Fabiji

A Tablet Kiosk to Facilitate Creating and Sharing Documentation at Fab Labs

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

by
Zhao He

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof Jon Oberlander

Registration date: 6th February, 2012
Submission date: 30th March, 2012

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, March 2012
Zhao He

Contents

List of Figures

List of Tables

Abstract

Fab labs, as coined by Neil Gershenfeld, are global networks of local workshops equipped with a number of tools and machines offering digital fabrication for individuals. Since fab labs aim to provide the opportunity for people to create almost anything, they are popular among open source hardware enthusiasts. However, due to the lack of effective system for documenting open source hardware projects, many great projects created at fab labs either have no written documentation or their documentation has a limited range of audience.

The first part of this thesis describes a contextual inquiry study on fab lab users. The study reveals four types of typical user groups, and the analysis of the study explains the motivation for each user group to create and share project documentation. Requirements for Fabiji, a dedicated documentation system for fab lab users, are raised accordingly, and the Fabiji system is proposed to be the combination of an iPad application and a photo kiosk.

The second part of this thesis describes the design and implementation of Fabiji. Its iPad application focuses on helping and encouraging users to create basic documentation for physical objects in a short time, and it also aims to give users opportunities to meet in real life. The photo kiosk helps users to create quality photos with the iPad camera. Both the iPad application and the photo kiosk can be easily deployed and configured at other fab labs.

Results from the user study shows that Fabiji fulfils all the requirements, and participants think that it non-intrusively helps them to document physical projects at the fab lab.

Überblick

Fab Labs, wie Neil Gershenfeld sie prägte, sind ein globales Netzwerk von lokalen Werkstätten mit einer Reihe von Werkzeugen und Maschinen, mit denen digitale Fertigung für Privatpersonen möglich ist. Da Fab Labs den Menschen die Gelegenheit geben, fast alles zu fertigen, sind sie unter den Open-Source-Hardware-Enthusiasten sehr beliebt. Da ein wirksames System für die Dokumentation von Open-Source-Hardware-Projekten fehlt, werden viele großartige Projekte, die in Fab Labs erstellt werden, nicht schriftlich dokumentiert oder ihre Dokumentation ist nur einem begrenzten Publikum zugänglich.

Der erste Teil dieser Arbeit beschreibt eine kontextuelle Umfrage-Studie von Fab Lab Benutzern. Die Studie lässt vier Typen von typischen Benutzergruppen erkennen, und die Analyse der Studie erklärt die Beweggründe für jede Benutzergruppe, die Projektdokumentation zu erstellen und zu teilen. Anforderungen an Fabiji, einem Dokumentationssystem für Fab Lab Benutzer, werden heraus gearbeitet und es wird vorgeschlagen, dass das Fabiji System eine Kombination aus einer iPad-Anwendung und einem Foto-Kiosk sein soll.

Der zweite Teil dieser Arbeit beschreibt die Konzeption und Umsetzung von Fabiji. Die iPad-Applikation konzentriert sich auf die Unterstützung und Ermutigung von Nutzern, grundlegende Dokumentation für physikalische Objekte in kurzer Zeit zu erstellen, und es soll den Anwendern die Möglichkeit geboten werden, sich im realen Leben zu treffen. Das Foto-Kiosk unterstützt Anwender dabei, hochwertige Fotos mit der iPad-Kamera zu machen. Die iPad-Anwendung und das Foto-Kiosk können einfach in Fab Labs eingerichtet und konfiguriert werden.

Die Ergebnisse der Benutzer-Studie zeigt, dass Fabiji alle Anforderungen erfüllt, und die Teilnehmer denken, dass es ihnen nicht-intrusiv dabei hilft, physikalische Projekte in einem Fab Lab zu dokumentieren.

Acknowledgements

First of all, I would like to thank Prof Jan Borchers for being my advisor and René Bohne for being my thesis supervisor. They offered valuable suggestions throughout my thesis time, and helped me greatly with the organisation issues during my thesis writing.

I would also like to thank Prof Jon Oberlander for being my second examiner.

Special thanks go to Anna Witkopp, who helped design the photo kiosk during her internship at Fab Lab Aachen. I would never finish building such a beautiful kiosk without her.

Special thanks go to Chat Wacharamanotham for answering me numerous questions from the time before I entered EuMI program to the day of thesis submission. Special thanks also go to Yu Jing for offering me valuable advices and support in the last phase of my thesis.

I would like to thank Thomas Oster for building the web server and contributing to the API, and Zhang Ying for sketching the storyboard.

Many people from the Media Computing groups help me in various parts of my thesis, inlcluding Ceyda Avci (although her English is as poor as my German) and Cihan Aydinoglu for assisting me during the user study, Christian Corsten for advices in thesis writing, Simon Jakubowski for assisting video shooting.

Tang Siyu and Kittipat Virochsiri offered me useful tips in visa extension and semester extension, and it turns out that I owe them big thanks.

Last but not the least, I am deeply grateful to my beloved father and mother who always love and support me!

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in coloured boxes.

FAB LAB:
A fab lab is a small workshop with machines and tools to support digital fabrication.

Definition:
Fab Lab

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

`RootViewController`

The whole thesis is written in British English. Unidentified third persons are described in male form. The first person is written in plural form.

Download links are set off in coloured boxes.

File: Filename[a]

[a] http://hci.rwth-aachen.de/~he/thesis/Filename.zip

Chapter 1

Introduction

1.1 Background

Fab labs (fabrication laboratory) are global networks of local labs, which are small workshops equipped with a number of tools and machines offering digital fabrication for individuals [Gershenfeld, 2005]. A fab lab provides the opportunity for people to create almost anything. In order to share knowledge and encourage peer learning, a fab lab expects users to contribute to documentation and instruction. At present, there are nearly 70 fab labs running around the world, with another dozen under construction.

Fab labs are small workshops with machines and tools to support digital fabrication.

There are three fab labs established in Germany, namely in the city of Aachen, Munich, and Nuremberg, and another handful fab labs are under planning. Among them, Fab Lab Aachen[1] was the first fab lab in Germany, established in December 2009 at the RWTH Aachen University.

Fab Lab Aachen was established as the first fab lab in Germany in 2009.

[1]http://fablab.rwth-aachen.de

Definition:
Fab Lab Open Day

FAB LAB OPEN DAY:
Fab Lab Open Day is a once-per-week event. On this day, everyone can work at the fab lab on his personal or research projects with access to all machines and tools available at the lab. For visitors who have no previous experience in fab labs or digital fabrication, the technical staff will give them a tour of the lab.

On Fab Lab Open Day, everyone can have access to the machines and tools at the fab lab.

Fab Lab Aachen opens its door to everyone. Every Tuesday is its open day, depending on the project needs and lab supplies, one can come to the fab lab with or without prepared material and have access to all the machines and tools provided, such as 3D printers, laser cutters, and PCB milling machines. Currently, each of the machines mentioned above has only one or two of its kind. To ensure machine availability, one has to make an appointment online with the fab lab master in advance, and specify the machine(s) to be used. After receiving the appointment request, the fab lab master will arrange time slots and then gives the user a one hour session for that machine. If more time is required, the user is asked to separate the project into several one-hour session tasks. At the fab lab, a technical staff will first familiarise the user with the operation instruction of the machines and use of the tools, and then records project information while the user is busy with the machine. Depending on which machine is used, recorded information includes size, materials, and machine parameters. After the user finished his or her work, the technical staff takes a picture of the physical object that the user has just created.

Fab Lab promotes personal fabrication.

Unlike mass manufacturing where a series of identical objects are created at a time, fab labs make possible of making one-of-a-kind products or parts at low cost [Mota, 2011]. With various open source software supporting design physical objects, many people have created great open-source hardware projects and publish them online DIY communities [Mellis et al., 2011]. Since these projects are open-source, other people can make a same copy or add modification to the original design. With the ability and freedom to share, remake, and modify hardware design, it is possible for everyone to digitally fabricate everything, hence personal fabrication can be achieved.

Apart from the accessibility to machines and software that support digital fabrication, another key element to the development of personal fabrication is open-source hardware documentation. Open-source hardware documentation not only contains text fields to describe a project, such as project title, project description, tools and materials, but it also features original design files to produce the physical parts, such as schematics of PCBs, vector drawings of laser cut parts, source files of 3D models. Similar to open-source software where the source code is accompanied by instructions on how to install, preferably design files come with instructions on how to build or assemble.

Open-source hardware documentation contains text descriptions and design files.

1.2 Motivation

At Fab Lab Aachen, a typical user comes to the lab and creates his project in the one-hour session, leaving the lab without written documentation. Due to the lack of effective documentation system, the fab lab staff will have to create a project page at the internal website and describe the project instead of the author himself. Apparently, it is much harder to document if the documentation is not from the author himself. Currently, an easier way of documentation is used at Fab Lab Aachen. Each physical object made at the lab is taken photograph with a wifi-enabled camera, and the photographs are uploaded to a web gallery simultaneously. However, an obvious drawback is that there is no description for each photograph, so viewers cannot tell what is in the photograph, and the fab lab has difficulty in tracing and aggregating the photographs that belong to the same project.

No project documentation is created at the fab lab in current situation.

We aim to create Fabiji, a photo kiosk with an iPad to facilitate creating and sharing documentation, and to enrich the user experience at the fab lab. The iPad application offers interactive documentation guide to assist users to create project documentation. For example, a progress bar is updated after the user fills in certain documentation fields. The kiosk also assists users to create quality photographs of their physical objects.

Fabiji consists of an iPad application and a photo kiosk.

1.3 Research Questions

- **Why do people create and share project documentation at the fab lab?** In order to facilitate fab lab users to create and share documentation, first we need to know who the fab lab users are, and then we need to understand why they create and share their project documentation.

- **How to create a non-intrusive tool to help documenting physical projects at the fab lab?** How will an iPad application and a photo kiosk help users in documenting physical objects? And how to do that without intruding users?

1.4 Thesis Outline

- **Chapter 2** introduces existing documentation systems for designers and engineers as well as iPad applications for note taking tasks. We summarise this chapter by comparing key properties of these documentation system.

- **Chapter 3** describes a study on fab lab users in order to find out who come to fab labs to work and what their motivation of creating and sharing documentation are. This chapter also raises requirements for the design of Fabiji.

- **Chapter 4** presents the design and implementation of the iPad application of the Fabiji system. The iPad application went through three design iterations: paper prototype, non-functional software prototype and functional software prototype.

- **Chapter 5** presents the design and implementation of the photo kiosk of the Fabiji system. All parts for the structure of the photo kiosk are laser cut, and we use standard DIN materials for other parts. This makes it easy for other fab labs to reproduce a copy.

- **Chapter 6** discusses the final evaluation of the Fabiji system. The final evaluation consists of three parts: *iOS Human Interface Guideline* validation, user study to system requirement analysis. In the end of this chapter, Fabiji is compared with the existing documentation system described in Chapter 2.

- **Chapter 7** summarises the contributions of this thesis and points out possible future improvements and extensions.

Chapter 2

Related Work

Fabiji is a project documentation system specialised for fab lab users who design and create products, and the documentation software is written for iPad. In the first part of this chapter, we review documentation systems for designers and engineers existing in both research and in actual use. In the second part of this chapter, we describe two note-taking applications for iPad.

2.1 Documentation Systems for Designers and Engineers

2.1.1 SHARE

The SHARE [Toye et al., 1993] project at Stanford University continues the vision of the integrated knowledge sharing brought by Vannevar Bush in his article *As We May Think* [Bush, 1945]. SHARE seeks to create an environment for engineers to collaborate over the internet, and carry out daily tasks involving information creating, retrieving, sharing and many more (see Figure 2.1). Such environment enables team members to work remotely with their own tools while avoiding redo of analyses and manufacturing plans due to lack of documentation and information sharing.

SHARE is a vision of creating an environment for engineers to collaborate over the internet.

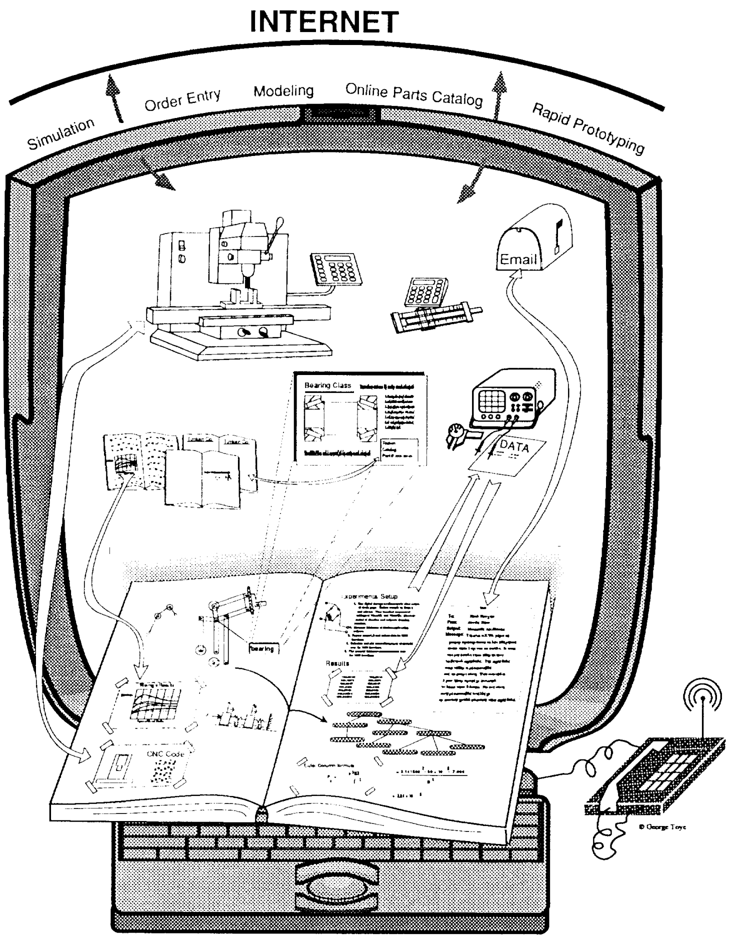

Figure 2.1: The SHARE vision: A collaborated working environment that helps engineers to store and share information and documentation on product *simulation*, *order entry*, *modelling*, *online parts catalogue* and *rapid prototyping*. [Toye et al., 1993]

The initial SHARE environment consists of three main components:

- **Macintosh PowerBook client**: PowerBook is chosen because the testbed users are students from Stanford University, who are familiar with Macintosh based software. Along with other software, FrameMaker was selected as documentation preparation software.

- **A UNIX workstation with email and filer server**: Emails are important medium for communication, so that users are required to log in to the email system at least once a day so that the queued messages are sent out.
- **Two generic document templates**: One template for notes with pre-defined tags and fields such as project title, meeting date, requirement; and another template for requirement reports with structured structured key elements contained in the notes. Templates offer consistent format and require little efforts in navigation through documentation, while it contains subtle danger of limiting users with certain kinds of information.

