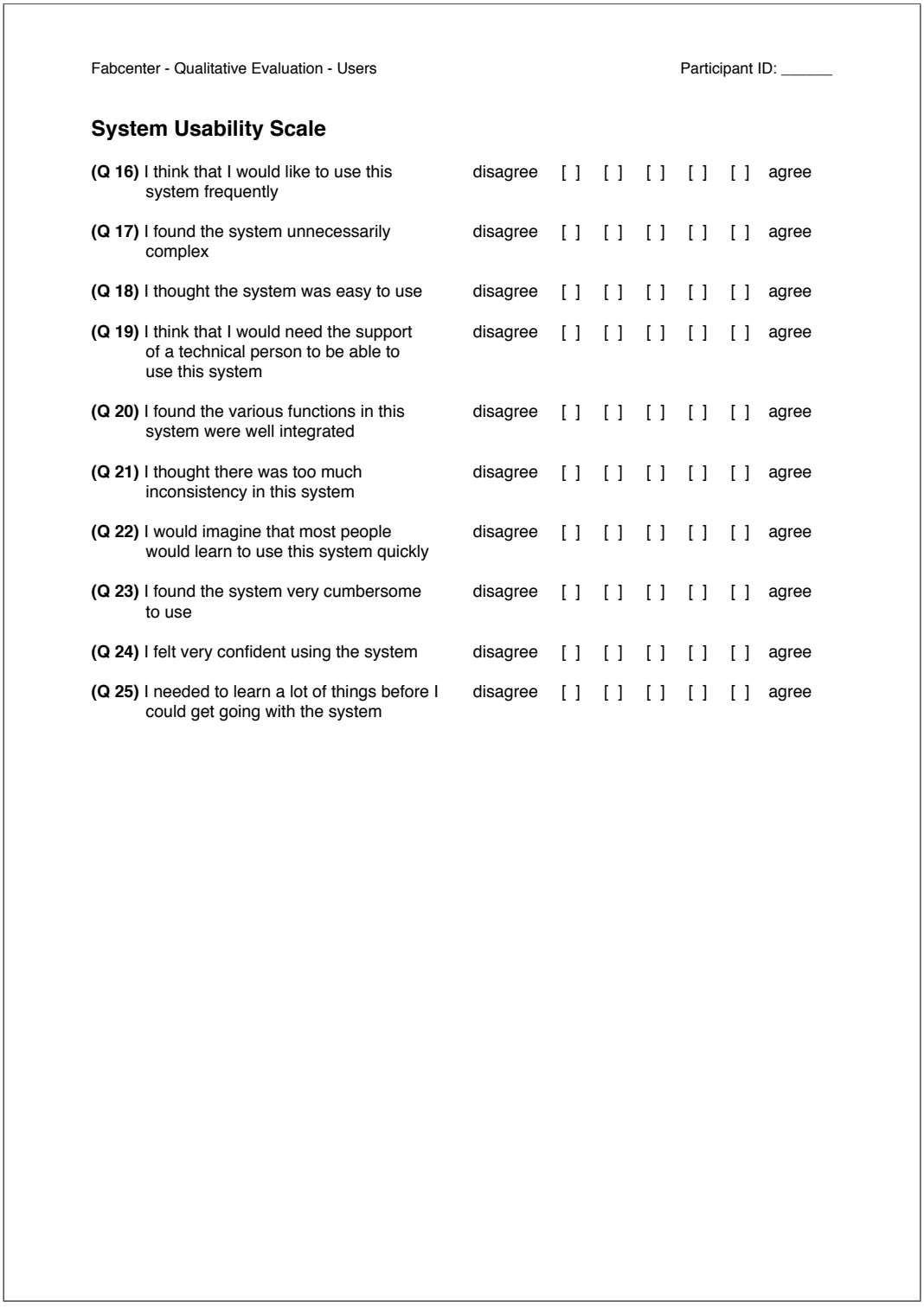
Task related

(Q 01) It was easy to login into the system	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 02) It was clear where to find my projects	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 03) I was able to schedule a visit in a uncomplicated way	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 04) It was easy to check in with the terminal	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 05) I found the way to download the needed file to be convenient	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 06) Opening the file in Viscut was simple	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 07) I had no problems lasercutting the file	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 08) I think that the documentation step is reasonable	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 09) I would like to use the system again for this tasks	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree

System related

(Q 10) The system gave feedback to my interactions	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 11) The system did show me how to reach my goals	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 12) I found it was worthwhile to document my new experiences	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 13) It was clear how to use the forms and what data to enter/select	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 14) I can see the documenting process adding value to the public	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 15) Overall, I am satisfied with the easy usage of the system in this condition	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree

Figure F.3: Page three of the qualitative user study including a section for usablity scale targeting users



Fabcenter - Qualitative Evaluation - Users Participant ID: _____

System Usability Scale

(Q 16) I think that I would like to use this system frequently	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 17) I found the system unnecessarily complex	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 18) I thought the system was easy to use	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 19) I think that I would need the support of a technical person to be able to use this system	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 20) I found the various functions in this system were well integrated	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 21) I thought there was too much inconsistency in this system	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 22) I would imagine that most people would learn to use this system quickly	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 23) I found the system very cumbersome to use	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 24) I felt very confident using the system	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 25) I needed to learn a lot of things before I could get going with the system	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree

Figure F.4: Page four of the qualitative user study including a section for usablity scale targeting users

Fabcenter - Qualitative Evaluation - Users Participant ID: _____

Fabcenter - Qualitative Evaluation - Administrators

Form of Consent

I agree, that the data generated in this user study will be stored and analyzed for research within the Fabcenter project. I agree, that this data may be published in anonymized form. Under no circumstances shall this data be published or be otherwise made available to a third party without prior removal of all personally identifiable data.

Aachen, _____ _____
(Date) (Signature)

Initial Questionaire

Gender: [ ] male [ ] female

Age: _____

Profession/field of study: _____

Please rate you familiarity with...

(Q a) Personal Fabrictation	unknown	[ ]	[ ]	[ ]	[ ]	[ ]	very familiar
(Q b) Fablabs	unknown	[ ]	[ ]	[ ]	[ ]	[ ]	very familiar
(Q c) Thingiverse.com	unknown	[ ]	[ ]	[ ]	[ ]	[ ]	very familiar

Figure F.5: Page one of the qualitative user study including a section for usablity scale targeting administrators

Fabcenter - Qualitative Evaluation - Users Participant ID: _____

Task 1

You are running a Fablab, and want to use the Fabcenter-platform to organize your staff‘s and visitor‘s schedule within the lab.

Please login into Fabcenter, and go to your lab‘s settings

Task 2

Add a staff-member who is taking over a shift every week on fridays from 9am to 3pm in future. the Thingiverse-account is fablabaachen the clear name is „FabLab Aachen“ and the tag is „LabAc“

Task 3

A new Lasercutter arrived. After setup you want your visitors to be able to book it. Please add it to the system. It is a Zing300 Lasercutter

Task 4

A visitor entered a request. Please confirm that request.

Task 5

One already scheduled visitor send an e-mail to cancel his/her visit, but did not cancel in the system on his/her own. Please cancel that visit.

Figure F.6: Page two of the qualitative user study including a section for usablity scale targeting administrators

Fabcenter - Qualitative Evaluation - Users Participant ID: _____

Fabcenter - Qualitative Evaluation - Administrators

Task related

(Q 01) It was easy to login into the system.	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 02) I found quickly where to setup my fablab	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 03) I had no problems to add a staff member	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 04) I could easily add th shift of the staff member	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 05) Accepting the visitor‘s request was simple	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 06) To cancel the scheduled visit was a short and easy task	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree

System related

(Q 07) The system gave feedback to my interactions	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 08) The system did show me how to reach my goals	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 09) It was clear how to use the forms and what data to enter/select	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 10) Overall, I am satisfied with the easy usage of the system in this condition	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree

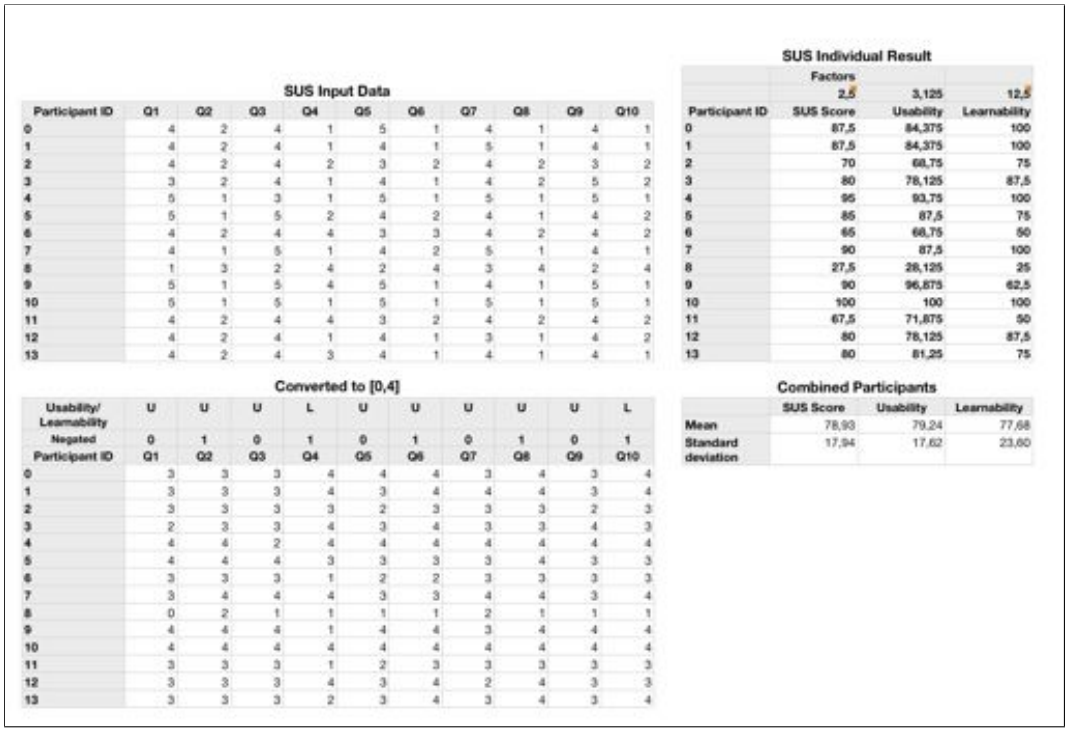
Figure F.7: Page three of the qualitative user study including a section for usablity scale targeting administrators

Fabcenter - Qualitative Evaluation - Users Participant ID: _____

System Usability Scale

(Q 11) I think that I would like to use this system frequently	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 12) I found the system unnecessarily complex	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 13) I thought the system was easy to use	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 14) I think that I would need the support of a technical person to be able to use this system	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 15) I found the various functions in this system were well integrated	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 16) I thought there was too much inconsistency in this system	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 17) I would imagine that most people would learn to use this system quickly	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 18) I found the system very cumbersome to use	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 19) I felt very confident using the system	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree
(Q 20) I needed to learn a lot of things before I could get going with the system	disagree	[ ]	[ ]	[ ]	[ ]	[ ]	agree

Figure F.8: Page four of the qualitative user study including a section for usablity scale targeting administrators



SUS Input Data

Participant ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
0	4	2	4	1	5	1	4	1	4	1
1	4	2	4	1	4	1	5	1	4	1
2	4	2	4	2	3	2	4	2	3	2
3	3	2	4	1	4	1	4	2	5	2
4	5	1	3	1	5	1	5	1	5	1
5	5	1	5	2	4	2	4	1	4	2
6	4	2	4	4	3	3	4	2	4	2
7	4	1	5	1	4	2	5	1	4	1
8	1	3	2	4	2	4	3	4	2	4
9	5	1	5	4	5	1	4	1	5	1
10	5	1	5	1	5	1	5	1	5	1
11	4	2	4	4	3	2	4	2	4	2
12	4	2	4	1	4	1	3	1	4	2
13	4	2	4	3	4	1	4	1	4	1

Converted to [0,4]