The next-generation SHARE environment supports working over the internet with various tools and services. KQML Agent [Finin et al., 1994] is the core of the SHARE architecture. Each agent represents the designer, his tools, database or services, and it is responsible for information exchange and representation.

Agents are added in the next-generation SHARE.

Both implementation of SHARE environment are constrained by then available software, and in the end the authors suggest two research directions,

- How designers (engineers) work with knowledge
- How information should be represented and shared

2.1.2 PENS

Within the unifying SHARE [Toye et al., 1993] vision, Hong et al. [Hong et al., 1995] developed PENS, an electronic notebook specialised for engineers to share notes directly. Compared to previous electronic notebook products, it brings the idea of ubiquitous computing [Weiser, 1999] into electronic notebooks to allow engineers to create notes anywhere and collaborate remotely, and eliminates limitations such as requiring network connectivity [Burger et al., 1991, Guzdial et al., 1995] and specialised workstations.

PENS is an electronic notebook that allows engineers to collaborate from anywhere.

Two types of knowledge source: *discourse-based* meetings and *thought-based* notes.

It is industry custom for engineers to create formal documentation throughout their design process. On one hand, due to the lack of immediate benefits, many engineers often do not invest great efforts on formal documentation; On the other hand, engineers write down informal personal notes, but difficult to share them with colleagues. In contrast with knowledge generated collectively from *discourse-based* meetings, Hong et al. introduced the notion of *thought-based* information which describes knowledge generated individually (e.g. personal notes), and characterised it (see Table 2.1). PENS integrates the benefits of both *discourse-based* information and *thought-based* information to capture knowledge in collaborative engineering design.

Attributes/Media Type	**Discourse-based**	**Thought-based**
Primary Audience	others	self
Native Tendency	shared but not recorded	recorded but not shared
Analog examples	conversations	personal notes
Internet example	e-mail	PENS

Table 2.1: Comparison of Discourse-Based vs. Thought-Based Tools [Hong et al., 1995]

PENS consists of client software and a HTTP server.

PENS consists of a client software on a Macintosh notebook computer and a HTTP server. Since Wireless LAN or 3G is not widely deployed in the mid 1990s, PENS developed both two different interfaces of notes to enhance accessibility, one for the notebook computer at workspace where there is no internet access, and one for the browser at home or office or anywhere with internet access. The client software on the notebook computer features an agile browser that displays text in sections and categories, similar to the later Markdown language[1] . In their workspace, Users can use mouse click or arrow keys to navigate through sections and then edit the respective fields. When users return to their offices, they can push the notes to the server with one mouse click using the client software. The client software looks into the updated notes based on pre-defined templates, and sends an HTTP `POST` command to the server, the server renders the notes into HTML and thus can be viewed from anywhere *regardless if they have PENS installed in their computer or not*. This client-server communication method is similar to the later REST architectural style

[1] http://daringfireball.net/projects/markdown/

[Fielding and Taylor, 2002] developed over HTTP.

A testbed was set up during the course project of ME210: Mechatronic Systems Design, a graduate-level engineering design class at Stanford University. A total of 14 teams consisting of 48 students used PENS as their collaboration tools throughout the 20-week project. Since the students of this course live in different places (dormitories, cities, other cities, other states), the geographical distances of team members within each team varies. Results showed that out of 14 teams, 70% of the total usages of PENS come from three most geographically-remote teams (where students are from different states). The results suggest the following three statements:

A testbed was set up in parallel of an engineering design class.

- PENS was more appealing to remote teams compared to local teams
- The number of documentation post can be used as quantitative assessment
- Multiple platform clients are needed for PENS

2.1.3 EEN

Over the years, traditional paper engineering notebooks are still popular compared to electronic notebooks because they are easy to read, to write and to sketch on. EEN (Electronic Engineering Notebook) [Gwizdka et al., 1996] tries to replace traditional paper notebooks while acquiring information in an non-intrusive way.

EEN aims to replace engineer's paper notebook in an non-intrusive way.

The authors claim that it is essential to capture and store conceptual design information, because engineering design constantly needs previous experience of similar design or similar design rationale. Yet in current design practices with paper notebooks, there are three disadvantages regarding conceptual design information,

There are three disadvantages regarding conceptual design information with current design practices.

- Conceptual design information recorded on paper is difficult to share to other designers

- Intermediate designs are only available on human memory, hence it is time consuming and unreliable
- It is hardly effective or efficient to share design information in complex projects

In order to solve the above mentioned problems, the authors propose four main functions to an electronic engineering notebook,

- *Information acquisition*: Capturing user input, including handwriting and sketch
- *Authoring*: Organising and structuring information
- *Browsing and navigation*: Use tags and forms to enhance accessibility
- *Information transfer*: Hyperlink annotations on tags and nodes and pages

Results show that it is not significantly different to read or to sketch on EEN compared to paper, but writing is less comfortable on EEN.

A prototype EEN was implemented on Apple Newton MessagePad[2] , and two usability studies and one controlled experiment were conducted. Results showed that, while paper was still better to write on than EEN, there was no significant difference to read or to sketch. Writing on touch screen glass lack of physical feedback that paper provides, thus users feel less comfortable writing on EEN than on paper.

2.1.4 Ars Electronica Fab Lab

The fab lab at Ars Electronica Center[3] in Linz, Austria serves as an interactive exhibition space for the Ars Electronica's museum [Posch et al., 2010]. As the main users of Ars Electronica Fab Lab are children, the lab focuses on creating a learning environment and providing easy accessibility to the realised work created at the fab lab.

[2] http://docs.info.apple.com/article.html?artnum=28047
[3] http://www.aec.at

Based on the previous work on creating interactive content [Lindinger et al., 2006], Ars Electronica Fab Lab is designed to have three main areas,

Ars Electronica Fab Lab focuses on creating a learning environment and providing easy accessibility for children.

- **Design Area** comprises various tools to support digital 3D models creation, including installations to allow freeform 3D design by moving around a physical object.
- **Fabrication Area** includes production tools that can realise the design into production. The tools include an A3 Printer, a laser cutter and a fab@home[4] 3D printer.
- **Gallery** showcases both physical objects that were created at the fab lab as well as a virtual digital representation of models that were designed.

The authors argue that the combined three areas within the fab lab create an educational environment for *creative prototyping* and *shared creativity*. Creative prototyping is achieved by giving freedom to visitors, allowing them to freely use all provided tools while providing certain guidance and routines for less experienced users (e.g. first time visitors). Gallery with both physical and virtual works provide users with inspirations, and the authors claim that children and adults are greatly motivated by leading examples or the things they like. Ars Electronica Fab Lab deliberately opens its door to all passers-by, which provides an opportunity for every museum visitor to have a glimpse of what is going on inside the lab, and thus brings diverse visitors to the fab lab.

The combination of design area, fabrication area, and gallery creates an educational environment for creative prototyping and shared creativity.

2.1.5 FabML

Määttä et al. [Määttä and Troxler, 2011] argue that it is relatively easy to share project documentation at the fab lab locally but there is a missing link to share it cross fab labs all over the world. The authors propose using a standard

[4] http://www.fabathome.org

exchange format for describing open source hardware documentation and share them across fab labs.

Existing online communities do not qualify as documentation systems for fab labs.

There are many existing online communities, such as Instructables[5] and Thingiverse[6] , for open source hardware projects, but there are some drawbacks so that they cannot serve as the documentation system for fab labs. First, these websites are independent from fab labs, and they are often commercial; Second, they are not offered as an open source package where each fab lab can install and customise according to its own needs; Third, it is questionable whether the task of knowledge sharing of fab lab network should depends on a third party.

FabML aims to be open source hardware project documentation format.

FabML, an XML file format was brought up in FAB6[7] conference in August 2010, it defines a minimal set of vocabularies that are used to describe an open source hardware project as well as the linked media files. The goal of FabML is that it can be created and shared regardless of documentation system used.

Among fab labs prototyping documentation platforms, ProtoSpace[8] (or Fab Lab Utrecht) use two dedicated computers as kiosks and a documentation website powered by Drupal[9] content management system. One kiosk is equipped with checkin and checkout functions to keep a record of the people who are in the lab as well as what they are currently working on, another kiosk is used for creating documentation for the Drupal website.

Content management systems such as Drupal can easily generate and aggregate customisable RRS feeds which include FabML, but the challenge is that, some important rich media file types included in FabML (such as `.STL` files for 3D design) are not currently included in *Internet Media Types* (or *MIME types*), and thus it is difficult to share over the internet.

[5]http://www.instructables.com
[6]http://www.thingiverse.com
[7]http://www.fab6.nl
[8]http://www.protospace.nl
[9]http://drupal.org

2.2 Tools and Devices for documentation

Apart from the above mentioned documentation systems, designers and engineers have created various tools and devices to record their design process. Not only text and files have been recorded, pictures and videos are also experimented as part of project documentation.

2.2.1 ThingDoc

ThingDoc[10] is a parser that translates a syntax specialised for describing bill of materials into various human readable formats, such as HTML, $\TeX$ or wiki documentation. Users need to create a text file using a syntax similar to *Javadoc*, and then use a python script to generate the bill of materials in desired format (see Figure 2.2).

ThingDoc generates bill of materials.

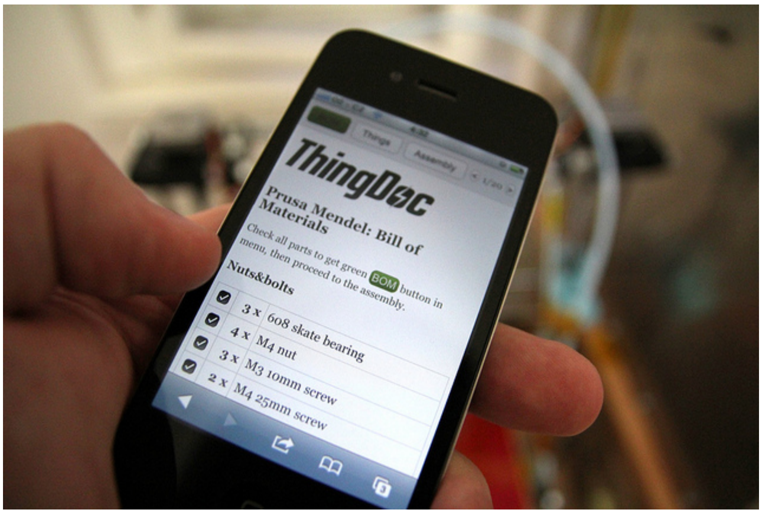

Figure 2.2: ThingDoc viewed in browser: bill of materials can be generated in many formats including HTML. Photo from thingdoc.org

[10] http://thingdoc.org

2.2.2 Timeline Tools

Timeline Tools takes pictures throughout designing process.

Timeline Tools[11] is a toolbox which automatically takes a picture whenever a tool is taken out or brought back to the toolbox, and it then compile these pictures into twitter[12] timelines. The user can also take pictures manually by pressing a button when needed. Timeline Tools keeps record of the design/engineering process, which happens in a fixed area such as a workbench with pictures. It focuses on the recording work in a certain period of time at a certain place.

2.2.3 HandMade

HandMade records video of handwork.

HandMade[13] is a wearable device for recording videos of handwork. Similar to the idea of *helmet camera* such as GoPro[14] , HandMade captures a wide angel view from a first-person perspective. The user wears a apron with a camera that recording users hand and the handwork currently being carried out.

2.3 Note-Taking Applications for iPad

Currently there is no application for iPad that specialised in open source hardware project documentation for fab lab users. But there are many applications available for iPad to create personal notes. Here we briefly describe two of the most popular applications among them: Evernote and Springpad.

[11] http://www.jackchalkley.com/portfolio/timeline-tools.html
[12] http://www.twitter.com
[13] http://www.distancelab.org/projects/handmade/
[14] http://gopro.com

2.3.1 Evernote

Evernote[15] is a powerful cross-platform note-taking application, it aims at capturing anything, accessing it anywhere, and finding things fast. Evernote supports desktop, tablet computers, and web access. A note can include text, audio recording, photos, links, screenshots, website snapshots, and many others. It is organised by being categorised into notebooks and tags. Notes created on any device can be synchronised though Evernote server. Another selling point of Evernote is that users can also share notebooks to others or link shared notebooks to their own libraries.

2.3.2 Springpad

Springpad[16] is a smart notebook application that allows users to quickly add notes and automatically add relevant information available on the internet. Similar to Evernote, Springpad is also available on many platforms (including iPad), and it can also synchronise notes automatically across devices. As a smart notebook application, Springpad features in-app reminders and alerts, which will send notification to users in due time. Springpad also features password protection on private notes, so that even if a user keeps his account logging in on the iPad all the time, the password prevents someone else reading his private notes.

2.4 Documentation Systems at a Glance

Table 2.2 presents a comparison of the key properties of the eight documentation systems (excluding iPad applications) discussed in this chapter. We analyse them regarding mobility, easy to install, size, ownership, target users.

[15] http://www.evernote.com
[16] http://springpadit.com

No.	Project	Mobile	Easy to install	Size	Ownership	Target Users
1	SHARE	✓	✗	Medium	Personal	Engineers
2	PENS	✓	✗	Medium	Personal	Engineers
3	EEN	✓	✗	Small	Personal	Engineers
4	Ars Electronica Fab Lab	✗	✗	Large	Shared	Fab lab users
5	FabML	-	✓	-	-	Fab lab users
6	ThingDoc	✓	✓	Small	Personal	Engineers
7	Timeline Tools	✓	✗	Medium	Personal	Designers
8	HandMade	✓	✓	Small	Personal	Craftsmen

Table 2.2: Comparison of existing documentation systems (Some properties of FabML are not listed because it is a file format).

Chapter 3

Initial Study

In order to facilitate fab lab users to create and share documentation, first we need to know who the fab lab users are, and then we need to understand why they create and share their project documentation. We answer these questions by carrying out contextual inquiry on fab lab open days, and asking users to fill out a follow-up questionnaire and discussion with them. Then we use the qualitative data to identify requirements for design of Fabiji. In order to study the current working environment and process of fab lab users, we did not bring any prototype during the initial study.

Initial study consists of contextual inquiry, questionnaire and discussion.

3.1 Contextual Inquiry

In this section, we explain which participants we choose, and describe how we execute contextual inquiry as well as showing our observations.

3.1.1 Participants

Since Fabiji focus on fab lab users, we avoid users from within the Media Computing Group, and eligible users are only those who come to work at the fab lab on Tuesday

All participants are visitors on Fab Lab Open Days.

open days. Since Fab Lab Aachen open its doors to every one, the users can be of various backgrounds, occupations, and age groups. But in order to work at the fab lab, they must have their design work done in advance and bring the design to the fab lab for fabrication. The study lasted 6 weeks (only on each Tuesday afternoon), and 22 users participated our study. The participants are aged 20-58 (M = 28.6, SD = 8.9), and among them there are four females.

3.1.2 Execution

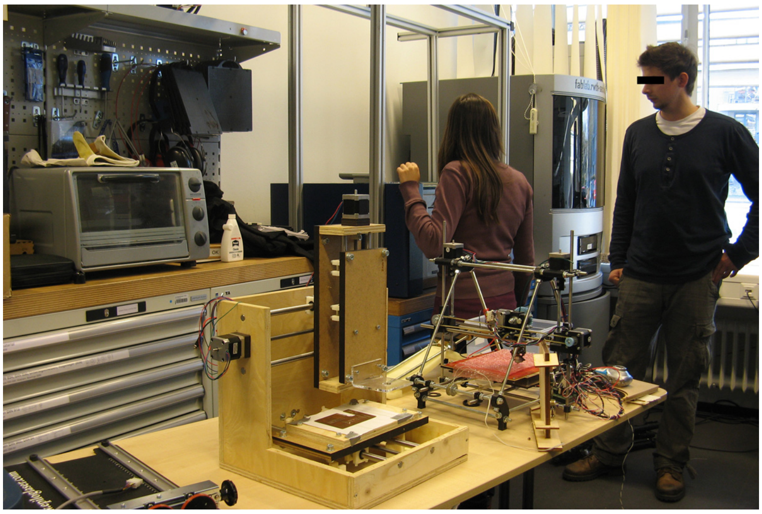

Figure 3.1: A technical staff at Fab Lab Aachen (left) is assisting the user (right) to operate the Epilog Zing laser cutter.

contextual inquiry takes place at the fab lab, which is the users' actual workspace.

The contextual inquiry technique [Wixon et al., 1990] defines the context of the study is in the user's actual workspace. In our case, the user's actual workspace *is* the fab lab. Before the contextual inquiry, we explained our purpose of study, and also asked the user to speak out what he was doing. During the contextual inquiry, we observed the user interaction with machines and fab lab staff. We also discussed with users about the documentation system in their minds. We requested the permission of the user to audio record the discussion.

Observations

During the observation and discussion, we discovered the following characteristics that users implied when they were working at the fab lab.

- **Users need technical assistance**: First time users often have no experience how to use a certain machine (see Figure 3.1), and even users who have previous experience using a certain machine may not be familiar with the software that is installed in the fab lab computers. For example, we found that users had to adjust their finished vector files before they were sent to the laser cutter, because the files were not fully compatible with the vector editing software in use at the fab lab. The technical staff also helps the user to choose the correct parameters for machines based on different materials, and these parameters come from documentation accumulated for a long time.

- **Users do not want to spend much time on documentation**: The purpose of users visiting the fab lab is to use the machines that they usually do not have access to. Each time a user wants to come to fab lab to work, he has to schedule a few weeks in advance. Furthermore, each user can only have a scheduled session of one hour. Thus the user really wants to spend more time with the machine to make sure it creates the exact thing that he had designed or planned. The user will feel like creating documentation only when he is certain that the machine will finish its job in time.

- **Users have free time when machine is working**: Although the time that users spend at the fab lab is limited, users do have free time when they have to wait for the machine to finish work. For example, laser cutting a piece of 30cm $\times$ 60cm medium-density fibreboard (MDF) takes up to 20 minutes, although first time users may find it a novel experience to watch laser cutter working and staring at the laser beam most of the time, experienced users may not find it that attractive and try to find some thing else to do.

- **Users like to explore the fab lab**: Whenever users have free time (either because of the machine is busy working or because today's work is done), they like to look around and examine the things people made at the fab lab. Users show great interest in browsing other people's projects.

- **Regular users are connected as a community**: We notice that many frequent users are also members of dorkbot[1] , where designers and engineers meet to all sorts of things with electronics, and these users are hackers who create and share the most number of open source hardware projects. However, compared to the number of fab lab users, dorkbot is relatively small. As a result, some users wish to join such a community but do not know such community exists. For instance, one user even suggested creating a web forum for fab lab users.

User Profiles

Based on our observations and discussions, we identify four typical user groups of fab lab users (see Figure 3.2). Note that we only list the typical users, which means not every user is necessary belong to one of the four groups here.

- **Hackers** create great open source hardware projects, they are open-minded and they know well about open source and new technology. They usually have no time or they are lazy to create documentation, but they are willing to if there is an dedicated application which is easy to use.

- **PhD students** or **Researchers** come to the fab lab to build parts for their research projects. Due to years of scientific practice, they have a good habit of documentation and it is required to share documentation with their own research groups. They may have

[1] http://dorkbot.org

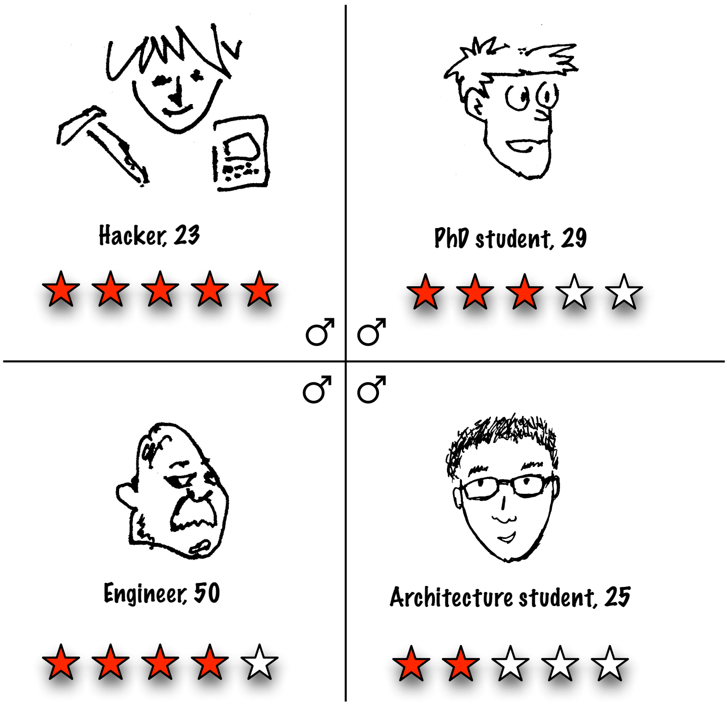

Figure 3.2: Four typical user groups at the fab lab: Hackers, PhD students or researchers, engineers and architecture students. The number of stars indicates their degree of involvement to the fab lab.

heard of the term *open source hardware*, and may try to build some personal projects in the future. Generally they do not have time to edit and upload documentation, and they are only interested in similar research projects.

- **Engineers** work at the fab lab because it is cheap and they want to talk to other people at the fab lab, to share and to discover interesting projects. They create private projects for their own needs. They like the idea of open source hardware, and they want to see open source hardware projects created by others, but they are reluctant to share their own projects.

- **Architecture students** come to the fab lab to finish school projects because laser cutting architecture models at the fab lab is a cheaper alternative to professional services. Most of architecture students haven't heard of open source hardware. They don't want to share their projects because they are part of homework assignments or master thesis, however, they would like to build portfolio documentation to showcase their work in order to find employment opportunity. As of sharing their work, some of them are concerned of intellectual property issues, that is, someone else might *steal* their ideas.

3.2 Questionnaire

questionnaire was filled out after the contextual inquiry.

After the contextual inquiry, which was carried out during the user's working session, we asked the user to fill out a questionnaire to collect qualitative data. We want to know the user's preference of project documentation, and we especially care about the motivation for fab lab users to create and share project documentation.

3.2.1 Questionnaire Design

Questions are selected from a survey for online DIY community users.

Kuznetsov et. al [Kuznetsov and Paulos, 2010] carried out a large-scale study of online DIY communities in 2010, 2600 users of six major online DIY communities participated in their survey. The survey focuses on the role of collaboration technologies in motivating and sustaining DIY communities. Although target users are different (fab lab users are not necessary online DIY community users, and vice verse), one of the goal of both studies is the same: what is the motivation of users contributing project documentation? As a result, during the design of the questionnaire we extracted two related questions from their survey and used in our questionnaire (Q15 and Q18). The full questionnaire can be found at Figure A.1 - A.3.

3.2.2 Results

Since there are many professional fabrication services that can also do tasks that a fab lab does: such as Ponoko[2] for laser cutting, Shapeways[3] for 3D printing, and BatchPCB[4] for PCB board printing. Often time, these professional fabrication services have better machines and thus do better on a certain task than what a fab lab is capable of. We asked the participants the reasons why they choose to work at the fab lab instead of those professional fabrication services. As shown in Figure 3.3, most users take the advantage of free entrance of Fab Lab Aachen. The other two main reasons are that, users want to meat creative people, and users want to learn to use fab lab machines. Some users also suggested that it is more convenient to use fab lab services than professional services. For example, one user commented, "*I only need to make an appointment online, and then come directly to the fab lab to work*", and another user commented "*I have to wait for the delivery if I use Shapeways... I can talk to the staff here in the fab lab to avoid unnecessary mistakes*".

People come to the fab lab to work because it is cheap and convenient

Generally, the participants would document their projects (Q11: 59% answered YES), and the majority of participants (Q13: 85% answered YES) would share their projects with others. However, since not every tester came to the fab lab to create open source projects, it is not sufficient to conclude from these two figures alone that the majority of users are *willing* to share their project documentation. As shown in Figure 3.4, while over 76% of users came to the fab lab to create the current project for hobby, for fun and for other personal use, 28% of users created things at the fab lab for their research or study. And among the latter, all of them are used to document in their work or study (Q10). This implies the purpose of user creating documentation can be of other reasons than simply willing to share to others.

Participants would like to document their projects.

As shown in Figure 3.5, The most dominating reasons for participants to create and share project documentation are "*Give back to the community*" (57% strongly agree, 28% agree), and "*Receive feedback about my own projects*" (43%

[2]http://www.ponoko.com
[3]http://www.shapeways.com
[4]http://batchpcb.com

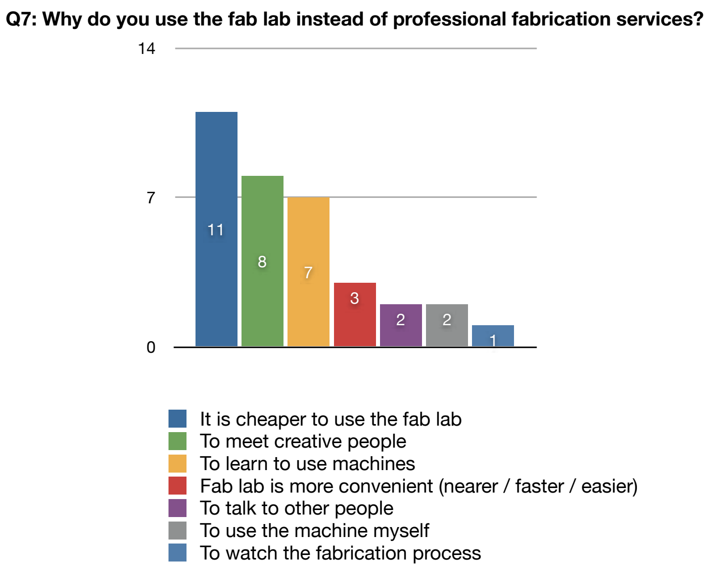

Figure 3.3: The reasons of using the fab lab instead of professional fabrication services.

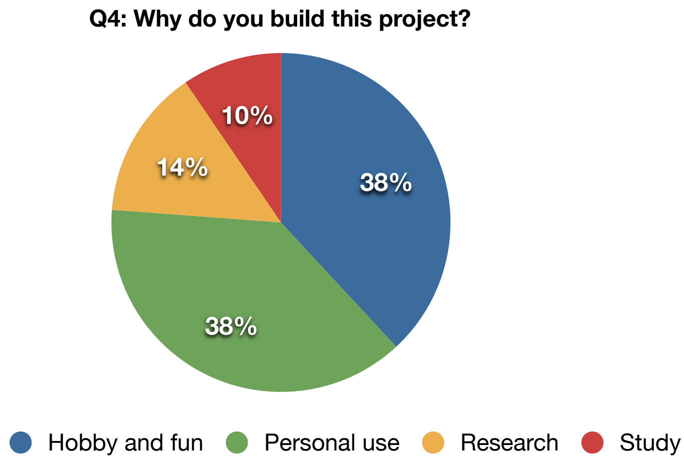

Figure 3.4: The purposes of building current project at the fab lab. Most users work on projects for hobby, for fun or for other personal use.

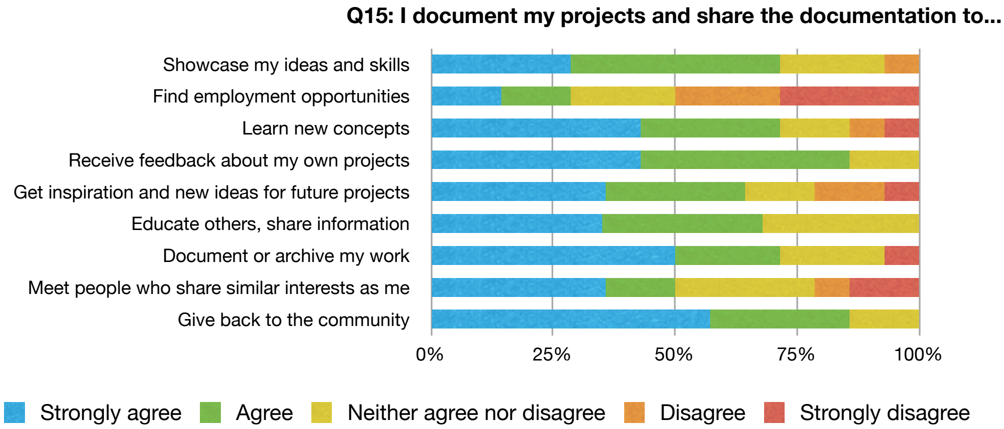

Figure 3.5: Motivation for documenting projects and sharing project documentation

strongly agree, 43% agree). And among all possible answers, "*Give back to the community*" receive the highest number of participants who choose "strongly agree". Other reasons include self learning (71% agree "*Learn new concepts*" and 64% agree "*Get inspiration and new ideas for future projects*"), altruism (66% agree "Educate others, share information", 71% agree "*Showcase my ideas and skills*") and Documentation for archive (71% agree "*Document or archive my work*"). Only 28% people agree "*Find employment opportunities*", which makes it the least agreed answer. Although only 50% people agree that "*Meet people who share similar interests as me*" is the reason why they create and share documentation, among those who agree the statement, 71% strongly agree, which is the highest strongly agree percentage all possible answers.

Reasons for participants to create and share documentation vary.