Usability/ Learnability	U	U	U	L	U	U	U	U	U	L
Negated	0	1	0	1	0	1	0	1	0	1
Participant ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
0	3	3	3	4	4	4	3	4	3	4
1	3	3	3	4	3	4	4	4	3	4
2	3	3	3	3	2	3	3	3	2	3
3	2	3	3	4	3	4	3	3	4	3
4	4	4	2	4	4	4	4	4	4	4
5	4	4	4	3	3	3	3	4	3	3
6	3	3	3	1	2	2	3	3	3	3
7	3	4	4	4	3	3	4	4	3	4
8	0	2	1	1	1	1	2	1	1	1
9	4	4	4	1	4	4	3	4	4	4
10	4	4	4	4	4	4	4	4	4	4
11	3	3	3	1	2	3	3	3	3	3
12	3	3	3	4	3	4	2	4	3	3
13	3	3	3	2	3	4	3	4	3	4

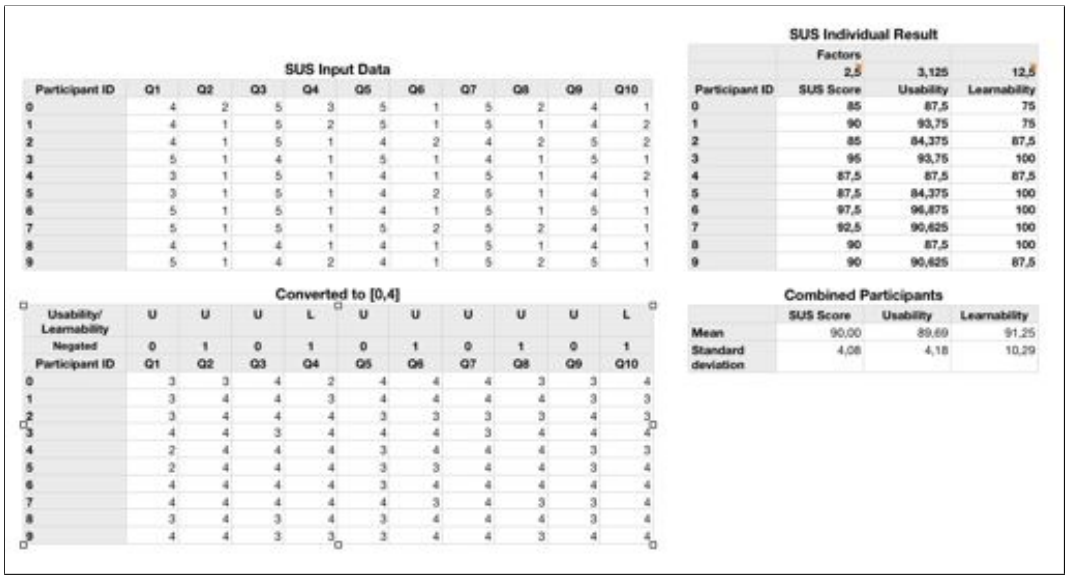
SUS Individual Result

	Factors		
	2,5	3,125	12,5
Participant ID	SUS Score	Usability	Learnability
0	87,5	84,375	100
1	87,5	84,375	100
2	70	68,75	75
3	80	78,125	87,5
4	95	93,75	100
5	85	87,5	75
6	65	68,75	50
7	90	87,5	100
8	27,5	28,125	25
9	90	96,875	62,5
10	100	100	100
11	67,5	71,875	50
12	80	78,125	87,5
13	80	81,25	75

Combined Participants

	SUS Score	Usability	Learnability
Mean	78,93	79,24	77,68
Standard deviation	17,94	17,62	23,60

Figure F.9: Calculation sheet of the SUS values for the SUS of the user-side



SUS Input Data

Participant ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
0	4	2	5	3	5	1	5	2	4	1
1	4	1	5	2	5	1	5	1	4	2
2	4	1	5	1	4	2	4	2	5	2
3	5	1	4	1	5	1	4	1	5	1
4	3	1	5	1	4	1	5	1	4	2
5	3	1	5	1	4	2	5	1	4	1
6	5	1	5	1	4	1	5	1	5	1
7	5	1	5	1	5	2	5	2	4	1
8	4	1	4	1	4	1	5	1	4	1
9	5	1	4	2	4	1	5	2	5	1

Converted to [0,4]