Lack of time is the primary reason for participants not sharing their work (see Figure 3.6), and "*My projects are too easy or simple*" being the second most agreed answer. Other common deterrents are the users negative views at their projects (not interesting, not novel or creative). Note that although only 21% people agree on the answer "*I don't want other people to steal my ideas*", 14% strongly agree it. No tester agrees the answers "*I don't want my work to be critiqued*" or "*Give back to the community*", which is corresponds to the results in Q15 (83% agree "*Receive feedback about my own*

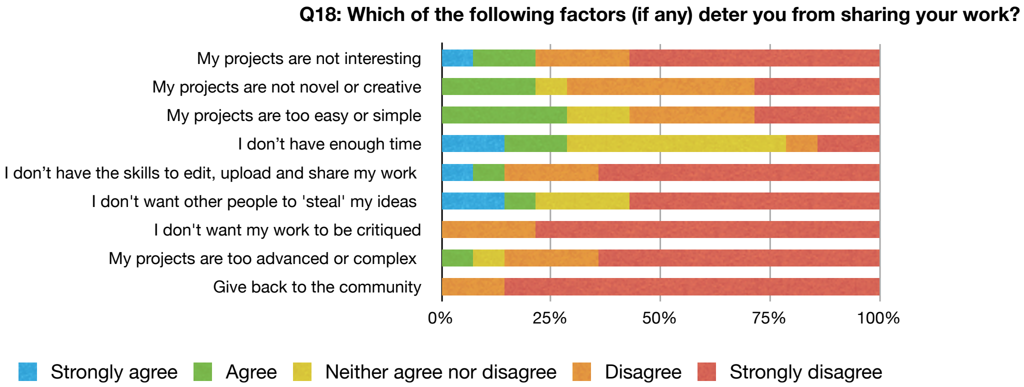

Figure 3.6: Deterrents for sharing work

projects", and 85% agree "*Give back to the community*").

Most participants prefer pictures and text to be the contents of documentation.

While pictures (93% agree) and text (86% agree) are the most preferred forms of documentation, 36% show preference with video (Q12). Surprisingly, no test prefer using voice recording to create their documentation, because voice recordings are "*difficult to index or search*", and they are often "too long to find the interesting part". One user mentioned he would prefer podcast, because he can listen to it on his way to work or home. Other preferences of documentation forms include handwriting and technical drawings.

Most participants share documentation in their familiar environment.

Figure 3.7 show places where participants currently share their project documentation (Q14). Most participants (73%) share their project documentations in their familiar environment (interest groups or internal website). According to the discussion after users filling out the questionnaire, we found that some users chose this answer because they are PhD students or researchers who are required to publish or share their projects with their colleagues. Personal websites or blogs are the second popular place to share while online DIY communities are listed as the third.

82% participants have interests in looking at other people's projects created at the fab lab (Q16). The most answered reason for that is to get inspiration, one user indicates that hey may collaborate with other people if he finds a really

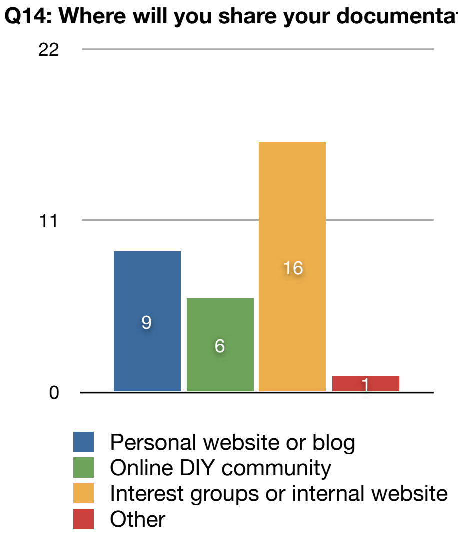

Figure 3.7: The places where user share their documentation.

interesting project. The other 18% participants who have no interests in browsing other people's projects are mostly researchers who come to the fab lab only to fabricate things that they need for their research, and thus they are not interested in other people's projects unless directly related to their research.

FORK IN OPEN SOURCE HARDWARE:
In software engineering, fork means copying source code from a software package and then modifying it or developing independently on it. Fork usually happens in open source software where people are allowed to obtain, modify and republish the code according to its original license. Similarly, in open source hardware, fork means copying source code and/or hardware design and then modifying it or developing independently on it. The term *fork* was first by Eric Allman in 1980 to describe branches: *creating a branch "forks off" a version of the program*.

Definition:
Fork in open source hardware

As shown in Figure 3.8, participants generally welcome receiving comments and feedback on their projects (Q17).

More than half participants also expect others to rate, replicate and fork their projects, and the latter two are the essence of open source hardware. However, some users think it is not a good way to criticise projects only by rating, because no further feedback can achieve. Other users are afraid that by allowing forking, other people might steal their ideas (Also shown in Q18).

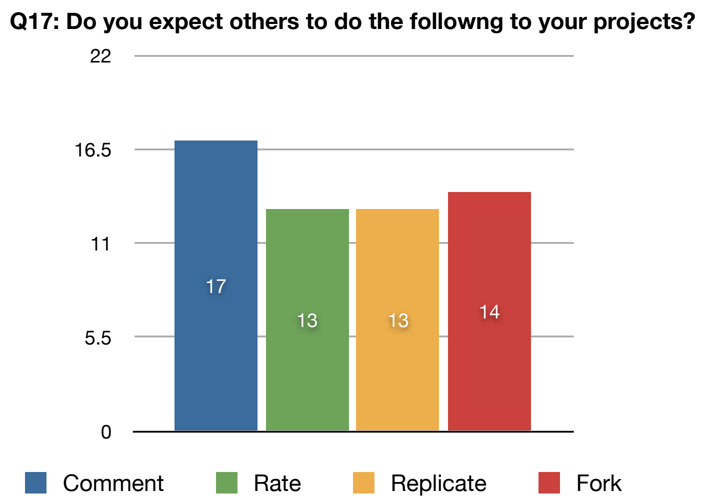

Figure 3.8: User expectation of others commenting, rating, replicating and forking their projects.

3.3 Findings

3.3.1 Motivation for Creating and Sharing Documentation

Each user group has its own motivation.

In the contextual inquiry, we discover that there are four typical user groups in the fab lab, and each group has its own goals of using the fab lab and motivation of creating and sharing project documentation. Here we answer the first research question raised in Section 1.3 by analysing each of the four user groups.

Hackers

Hackers work at the fab lab because they can create DIY projects here. Since most of the projects they create are open source, generally they are happy to create complete project documentation (including design files, source code and schematics) and share them at dorkbot meetings or online communities, so that other people will see their projects and leave feedback or even collaborate with them. Some hackers are reluctant to create documentation for certain projects, because they think their projects are of not great use to other people. Other hackers find documentation is time consuming and would rather skip it before the project is well known. Nevertheless, if a hacker decides to create the documentation for a project, he most definitely will share the documentation.

Hackers share documentation because they love open source.

PhD students or Researchers

PhD students or Researchers come to the fab lab because they need something for their research, and the fab lab is happen to be in the university. They document the things they made because it is part of their research projects, and thus it is scientific practice to document. But they will only share their documentation within their research groups, because research projects usually are not shared before the corresponding research papers are published.

PhD students and researchers create documentation for scientific routine.

Engineers

Engineers regard the fab lab as a public workshop where they can find tools and machines they don't have at home or work. Since they create private projects with hope that some day it may turn into business, they are reluctant to share their project documentation. Because of the same reason (that one day the project may bring profits) as well as years of engineering practice, most of the time engineers will fully document their projects.

Engineers document projects because of engineering practice.

Architecture students

Architecture students document projects for building portfolio.

Architecture students come to the fab lab mainly to create architecture models for their school projects. They usually generate 3D rendering of their models in the computer, and use DSLRs to make good photograph after the models are created physically. They document their projects because they want to build up their portfolio to prepare for employment after university. They are willing to share the rendered images and final photographs, but they normally don't share their design to prevent other people from reproducing them.

3.3.2 Requirements for the Documentation Tool

By collecting the results from the questionnaire and discussion with the participants, we propose the following requirements for the documentation tool.

- **R1: Help users to create simple documentation in a short time non-intrusively**: The documentation tool will not replace technical assistants at the fab lab, but it can ease documentation process with user-friendly interface.
- **R2: Encourage users to create documentation when they are at the fab lab**: For experienced users, they have plenty of time in the fab lab when the machine is working, the documentation tool will encourage the experienced users to create project documentation.
- **R3: Encourage users to explore other's projects when they are at the fab lab**: The tool can attract people's eyes and encourage them to explore the existing projects.
- **R4: Give users opportunities to meet in real life**: People, especially open source projects creators, like to meet others who share the same minds. The tool should allow users to interact with each other in real life through the project documentation they made.

- **R5: Help users to take better photographs**: The key of a good photograph is a clean background, good lighting, a good viewpoint, and accurate focus.

- **R6: Easy to deploy and configure for other fab labs**: The hardware can be made in other fab labs with existing machines and material, and the software should be easy to install and configure.

3.3.3 The Solution

In order to fulfil the above requirements, we find embedding an iPad[5] into a photo kiosk is the solution for the documentation tool.

Compared to a desktop or a notebook computer, an iPad (specifically iPad 2 or the new iPad) is equipped with a screen, a camera and it can be typed on directly, so that no external screen, camera, or mouse and keyboard is needed. As a result, the total cost is much cheaper, and it also saves much space to use an iPad. The limitation though, is that the iPad doesn't have support USB drives, so that users cannot bring their source files with USB sticks, the files need to be transferred to the iPad via WLAN. There are also commercial applications of iPad as a kiosk to speed up entering information, for instance, TD bank deployed iPad Kiosk at the Buffalo Airport as self-service point for credit card application[6].

iPad has every thing needed for creating project documentation.

The photo kiosk with LED lighting and white background provides users with a good place for photographing the things they made at the fab lab. The lighting and the view angle that the photo kiosk offers will be just perfect for photographing most objects created with laser cutter, 3D printer or the PCB milling machine. With an iPad embedded into a photo kiosk, one may argue that if something is too big to fit in, the iPad camera cannot capture the whole image of the object. So the iPad should be easily removed

Photo kiosk assists users to take better photographs.

[5] http://www.apple.com/ipad/

[6] http://www.pivotdesigngroup.com/work/casestudies/td-visa.html

from the kiosk so that users can take it around freely when needed.

Another advantage with iPad is that it is easy to install and configure software. With Apple App Store[7] , any fab lab or even anyone who has an iPad can download and install the App (or software) by a few taps.

[7]http://www.itunes.com/appstore

Chapter 4

iPad Application

In this chapter, we describe the design and implementation of the Fabiji iPad application. We went through three design iterations: a paper prototype to discover the main functions we need, a non-functional software prototype to validate the user interface and a functional software prototype to realise the design.

4.1 Paper Prototype

Snyder [Snyder, 2003] has defined four dimensions of a prototype: breadth, depth, look, and interaction. *Breadth* is the percentage of the functionality of a product that is covered in a prototype. *Depth* describes how deep that the function is implemented beneath the user interface. *Look* refers to how accurate the visual representation is. *Interaction* is that how a prototype responds to the user input and what feedback it gives. According to these dimension definitions, our paper prototype is broad and shallow. Since the paper prototype is hand drawn, its look is rough and only represents the basic layout on the iPad. Not much interaction is covered in this paper prototype, but rather demonstrated in the first software prototype in Section 4.2.

We created a broad and shallow paper prototype.

4.1.1 Brainstorming

First we list all possible fields for hardware documentation.

We start designing our paper prototype by listing fields needed for hardware documentation and the features needed for a documentation system at fab labs (result shown as a mind map in Figure 4.1). Since a complete project documentation not only allows people to understand what the physical object is and what it is used for, but also includes everything needed to reproduce the physical object. Therefore, the fields of a project documentation may include text, photo, voice recording, video, 3D scan, license, tools, parameters, tags, parts (bill of materials), raw files (e.g. design source files) and source code.

We derive all possible features for the iPad application based on the documentation fields.

With these information in the application, users first schedule a meeting online, and the application knows who the currently user is based on the schedule. If a user has his design files stored in the documentation already, he can send the task to the respective machine (e.g. laser cutting task is generated based on the 2D vector files in the documentation). After finishing the work, users can upload the documentation to their email, Evernote, or Dropbox[1] , and publish their documentation at Thingiverse. Users can also explore and comment other people's documentation. The original author will receive email notification if someone commented his project. To meet the requirement *R4: Give users opportunities to meet in real life* in Section 3.3.2, we create a storyboard to show a scenario where users can discover interesting projects and people with Fabiji (see Figure B.1 - B.2).

4.1.2 User Interface

Since we develop the application on the iPad, it is essential that the user interface must be simple intuitive. We identify the main features from the brainstorming: Explore all projects, find scheduled projects, create a new project, and edit existing projects. And this results in five main views in the user interface (see Figure B.3 - Figure B.6):

[1] http://www.dropbox.com

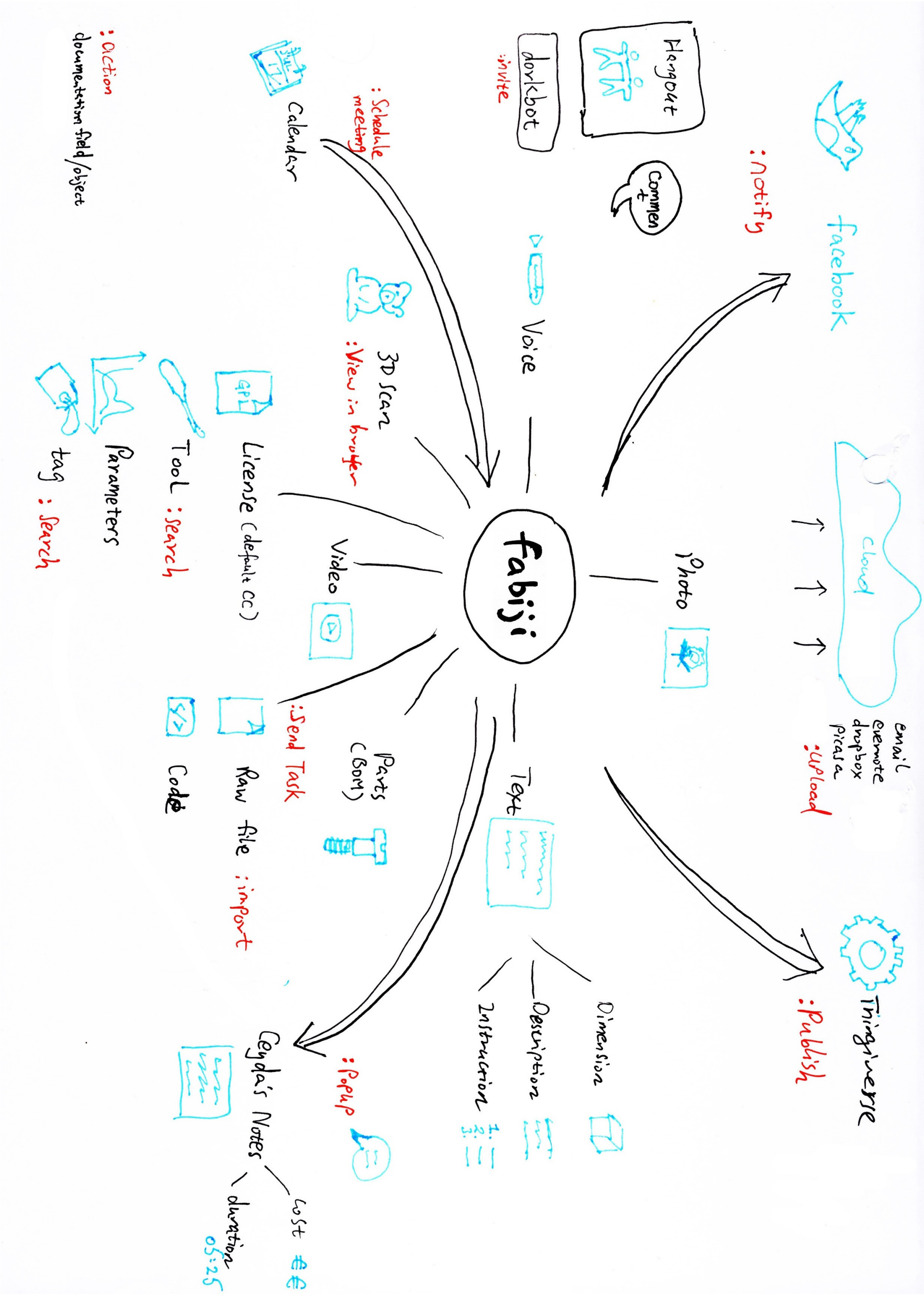

Figure 4.1: Mind map of features and functions in the Fabiji iPad application.