Usability/ Learnability	U	U	U	L	U	U	U	U	U	L
Negated	0	1	0	1	0	1	0	1	0	1
Participant ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
0	3	3	4	2	4	4	4	3	3	4
1	3	4	4	3	4	4	4	4	3	3
2	3	4	4	4	3	3	3	3	4	3
3	4	4	3	4	4	4	3	4	4	4
4	2	4	4	4	3	4	4	4	3	3
5	2	4	4	4	3	3	4	4	3	4
6	4	4	4	4	3	4	4	4	4	4
7	4	4	4	4	4	3	4	3	3	4
8	3	4	3	4	3	4	4	4	3	4
9	4	4	3	3	3	4	4	3	4	4

SUS Individual Result

	Factors		
	2,5	3,125	12,5
Participant ID	SUS Score	Usability	Learnability
0	85	87,5	75
1	90	93,75	75
2	85	84,375	87,5
3	95	93,75	100
4	87,5	87,5	87,5
5	87,5	84,375	100
6	97,5	96,875	100
7	92,5	90,625	100
8	90	87,5	100
9	90	90,625	87,5

Combined Participants

	SUS Score	Usability	Learnability
Mean	90,00	89,69	91,25
Standard deviation	4,08	4,18	10,29

Figure F.10: Calculation sheet of the SUS values for the SUS of the administrator-side

Bibliography

Aaron Bangor, Philip T. Kortum, and James T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24 (6):574–594, 2008. doi: 10.1080/10447310802205776. URL http://www.tandfonline.com/doi/abs/10.1080/10447310802205776.

Robert Battle and Edward Benson. Bridging the semantic web and web 2.0 with representational state transfer (rest). *J. Web Sem.*, 6(1):61–69, 2008.

Tim Berners-Lee. Information management: A proposal. Technical report, CERN, March 1989. URL http://www.w3.org/History/1989/proposal.html.

Owen Briggs, Steven Champeon, and Eric Costello. *Cascading Style Sheets: Separating Content from Presentation, Second Edition*. APress, 2004. ISBN 159059231X.

John Brooke. Sus: A quick and dirty usability scale, 1996.

V. Bush. As we may think. *The Atlantic Monthly*, July, 1945.

Donald D. Chamberlin and Raymond F. Boyce. Sequel: A structured english query language. In *Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control*, SIGFIDET '74, pages 249–264, New York, NY, USA, 1974. ACM. doi: 10.1145/800296.811515. URL http://doi.acm.org/10.1145/800296.811515.

Douglas Crockford. Json: The fat-free alternative to xml, December 2006. URL http://www.json.org/fatfree.html.

David Flanagan. *JavaScript: The Definitive Guide*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 3rd edition, 1998. ISBN 1565923928.

N.A. Gershenfeld. *Fab: The Coming Revolution on Your Desktop–from Personal Computers to Personal Fabrication*. Basic Books, 2005. ISBN 9780465027453. URL `http://books.google.com.ph/books?id=Oil3bH6fKBkC`.

Zhao He. Fabiji - a tablet kiosk to facilitate creating and sharing documentation at fab labs. Master's thesis, RWTH Aachen, March 2012.

G. E. Krasner and S. T. Pope. A cookbook for using the model-view controller user interface paradigm in smalltalk80. 1988.

Anu Määttä and Peter Troxler. Developing open & distributed tools for fablab project documentation. In Sebastian Hellmann, Philipp Frischmuth, Sören Auer, and Daniel Dietrich, editors, *OKCon*, volume 739 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011. URL `http://dblp.uni-trier.de/db/conf/okcon/okcon2011.html`.

Catarina Mota. The rise of personal fabrication. In *Proceedings of the 8th ACM conference on Creativity and cognition*, C&C '11, pages 279–288, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0820-5. doi: 10.1145/2069618.2069665. URL `http://doi.acm.org/10.1145/2069618.2069665`.

Donald A. Norman. *The Design of Everyday Things*. Basic Books, New York, reprint paperback edition, 2002. ISBN 0-465-06710-7.

George Toye, Mark R. Cutkosky, Larry J. Leifer, J. Marty Tenenbaum, and Jay Glicksman. Share: A methodology and environment for collaborative product development, 1993.

Mark Weiser. The Computer for the Twenty-First Century. *Scientific American*, 265(3):94–104, 1991.

Index

Typeset June 10, 2013