- **Today**: List all scheduled appointments and allow users to start project from the appointment as well admins to edit the appointment,
- **All**: List all finished projects.
- **Inspire**: Shortcut to explore projects.
- **Tags**: Tag cloud of all projects
- **User**: Since there can be several users working at the fab lab at a time, the application needs to support multiple login. Our initial idea is to create a tab view for each user, so that it is clear to see who are currently using the application.

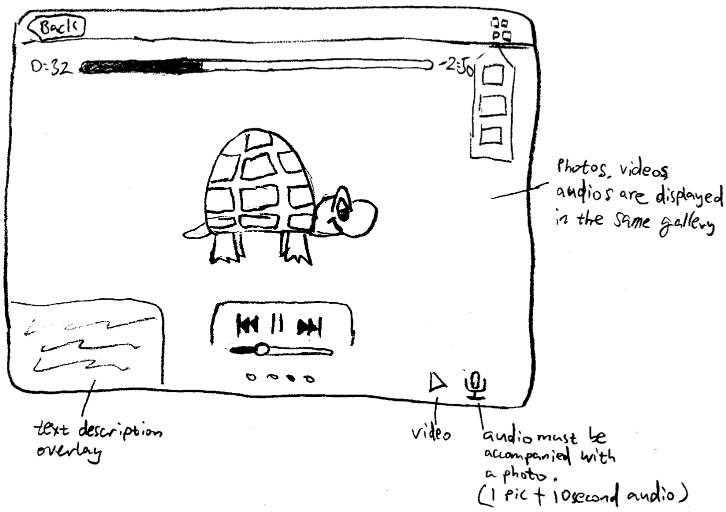

Figure 4.2: Gallery view that supports photos, videos and audio recordings.

We come up with the concept of Photo Story.

Since we support multimedia types (photos, videos, audios) in the documentation, and each project may have any combination of these multimedia types. We integrate all media types into a single gallery view, and with the support of gestures, users can simply swipe though all photos, videos and audios attached to the documentation (see Figure 4.2). Since users disliked typing on the touchscreen (see Nielsen Norman Group's reports on iPad Usability[2]), we

[2]http://www.nngroup.com/reports/mobile/ipad/

introduce *Photo Story*. A photo story is a picture accompanied by a short audio recording, while the audio recording can either be a few sentences to describe the photo or ambient sound recording the machine work in action. This reduces users burden to type on touchscreen, but the trade off is that users will have to record audio (including their voices). We need to verify this with further user study.

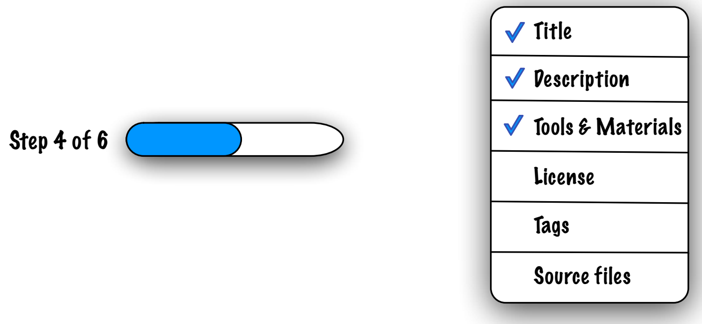

Figure 4.3: Progress bar comparison: With percent-done progress indicator (left), and with checkmark progress indicator (right).

To meet the requirement *R1: Help users to create simple documentation in a short time*, we use a step-by-step project creation guide (see Figure B.5 - B.6) suggested by BJ Fogg [Fogg, 2003]. The guide consists of six steps, including the basics fields of a hardware documentation: project title, project description, tools and materials, dimension, parts, license, tags, and source files. Except the project title in the first step, other steps can be ignored by tapping "next" button, and the user can always edit the documentation later. The step-by-step guide is designed to finish in five minutes or less. After completing the steps in the guide, a basic hardware documentation is created, so that even users who forget or do not want to edit the documentation later have already create a useful documentation.

We use a step-by-step project creation guide.

When designing the progress bar for the step-by-step project creation guide, we first came up with two designs, one with percent-done progress indicator, and another with checkmarks on finished items so far (see Figure

We also use a percent-done indicator.

4.3). Percent-done progress indicator offers a clear visual feedback about the current progress while the checkmarks tell users what fields have been completed and what fields haven't. As the step-by-step guide aims for creating in a short time, time affordance [Conn, 1995, Myers, 1985] is more important than the names of completed fields, so we decided to use the progress bar with percent-done indicator.

4.2 First Software Prototype

We created a non-functional software prototype using OmniGraffle.

While an iPad application is rich in user interaction in terms of multitouch gestures, device orientation and rotations, camera and audio recording, our paper prototype offers little user interaction, and it is difficult to test with users without implementing interactions. Therefore, we create a software prototype using OmniGraffle[3] (a diagramming application for Mac OS X) with simulated iPad user interface and deeper functionalities, and then we export the prototype in interactive PDF with internal links and view it in GoodReader[4] (a PDF reader application for iOS). In this way, users can navigate through the prototype by tapping the buttons containing links to different pages.

4.2.1 Tools

With third party stencils, it is easy to create iPad UI mockup in OmniGraffle.

OmniGraffle offers a great feature called *stencil*, which is an extension for OmniGraffle that consists of a collection of shapes, and users and create their own stencils. We use an open source stencil that has most UI elements in iOS, and with simple drag and drop we can create an iPad screen easily. To simulate tapping actions, we embed links into buttons, search bars, and other tappable UI elements, so that when a user clicks on a button, the embedded link will navigate the current page to another page. Similarly, if the prototype is viewed in GoodReader on the iPad, when a user taps on a button, the embedded link will navigate the

[3]http://www.omnigroup.com/products/omnigraffle/
[4]http://www.goodiware.com/goodreader.html

current view to another view, as if the button is really working. We are unable to support other gestures such as swipe in this software prototype due to the limitation of PDF.

4.2.2 Design Changes

We first implemented the OmniGraffle software prototype based on the sketches of the paper prototype described in Section 4.1.2, and then tested the software prototype on the iPad with five fab lab users aged 20-28 (two of them have iPad experience). We asked the users to carry out the following two tasks with think aloud technique [Lewis and Rieman, 1994], which allows us to find critical problems in both interaction and user interface design.

Several design changes are made after an informal user test.

- Task 1: Create project documentation for the 3D printed turtle that you are scheduled to work on today, and add photographs afterwards.
- Task 2: Suppose that you are Michael, switch to your project page.

To test step-by-step guide and multiple login functions, we end the user test with a discussion focusing on two questions: *Do you prefer step-by-step guide or project editing page*? *Are you afraid that other people might edit your projects if there is no login required*?

After reviewing the qualitative feedback from the user test, we made the following design changes (see Figure 4.4).

- Use *My Projects* tab instead of multiple user tabs: During the user test we found that tab bars are the last places for users to tap, users should be able to finish most their work in the current tab. *iOS Human Interface Guideline*[5] also suggests that "on iPad, avoid crowding the tab bar with too many tabs" because too many tabs not only make it difficult for users to tap

[5]http://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/

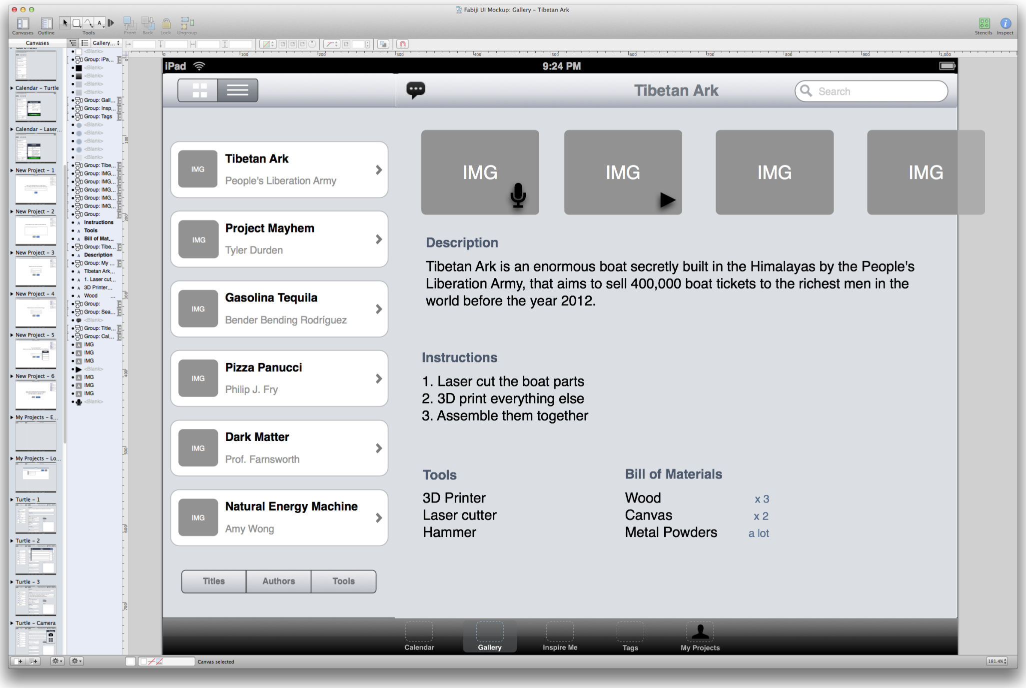

Figure 4.4: The new Gallery page with a segmented control button.

on the one they want, but also increase the complexity of the application. As we may have many users at the same time, if we use a dedicated tab for each user, there will be too many tabs for the screen to hold.

- Change the tab *All* to *Gallery*, and change the tab *Today* to *Calendar*: Some users were confused when asked to carry out Task 1 (start a new scheduled project), they do not know whether to start from *Today* or *All*, because the scheduled project is for today, and it also belongs to All projects. By changing the tab *All* to *Gallery*, we implicitly tell users that the contents in tab are not editable. And by changing the tab *Today* to *Calendar*, we tell the users this tab is only for displaying calendar, not a start point of today's work.

- No login required: Four out of five users suggest that login is not necessary because "it is complicated if I have to register with email" and login also requires more workload to remember and type password.

- Remove Photo Story: We introduced Photo Story when designing the paper prototype (see Section 4.1.2) where each photo is accompanied by a short piece of audio recording. This proved to be not a good idea by test users, because people generally do not want to record their voices or hear their voices when played back afterwards. Voice recording works the best in two-direction communications such as phone calls, voice messages and interviews [Harper et al., 2005]. Users also showed little interests in recording the ambient sound at the fab lab.

4.3 Second Software Prototype

We created a functional prototype for iPad.

The second software prototype is a functional iPad application written in Objective-C with Xcode IDE. In this section we explain the design and implementation of the prototype, and in Chapter 6 we describe the evaluation of the prototype together with the photo kiosk with fab lab users.

4.3.1 Design

We develop the code using the Model-View-Controller (MVC) design paradigm, where the UI (view), view controller (controller) and data (model) are separated. In this section we introduce our view controller hierarchy and how we handle our data.

View Controller Hierarchy

Basic introduction to `UIViewController`.

iOS provides a fundamental view-management class called `UIViewController`, every view belongs to a view controller and each view controller is responsible for managing its views (e.g. creating, changing, removing) and handle user interaction on its views. Programmers usually do not manipulate the views directly but through view controllers. iOS also provides a set of subclasses of `UIViewController`,

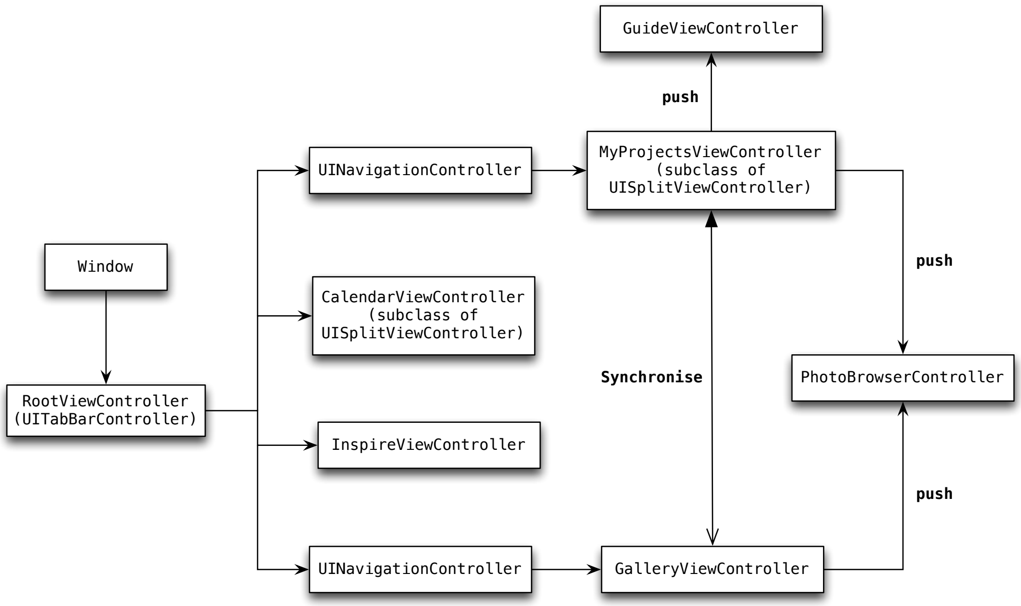

Figure 4.5: The Fabiji app's main view controller hierarchy.

such as `UINavigationController`, `UITableViewController`, `UISplitViewController` and `UITabBarController`. Each of them has its own unique views and methods, and they can be inherited or extended to realise complex behaviours.

View controller hierarchy of the iPad application.

Figure 4.5 shows a simplified view controller hierarchy listing only main view controllers in our application. The `RootViewController` of the window is a `UITabBarController`, the `UITabBarController` has four child view controllers, and each of them represents a tab in the user interface. While `CalendarViewController` and `InspireViewController` are two static view controllers, `MyProjectsViewController` and `GalleryViewController` will push another view controller at some point, and this push event needs to be done via `UINavigationController`. This is the reason why we need to put these two view controllers inside `UINavigationController` and together as children of the `UITabBarController`.

As one may notice, we removed the *Tags* tab in this prototype, because the tags can be implicitly incorporated into the search bar in the *Gallery* view. Whenever a user types a keyword in the search bar, he is searching project title, project author, and project tags. In this way, we successfully decrease the number of tabs by one.

Since MVC encapsulates each view controller, we cannot directly access other view controllers from the current view controller, and it is generally not a good practice to have view controller instances as global variables. iOS provides several useful properties that allow us to access other view controllers, such as `parentViewController`, `childViewController` and `navigationViewController`. For example, if we need to access `GalleryViewController` from `InspireViewController`, we first get `UITabBarController` by finding the `parentViewController` of the `InspireViewController` instance, then the `GalleryViewController` is the child of the fourth tab of the `UITabBarController`.

We access other view controller using their relationships.

Each time a photo or video is added or removed to the project documentation in `MyProjectsViewController`, we need to notify `GalleryViewController` to refresh the thumbnails. This is done by implementing a delegation method in the `GalleryViewController` and set it as the delegate of `MyProjectsViewController`. So that each time a photo or video is added in `MyProjectsViewController`, the delegation method in `GalleryViewController` will get evoked to add a new thumbnail.

View controllers notify each other with delegation methods.

Data Structure and Data Storage

There are three main classes in our application: `FABUser`, `FABProject`, and `FABAppointment`, Figure 4.6 shows the members of these classes and the relationships between these classes. Note that the relationship between `FABUser` and `FABProject` is one-to-many, meaning that a user can have many projects but each project belongs to only one user.

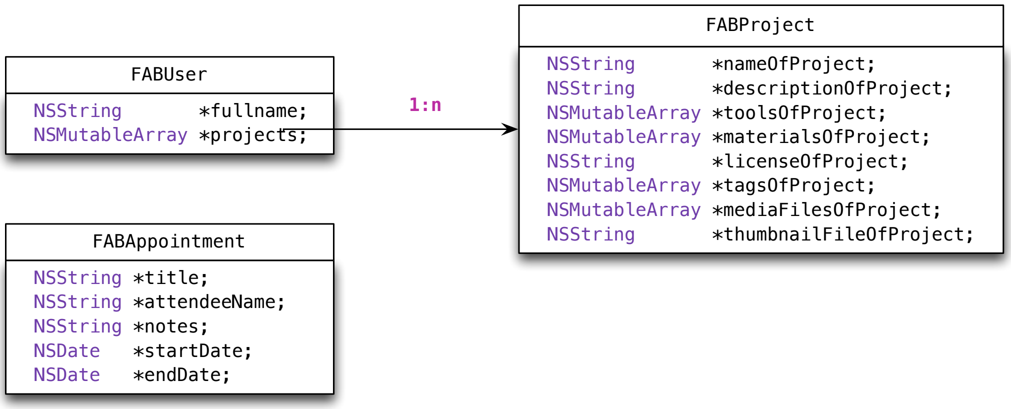

Figure 4.6: Main data structures and their relationships.

As for storing the data, we compare the following three popular solutions that are in iOS applications,

- **Property list** is a simple nested key-value pairs in XML files to store basic data types such as strings, numbers, arrays and dictionaries. It is easy to understand and can be directly read or written using any text editor. The drawback is that it is very difficult to perform complex queries on property list, and the complete file has to be loaded to a dictionary in order to get any data out or store any data in.
- **Core Data** work as a database inside the application, and do not require loading the whole database like in property list, and thus it is suitable for applications with large and complex data structures. But it has relatively steeper learning curve and requires more code.
- **WebDAV** is a protocol to transfer files over HTTP. It allows users to create, modify and move documents on web servers. Applications such as Keynote for iOS[6] use WebDAV for storing large files on user-specified servers.
- **iCloud** is a cloud storage service introduced in iOS 5. It is a infrastructure that allows applications to synchronise its documents and data across the user's all

[6]http://itunes.apple.com/app/keynote/id361285480?mt=8

iOS devices and Mac computers running Mac OS X 10.7 and above.

We decide to use `property list` files to store our data because our data structure is easy and no complex quires are needed. Nevertheless, in order to allow users to edit their documentation from anywhere, the data must be stored on a server. So we define a set of RESTful API [Fielding and Taylor, 2002] to communicate with the server (shown in Table 4.1[7]). In order to avoid name conflicts, each time our application creates a new binary file, which can be an image, video, or audio, we generate a new UUID (Universally Unique Identifier) as its file name.

We define a set of RESTful API to communicate with the server.

Method	URL	Description
`GET`	`users`	Get a list of all User Objects
`POST`	`users`	Create a new User Object
`GET`	`users/id`	Get the User Object with ID id
`PUT`	`users/id`	Change the user Object with ID id
`DELETE`	`users/id`	Delete user Object with ID id
`GET`	`users/id/projects`	Get a list of Projects of the user with ID id
`GET`	`users/idu/projects/idp`	Get the Project with ID idp
`PUT`	`users/idu/projects/idp`	Add the user with ID idu to project with UD idp

Table 4.1: Part of RESTful API list

4.3.2 Implementation

We develop the iPad application with Xcode 4.2 using iOS 5 SDK. The final user interface screenshots can be found at Appendix C.

Multimedia

In order to display audio recordings, photos, and videos using a single media browser, we create our own media browser based on an open source photo browser KTPhotoBrowser[8] . Although the original

[7] See full list of API at http://hci.rwth-aachen.de/fabijiapi
[8] https://github.com/kirbyt/KTPhotoBrowser

We create a media browser that displays audio recordings, photos and videos in a single browser.

KTPhotoBrowser can only display photos, it provides several useful data source methods (such as `imageAtIndex`, `thumbImageAtIndex`, `deleteImageAtIndex`, `exportImageAtIndex`) for other controllers to implement. We extend the `imageAtIndex` method to `mediaAtIndex` by using `MPMoviePlayerController` from `MediaPlayer` framework to playback audio and video.

We use Quartz 2D to resize images properly.

We use `UIImageView` to display thumbnail images of projects. `UIImageView` offers automatic resizing function whenever the size of the image inside its view is larger than the size of its bounds. But this automatic resizing results in jagged edges since it simply squeezes the image and display it in a smaller frame (see Figure 4.7). In order to have smoothly resized images as thumbnails, we use `CGContextSetInterpolationQuality()` method in Quartz 2D (A 2D drawing engine with API in C language) with `kCGInterpolationHigh` parameter. Since the camera resolution of iPad 2 is 960x720 pixels, this method is fast enough to handle the photos that users take.

Figure 4.7: Comparison of resizing image of 1024×680 pixels to 300×200 pixels: Automatic resizing from `UIImageView` (left); Resizing with `CGContextSetInterpolationQuality` method using `kCGInterpolationHigh` parameter in Quartz 2D (right).

Thumbnails for videos are created as well.

Not only do we need thumbnail images for photos, we also need to display thumbnails if users record videos. Since a video is essentially a sequences of images, we simply find the image of the first frame of the video with `copyCGImageAtTime:actualTime:error:` method of `AVAssetImageGenerator` class, and then resize that image using the method we mentioned above.

Calendar

Fab Lab Aachen currently uses Google Calendar[9] to manage online appointments, in order to extract events from Google Calendar to our application, first we need to set up the calendar on the iPad to sync with the Fab Lab Aachen Google Calendar. This is done by adding the google account of Microsoft Exchange type, and login to google mobile website[10] to choose what calendars to synchronise. After this step, the Calendar app in the iPad should be able to display (and in sync with) Fab Lab Aachen calendar properly. Then in our application we use `EventKit` framework to retrieve events from the Calendar app.

We use `EventKit` to retrieve calendar events.

[9]https://www.google.com/calendar/
[10]http://m.google.com

Chapter 5

Photo Kiosk

In order to assist users to take better photos, we created a photo kiosk for iPad. It offers a great viewpoint for the iPad camera as well as an LED lit background. Since all parts of the photo kiosk are laser cut, it is easy for other fab labs to build its own copy. The photo kiosk together with the iPad also serves as the complete Fabiji system.

5.1 First Prototype

Since users should be able to put their physical objects in the kiosk, and take a picture without moving the iPad (given the size of the object does not exceed the inner volume of the photo kiosk), we have to make sure that the view volume of the iPad camera covers all angles of the bottom area of the kiosk.

We made our first prototype to test the view volume of the iPad camera (see Figure 5.1). The first prototype is laser cut from 2mm MDF. Since in the landscape mode, the camera on the iPad is at either the upper-left or the bottom-right corner, if one holds the iPad in a way that the camera is in the upper-left corner, only views ahead of it will be covered by the camera but not views under the iPad. In order to cover the views under the iPad, we fix the iPad in the landscape mode and the camera on its bottom-right corner.

The first prototype is made to test the view volume of the iPad camera.

The top panel in Figure 5.1 shows a hole, which is reserved for the iPad camera to see through. The bottom panel is a rectangle that shows the area that the iPad camera sees. The cross target in the middle of the bottom panel is the centre of the camera viewing volume. In order to take pictures in a perspective view instead of top view, we made the top panel tilting at $30\,^{\circ}$.

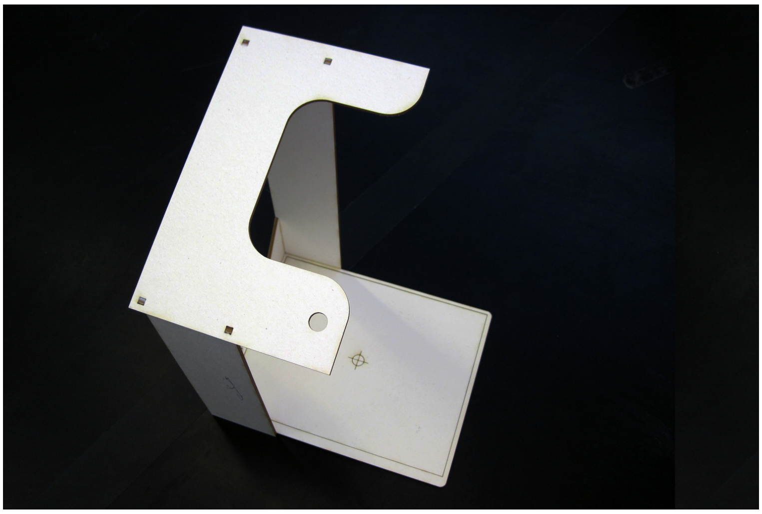

Figure 5.1: First prototype to test iPad camera view volume.

5.2 Second Prototype

Now that the first prototype proved that the camera view volume is the same as we expected, we make our second prototype to really embed an iPad in.

5.2.1 3D Design

We design the 3D model of the photo kiosk in Rhinoceros 3D[1] (see Figure 5.2). Based on the first prototype, we increase the size of the bottom panel to $30cm \times 20cm$ (roughly A4 paper size), and attach a door to it, so that users can

[1] http://www.rhino3d.com

close the door and make use of the LED light inside the kiosk. As one may have noticed, because of the camera position, the iPad is not in the centre of the top panel. In order to support the iPad weight, we have to extend the bottom panel further to the left, and we make a closed compartment for cables and power socket off the extended areas.

The front part of the photo kiosk is a half cylinder made of four pieces. We design it this way so that there won't be sharp edges preventing users' hands from leaning forward. In addition, users can easily rest their hands on the front panel when typing on the iPad.

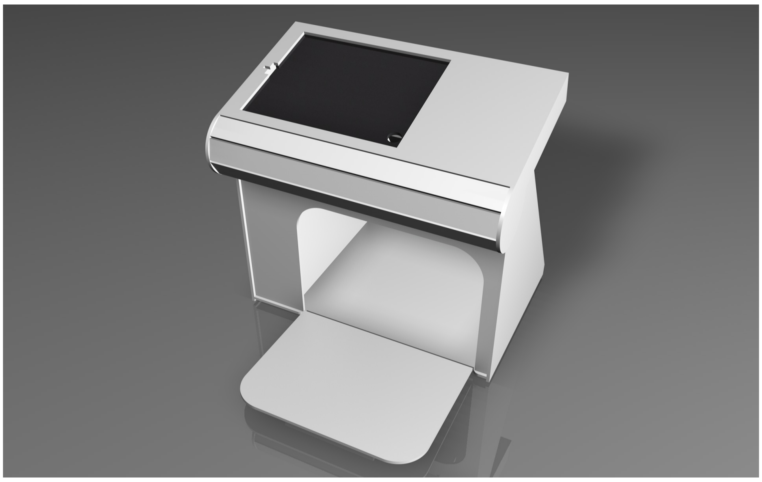

Figure 5.2: 3D rendering of the photo kiosk in Rhinoceros 3D.

5.2.2 Materials and Assembly

According to the 3D design, we create the 2D vector cutting pieces (download link at Appendix C). Altogether there are 29 pieces, and we manage to put them in eight $30cm \times 60cm$ $5mm$ MDF boards. We cut the pieces using Epilog Zing laser cutter[2] with the parameters listed in Table 5.1. During experiments, we found that cutting the bolt holes (see the first picture inFigure 5.4) three times using raster parameters provided in this table produce the best result. Here we

We laser cut eight pieces of MDF to make the structure of the photo kiosk.

[2]http://hci.rwth-aachen.de/lasercutter

use foam rubber as a protection layer against MDF for the iPad.

Material	Raster Speed	Raster Power	Vector Speed	Vector Power	Vector Frequency
MDF ($5mm$)	100	45	26	100	500
Foam rubber ($2mm$)	-	-	30	5	5000

Table 5.1: Laser cut profiles for the photo kiosk

We use socket flat head screws so that the surface of the kiosk is smooth.

Instead of socket head cap screws commonly found at fab labs, we use socket flat head screws (see Figure 5.3). While head cap screws always keep the heads outside, flat head screws can have their heads embedded into the material. We designed a connection (see Figure 5.4) where a hole of the screw head size is drilled, and the screw head is nicely embedded into MDF, making the surface smooth. We used 89 pairs of screws (M3x10 DIN7991) and nuts (M3 DIN562) together with this type of connections in the second prototype[3].

Figure 5.3: Comparison of screws with different heads: Socket head cap screws (Left); Socket flat head screws (Right). Images from drillspot.com.

After finishing the assembly all laser cut pieces (see Figure 5.5), we attach a door to it with hinges at the bottom and magnets at the top, so that users can open and close the door easily. We then light up the inside of the photo kiosk with several stripes of 12V white colour LEDs.

[3] An assembly video can be found at http://vimeo.com/38198400

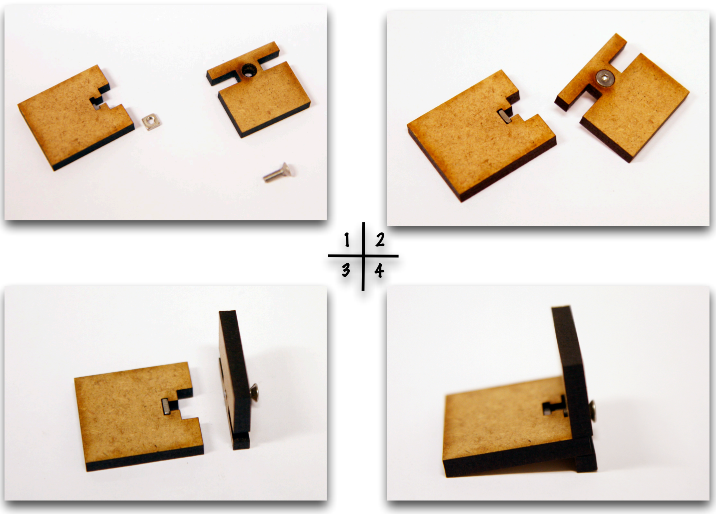

Figure 5.4: Connection using socket flat head screws.

Figure 5.5: Photo kiosk right after assembly (without the door and the foam rubber).

Chapter 6

Evaluation

We evaluate our system in three ways, first we validate the user experience design of the iPad application with *iOS Human Interface Guidelines*, then we describe and analyse the results of the user study that we conducted on fab lab users, finally we explain whether or not we have met the system requirements raised in Section 3.3.2, and find if we .

6.1 Guidelines

As its name suggests, *iOS Human Interface Guidelines* provide a set of guidelines that help programmers to create a better user interface. It is not a rulebook, but the guidelines reflect good practices in iOS application design. Here we validate the user experience design of our iPad application with guidelines from the *User Experience Guidelines* chapter.

We validate our iPad application with six guidelines from *iOS Human Interface Guidelines*.

- **Focus on the primary task**: The primary task of an iOS application is defined in *application definition statement*, which in our application, is "a documentation creation and exploration tool for fab lab users". In our application there are four tabs in total: Calendar, Gallery, Inspire Me and My Projects. My Projects tab is dedicated to create project, Gallery and Inspire Me tab are exclusively for users to explore existing tabs.

In the My Projects tab, the application is focused on documentation fields, users can edit the text fields and the image view directly by tapping on them. In the Gallery tab, only project thumbnail images and a search bar are in display, so it clearly indicates that users can only tap on the thumbnail images to view the project gallery. Similarly, in the Inspire Me tab, a list of statements are shown in a table view, users can simply select the statement that describes the projects they are interested in with one tap.

- **Think top down**: Except for the tab bars which must be fixed at the bottom by the iOS SDK, all tappable buttons are in the top bar of our application. This makes controls more visible and easy to tap on.

- **Use user-centric terminology**: We use text in our application according to its context to avoid misunderstanding. For example, in the Calendar tab, we use user-friendly term "today" instead of displaying the actual date. Since Fab Lab Aachen Open Days are once a week, there are only appointments on one certain day per week (currently Tuesdays). By clicking "last" or "next" button, users navigate appointments to last or next *week*, instead of last or next *day*.

- **Minimise the effort required for user input**: Since our application requires extensive user input, it is rather important to minimise user input. Compared to typing on iPad, tapping requires much less workload, so we replace several text fields with picker where users can tap on pre-defined choices. For example, since most users will use existing tools at the fab lab, we programmed a list of tools into the application. When a user wants to add a tool, he simply taps on the tool text field and a list of tools will pop up for selection.

- **Use subtle animation to communicate**: Animation in iOS application is mainly used to report current status with users, and provide visual feedback so that users know something has changed. Since our application is a productivity application, it does not need fancy animations but simple ones to communicate current status. Throughout our application, only two types of animations are used: the standard transition animation when a view is pushed or popped, and the `UIViewAnimationTransitionFlipFromLeft` animation to show the step-by-step guide when users create new projects. The first animation is to notify the user that the current view is changed, the second animation is to change the current view complete (hence *flip* the screen).

- **Ask people to save only when necessary**: There is no save button of any kind in our application, instead, all edits such as changing project description and changing project thumbnails are saved automatically. Only delete and cancel buttons are explicitly displayed.

6.2 User Study

In order to understand how fab lab users think of Fabiji, we ran a user study on fab lab open days. In this section we first propose four hypotheses, and then explain the procedure of our user test, finally analyse results from the user study to support these hypotheses.

6.2.1 Hypothesis

Corresponding to the requirements R1-R4 raised in Section 3.3.2, we proposed the following four hypotheses. Requirement R5-R6 will be validated later in Section 6.3.

We come up with four hypotheses based on requirements raised earlier.

- **H1** Users find that Fabiji non-intrusively helps them to create simple documentation in a short time.
- **H2** Users find that Fabiji encourages them to create documentation at the fab lab.
- **H3** Users find that Fabiji encourages them to explore other's projects when they are at the fab lab.
- **H4** Users find that Fabiji gives them opportunities to meet in real life.

6.2.2 Experimental Setup

To support the above mentioned hypotheses, we need to obtain qualitative feedback from questionnaire results. Here we explain how we designed the user study.

Participants

Participants are a mix of visitors on the Fab Lab Open Day and members of Media Computing Group.

Similar to the initial study (Chapter 3), we recruit only fab lab users as participants. Although there can be eight users coming to a fab lab open day, in practice there are three to five users who actually come. Since our eligible users are scarce, we ask all eligible users to be our participants. We did not require previous fab lab experience because our target users include both experienced and inexperienced users. In order to avoid all participants to be inexperienced fab lab users, we also have three members of Media Computing Group who work at the fab lab quite often as our participants.

Methodology

We first introduce participants the functions of Fabiji.

At the beginning of the study, we asked all participants to fill out consent forms. Then we gave participants a short introduction of the Fabiji iPad application and the photo kiosk. We showed the user interface and features of the iPad application, in particular how to create new project documentation from the scheduled appointment and how to view other people's projects.

No specific task was assigned in the study.

Since the object of the study is not to test specific features of the iPad application or the photo kiosk, we wanted to test the whole user experience of creating physical objects and create documentation with Fabiji at the fab lab, we did not assign specific tasks to our participants. Instead, we asked the participants to first create a project documentation with the step-by-step guide, and then work with the fab lab machines as they had planned, after they finish making the physical objects or whenever they have time, they can come back to continue editing the project documentation.

During the study, we used the think aloud technique by chatting with the participants and letting them to talk about their opinions. When questions about finding a specific function on the iPad were asked (e.g. Where to start a new project), we first asked the participant where he would like the functions to be. If he still couldn't find it, we would give hints to the user.

We used think aloud technique during the study.

Questionnaires

After participants finished their work and the project documentation, we asked them to finish the System Usability Scale (SUS) questionnaire [Brooke, 1996].

The post session questionnaire consists of five questions and ten statements of 5-point Likert scale (see Appendix A for full questions). The first three questions Q1-Q3 are common demographics questions asking participants' age, gender and occupation. Q4 and Q5 are regarding to user's fab lab experience and iPad experience. The ten statements of 5-point Likert scale ask participants' opinions about how they think of the system. Statement S1-S2 ask participants if they think fewer tools and less time are needed in creation project documentation, Statement S3 asks if the user thinks that it is a burden to create documentation at the fab lab, and Statement S4 asks if the step-by-step guide is helpful in creating documentation. Statement S1-S4 altogether are used to testify hypothesis H1. Statement S5-S6 ask if users feel like to create more documentation at the fab lab with the help of Fabiji, and these two statements are to testify hypothesis H2. Statement S7-S8 ask if users think they can see more project documentations by others and if their own projects can receive more audience with Fabiji, Statement S9 asks the accessibility to other people's projects. Statement S7-S9 are asked to testify H3. The last statement S10 asks if participants would like to meet more people and their projects at the fab lab, which is used to testify H4.

The post session questionnaire directly relates to the hypotheses.

SUS was originally proposed as a "quick and dirty" usability scale but proved to produce very reliable results [Bangor

We also asked users to fill out the SUS questionnaire.

et al., 2008]. It consists of ten statements of 5-point Likert scale ranging from "strongly agree" to "strong disagree". Each rating of the statement has a score between 0 and 4, and the final score of SUS is the sum of all rating scores of the ten statements multiplies a factor of 2.5. As the maximum score of each statement is 4, the maximum score of SUS is $4 \times 10 \times 2.5 = 100$ while the minimum score is 0.

We evaluated usability and learnability factors of SUS as well.

A factor analysis of the SUS was done by Lewis and Sauro [2009], the results of which show that the SUS has two factors: Usability and Learnability. While Usability is aligned with Statement 1-3 and 5-9, Learnability is based on Statement 4 and 10. Although Lewis and Sauro also argue that these two factors are reliable, such findings are not confirmed with Bangor et al. [2008]. As a result, Usability and Learnability should be carefully used. The post session questionnaire and SUS were finished immediately after participants finished the study.

6.2.3 Results

Participants

Eight participants were tested in our study. The participants are aged 23-27 (M = 25.4, SD = 1.6), and among them there are four females. All of them are students of RWTH Aachen University, and their fields of studies include architecture, bioinformatics, computer science and electrical engineering.

Demographics of the participants.

As shown in Figure 6.1, among the participants, 50% are the first time user of the fab lab, meaning they heard of fab labs but don't know how fab labs work and probably never used the machines in the fab lab. They need explanation and technical assistants in using the machines. 13% of participants are frequent users, who work at the fab lab for several times before, they probably know how to use the machines at the fab lab and need little technical assistants for using them. 38% of participants are regular users, who work at the fab lab regularly, and know how to most machines at the fab lab and do not need technical assistants

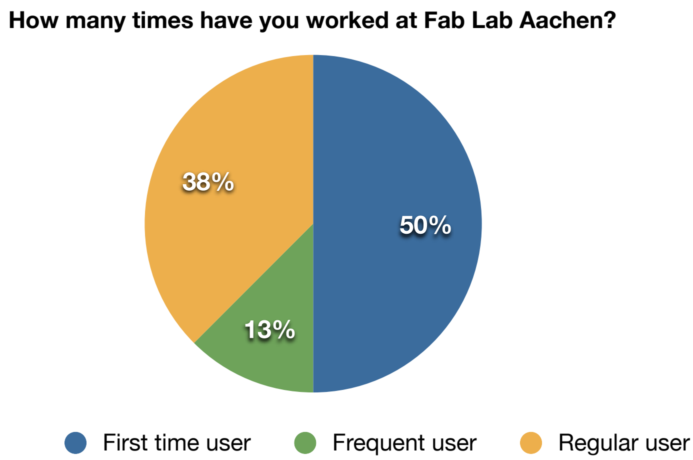

Figure 6.1: Participants' experience of fab lab.

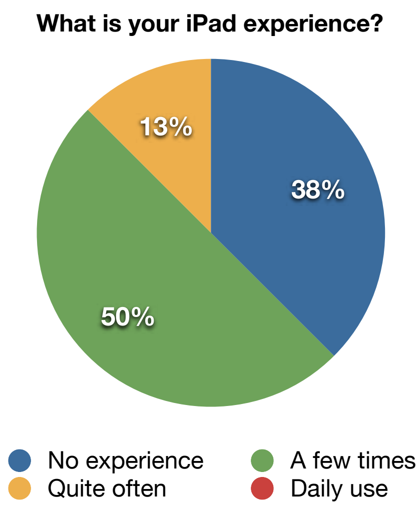

Figure 6.2: Participants' experience of iPad.

for using them. Note that although regular users of the fab lab are participants are students from the Media Computing Group, they have never used Fabiji before and yet they have intense experience with the fab lab, so they can eligible participants and thus can be regarded as experienced fab lab users.

Most of the participants don't use iPad quite often.

Most of the participants don't use iPad quite often (see Figure 6.2), with only 13% participants answered "I use it quite often" when asking their iPad experience, and no participant is daily user of iPad. While 38% participants have no experience using iPad, 50% participants use iPad a few times before.

Post Session Questionnaire

The results of the post session questionnaire are analysed using boxplot (see Figure 6.3). We use boxplot because we want to demonstrate the central and the distribution of users' opinions.

Most participants find that fewer tools are needed to create documentation with Fabiji.

Regarding Fabiji's helpfulness in creating documentation at the fab lab, most users strongly agree that fewer tools are needed to create a project documentation using Fabiji (S1). Indeed, since the iPad is equipped with most of the features needed to finish basic hardware project documentation, users do not need to their own pencils and cameras.

Participants strongly agree that the step-by-step guide helps creating documentation.

Nearly all users, both inexperience and experienced users of the fab lab, strongly agree that they find the step-by-step guide is helpful to create documentation (S4). This is not surprising because for inexperienced users, they need such as guide as the first visit of fab lab is already information overwhelming, they do not want to spend more time learning how to use an application, and such a guide serves as a virtual assistant helping them to go through steps in documenting their projects. For experienced users, it is also beneficial to have a guide so that they don't forget any important things to be added into the documentation.

Participants agree that by using Fabiji, less time is needed to create project documentation (S2). The statement is not

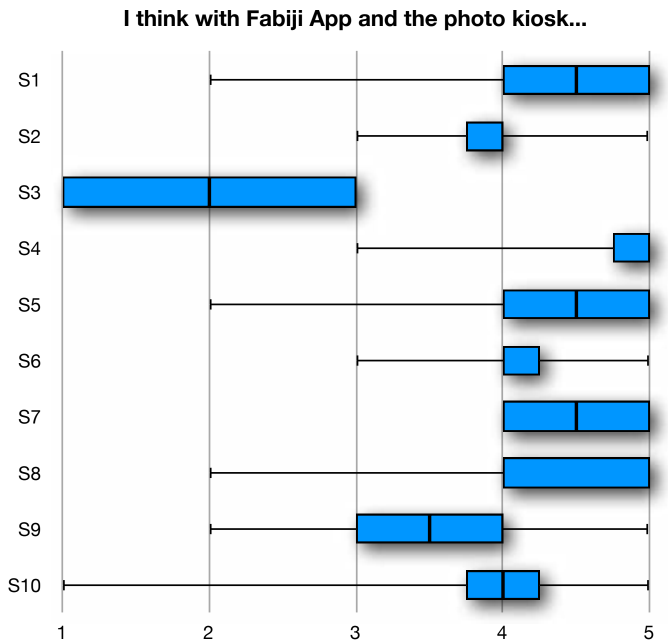

Figure 6.3: Boxplot of results of ten Likert statements in the post session questionnaire.

strongly agreed by participants because there is a learning curve for uses to adapt to iPad. Especially in our study, most participants do not use iPad very often. Nevertheless, users can finish basic project documentation at the fab lab with fragments of time such as waiting machines to work, they don't need to spend extra time at home for the documentation. We think this is the primary reason why users think less time is needed to create the documentation.

Participants agree that less time is needed to create documentation with Fabiji.

Participants generally do not agree that it is a burden to create documentation at the fab lab (S3). Some users chose the answer "Neutral" to this statement because they do not have the habit of creating documentation or they don't want to share their documentation to the public, but in the user study they were asked to do so. For example, two of the participants were architecture students, and they came to the fab lab to laser cut architecture models for their thesis projects. They do not want to write any detailed things about their projects but rather keep them in their theses.

Participants do no find it is a burden to create documentation at the fab lab.

S1-S4 confirm H1.

Statement S1-S4 together prove that users find Fabiji make them create project documentation with fewer tools and less time, and users generally find Fabiji helpful and it is not a burden. Thus, these facts confirm Hypothesis H1, and they also prove the idea of using a photo kiosk with embedded iPad as documentation system, which is raised in Section 3.3.3, is a viable and user preferred solution.

S5-S6 confirm H2.

Many participants strongly agree that Fabiji is an integral part of the fab lab (S5). We think the reason is that many users already create project documentation before the test, and they enjoy it when Fabiji helps them to create documentation more efficiently. This partly supports Hypothesis H2, because if users think Fabiji is an integral part of fab lab experience, they will create documentation at the fab lab if they can. Most participants agree that by using Fabiji, they will create more project documentation (S6), which clearly confirms Hypothesis H2.

All participants either agree or strongly agree that they can see more projects created at the fab lab by other people (S7). This is quite clear since there is some example project documentation already in the Fabiji application before they start test. We deliberately prepared these examples in advance because when Fabiji is deployed in the fab lab, project documentation will accumulate as time goes.

Statement S8 is regarding the audience of the Fabiji users' projects, and most participants strongly agree that their projects can be viewed by more people with Fabiji. We think this is true because before there are two main places where people can see others' projects: at the fab lab while the creator is present, or at the less known online Picasa web gallery where there are not much detailed descriptions but pictures. Fabiji saves their documentation, and whoever comes to the fab lab can see their projects.

Result from Statement S9 does not prove that participants find it is easy for them to find documentation they are interested in. We think this is due to incompleteness of the prototype during the test. The prototype we presented during the test does not have the Inspire Me page implemented functionally, but only have the user interface implemented. That is, whichever option that users select (e.g. projects that are popular, or projects that are created for gifts), the application will redirect the current view to the same Gallery page. We did not implement the functions in Inspire Me page because it needs potentially large amount of tags generated by users, and based on those tags we could either create a finite state machine or a machine learning algorithm to match the projects with the user selected options. This gulf of execution results in users opinion on accessibly to the projects they are interested in. Although the function is not fully implemented, users did confirm that they liked Inspire Me page, and feel it would be helpful if implemented.

Statement S7-S9 together prove that Fabiji users can find more projects and Fabiji helps project documentation created at the fab lab to have broader viewers. Thus, Hypothesis H3 is confirmed.

S7-S9 confirm H3.

Most participants agree that they would like to meet more people and/or see their projects at the fab lab (S10). There are participants who choose disagree or strongly disagree because they are student assistants or thesis student working at the fab lab, so they do not want to work longer. For participants who wanted to meet more people at the fab lab, we could say that the chance for them to meet in real life is increased because Fabiji provided a platform. Therefore, we could confirm Hypothesis H4.

S10 confirms H4.

SUS

The SUS score for Fabiji was $77.9 (SD = 13.5)$, and the system is rated as *Good* from 72.75 to 85.58 points according to Bangor et al [2008]. Although the SUS score does not guarantee good usability, but together with the post session questionnaire they prove that Fabiji has good usability. We also calculated the two factors from SUS proposed by Lewis and Sauro [2009], usability and learnability. Fabiji has a usability score of $76.8 (SD = 14.3)$ and Learnability $82.0 (SD = 21.5)$. Note that we have a relatively higher learnability score, which indicates that although most participants were not frequent iPad users, they can still use our iPad application without too much assistance.

SUS score indicates that Fabiji is a good system.

6.3 Requirement Analysis and Comparison

We have raised six requirements (R1-R4) in Section 3.3.2. As in the beginning of last section we have proposed four hypotheses H1-H4 corresponding to the requirement R1-R4, and in the post session questionnaire we confirmed all four hypotheses. This means that requirement R1-R4 are met. Here we analyse our system to show that R5 and R6 are also met.

- **R5: Help users to take better photographs**: We used iPad 2 (the latest iPad by the time of development) in Fabiji, and the resolution of its camera is only 960×720, this resolution generally does not produce great quality photographs. But with the white background, the LED lightening, and the good shooting angle, we managed to maximise the quality of images taken from the iPad camera. A comparison of photographs taken with and without the photo kiosk from the iPad camera can be seen at Section C.2. A clear improvement can be found there.
- **R6: Easy to deploy and configure for other fab labs**: The Fabiji system consists of two parts, the iPad application and the photo kiosk. With the help of Apple App Store, it is really easy to install the iPad application within a few taps. For the most important parts of the photo kiosk, we use standard DIN materials, and the laser cut pieces (see Appendix D) are open source and can be cut from any laser cutters that has a cutting size of (or greater than) $30cm \times 60cm$. Other non-standard materials such as LED stripes or magnets, do not require the exact same materials as we have, but common ones found in any electronic markets can do. As a result, both the software and the hardware can be deployed and configured in other fab labs with ease.

In Chapter 2, we created a table (Table 2.2) to compare main properties of the existing documentation systems which are discussed in this thesis, now we add Fabiji into that comparison table and thus form a new comparison table (Table 6.1). We can see that Fabiji is a medium sized documentation system dedicated to fab lab users, it is mobile, easy to install, and it can be used by multiple users at a time (hence shared).

No.	Project	Mobile	Easy to install	Size	Ownership	Target Users
1	SHARE	✓	✗	Medium	Personal	Engineers
2	PENS	✓	✗	Medium	Personal	Engineers
3	EEN	✓	✗	Small	Personal	Engineers
4	Ars Electronica Fab Lab	✗	✗	Large	Shared	Fab lab users
5	FabML	-	✓	-	-	Fab lab users
6	ThingDoc	✓	✓	Small	Personal	Engineers
7	Timeline Tools	✓	✗	Medium	Personal	Designers
8	HandMade	✓	✓	Small	Personal	Craftsmen
9	Fabiji	✓	✓	Medium	Shared	Fab lab users

Table 6.1: Comparison of existing documentation systems and Fabiji.

Chapter 7

Summary and Future Work

7.1 Summary and Contributions

In this thesis, we presented Fabiji, a system that non-intrusively helps users to document physical objects at the fab lab. Fabiji consists of two parts, a photo kiosk for assisting users to take quality photographs, and an iPad application that helps users to create and explore hardware project documentation. We made both the software and the hardware of the Fabiji system easy to deploy and configure, allowing other fab labs to reproduce copies. While most documentation systems are intended for personal uses, Fabiji can be used by multiple users at a time, hence it is a shared platform.

We created an easy to install and configure documentation system for fab labs.

In the initial study that we conducted on fab lab users, we identified four types of typical users groups: hackers, PhD students or researchers, engineers, and architecture students. Each user group has its own goal before coming to the fab lab, and thus each group has its own motivation of creating and sharing project documentation. Hackers create mostly open source projects, and they most definitely will share their project documentation in hope of other people like their projects and improve them. PhD or researchers mainly create projects at their own research

In the initial study, we identified who the fab lab users are, and their motivation of creating and sharing project documentation.

needs. As scientific practice, they will create complete and detailed project documentation, and share it with their colleagues. However, before the corresponding research papers are published, they will not share the project documentation with others. Engineers work at the fab lab to make the most of its machines and tools. Because their projects may bring profits some day, they are reluctant to share the details to other people, but they would like other people to see their projects. Since architecture students mostly come to fab lab to laser cut their models for school projects, they are also reluctant to share their design, but they are willing to share text and images.

Final evaluation shows that all requirements are fulfilled.

The software prototype of the iPad application focuses on two features: assisting users to create a basic hardware project documentation in a short time, and encourage users to explore other people's projects. Our user study showed that users found step-by-step project creation guide helpful, and meanwhile they didn't fill it was a burden using Fabiji. The user study also confirms that Fabiji encourage users to create and sharing documentation.

7.2 Future Work

7.2.1 The Fab Lab Experience

Together with a back-end server and front-end web community, Fabiji will bring fab lab users a whole new experience.

Currently, users need to schedule appointments online using third party schedule services, for example, Fab Lab Aachen uses Tungle.me[1] , a free meeting scheduling service that supports many platforms. The problem is that these services do not support customised fields that can be used later in the documentation system. For example, the customised fields can be a list of machines, time slots and fab lab experience. If this information is provided, Fabiji can incorporate it into one's documentation and thus he does not need to write it again. Moreover, if there is a dedicated scheduling service for the fab lab, one can add user login, as well as a database to store all user information and project documentation, and eventually it becomes a community

[1]http://www.tungle.me/

website for fab lab users. By then, Fabiji can communicate with the server of that community, and retrieve and store information directly to the server.

With the support of such community, users can store their design files online under their accounts before they come to the fab lab. When they arrive at the lab, they can use Fabiji to get their files to either put them into documentation or send the printing job (given they are laser cutter supported files) directly to the laser cutter. With user account in such community, documentation sharing can be even easier. It is possible to let users publish their project documentation to the server of the community website from Fabiji with one single tap.

With the combination of Fabiji, back-end server of and front-end website of the fab lab community, a whole new user experience is brought to working at fab labs.

7.2.2 The Fab Documentation Format

As Määttä and Troxler [2011] suggested, a universal file format that can easily share between fab labs are needed. The first step is to identify which elements are needed to describe hardware project documentation, and what source file types it should support. We have summarised a few fields that we think it's essential to be included in such documentation, but further discussion should be made to come up with a standard format with which every fab lab complies.

Design a general fab documentation format that every fab lab uses.

7.2.3 Further Experiments

The user study we carried out in our final evaluation has only eight participants, and they were introduced to Fabiji in the beginning of the study and observed throughout the study. In the future, a test bed can be carried out for a longer period (e.g. six months), without observing user behaviour but simply let users explore Fabiji and in then

Run a user study for longer time with no introduction to the Fabiji system.

end count the number of project documentation generated. This produces the best results on users acceptance to Fabiji.

Appendix A

User Study Materials

A.1 Questionnaire for the Initial Study

This questionnaire contains 18 questions.

A.2 Post Session Questionnaire for the Final Evaluation

This questionnaire contains five questions and ten statements of 5-point Likert scale.

Fabiji - Initial Study Questionnaire

Age: ________ **Sex**: ☐ M ☐ F **Occupation**: ____________

Part I: User Background

Q1: How many times have you been to Fab Lab Aachen (including today)?

Q2: What are you going to make today?

Q3: For how long are you going to work on this project at Fab Lab Aachen?
☐ 1h ☐ 2h ☐ 3h Please specify ____________

Q4: Why do you build this project?

Part II: Fab Lab Aachen and Personal Fabrication

Q5: Where did you hear about Fab Lab Aachen?

Q6: Have you heard of the following professional fabrication services?
☐ Shapeways ☐ Ponoko ☐ BatchPCB ☐ Other ____________

Q7: Why do you use the fab lab instead of professional fabrication services?
☐ Because fab lab is cheaper ☐ To meet creative people
☐ To learn to use machines ☐ Other ________________

Q8: If you have to either pay or share documentation to use the fab lab, What would you prefer?
☐ Pay ☐ Share documentation ☐ Choose one myself ☐ Won't come

Part III: Documentation and Sharing

Q9: Have you heard of the following terms?
☐ Open source hardware ☐ Creative Commons ☐ Make Magazine
☐ Thingiverse ☐ Instructables

Q10: Are you used to document in your work or study?
☐ Yes ☐ No

Q11: Will you document your project?
☐ Yes ☐ No

Q12: In what form do you prefer to document your projects?
☐ Text ☐ Picture ☐ Video ☐ Voice recording ☐ Other ___________

1

Figure A.1: Questionnaire for the initial study, page 1 of 3

Q13: Will you share the documentation with other people?
☐ Yes ☐ No

Q14: Where will you share your documentation?
☐ Personal website or blog ☐ Online DIY community
☐ Interest groups or internal website ☐ Other ______________

Q15: I document my projects and share the documentation to...

	Strongly Disagree				Strongly Agree
Showcase my ideas and skills					
Find employment opportunities					
Learn new concepts					
Receive feedback about my own projects					
Get inspiration and new ideas for future projects					
Educate others, share information					
Document or archive my work					
Meet people who share similar interests as me					
Give back to the community					
Other					

Q16: If possible, will you look at documentation of other people's projects created at Fab Lab Aachen? And why?

Q17: Do you expect others to do the following to your projects?
☐ Comment ☐ Rate ☐ Replicate ☐ Fork

2

Figure A.2: Questionnaire for the initial study, page 2 of 3

Q18: Which of the following factors (if any) **deter** you from sharing your work?

	Strongly Disagree				Strongly Agree
My projects are not interesting					
My projects are not novel or creative					
My projects are too easy or simple					
I don't have enough time					
I don't have the skills to edit, upload and share my work					
I don't want other people to 'steal' my ideas (Intellectual Property)					
I don't want my work to be critiqued					
My projects are too advanced or complex					
Give back to the community					
Other					

3

Figure A.3: Questionnaire for the initial study, page 3 of 3

Fabiji - Post Session Questionnaire

Participant ID: ____________

Part I: User Background

Q1: What is your age?

Q2: What is your gender?

Q3: What is your occupation?

Q4: How many times have you been to Fab Lab Aachen (including today)?

Q5: What is your iPad experience?

☐ No experience ☐ I've used it a few times before ☐ I used it quite often

☐ I use iPad every day

Part II: Documentation and Sharing

I think with Fabiji app and the photo kiosk…

	Strongly Disagree				Strongly Agree
S1: Fewer tools are needed to create project documentation					
S2: Less time is needed to create project documentation					
S3: I feel it is a burden to create documentation at the fab lab					
S4: I find the step-by-step guide helpful to create documentation					
S5: I feel Fabiji is an integral part of fab lab					
S6: I will create more project documentation					
S7: I can see more projects created at the fab lab by other people					
S8: My projects can be viewed by more people					
S9: I can easily find documentation I am interested in					
S10: I would like to meet more people and see their projects at the fab lab					

1

Figure A.4: Post session questionnaire for the final evaluation.

Appendix B

Storyboard and Paper Prototype

This appendix includes a pair of storyboards and a paper prototype of the iPad application.

B.1 Storyboard

This pair of storyboards illustrates that Fabiji is able to connect people in real life.

B.2 Paper Prototype

The paper prototype includes five views of the first iPad application user interface sketch.

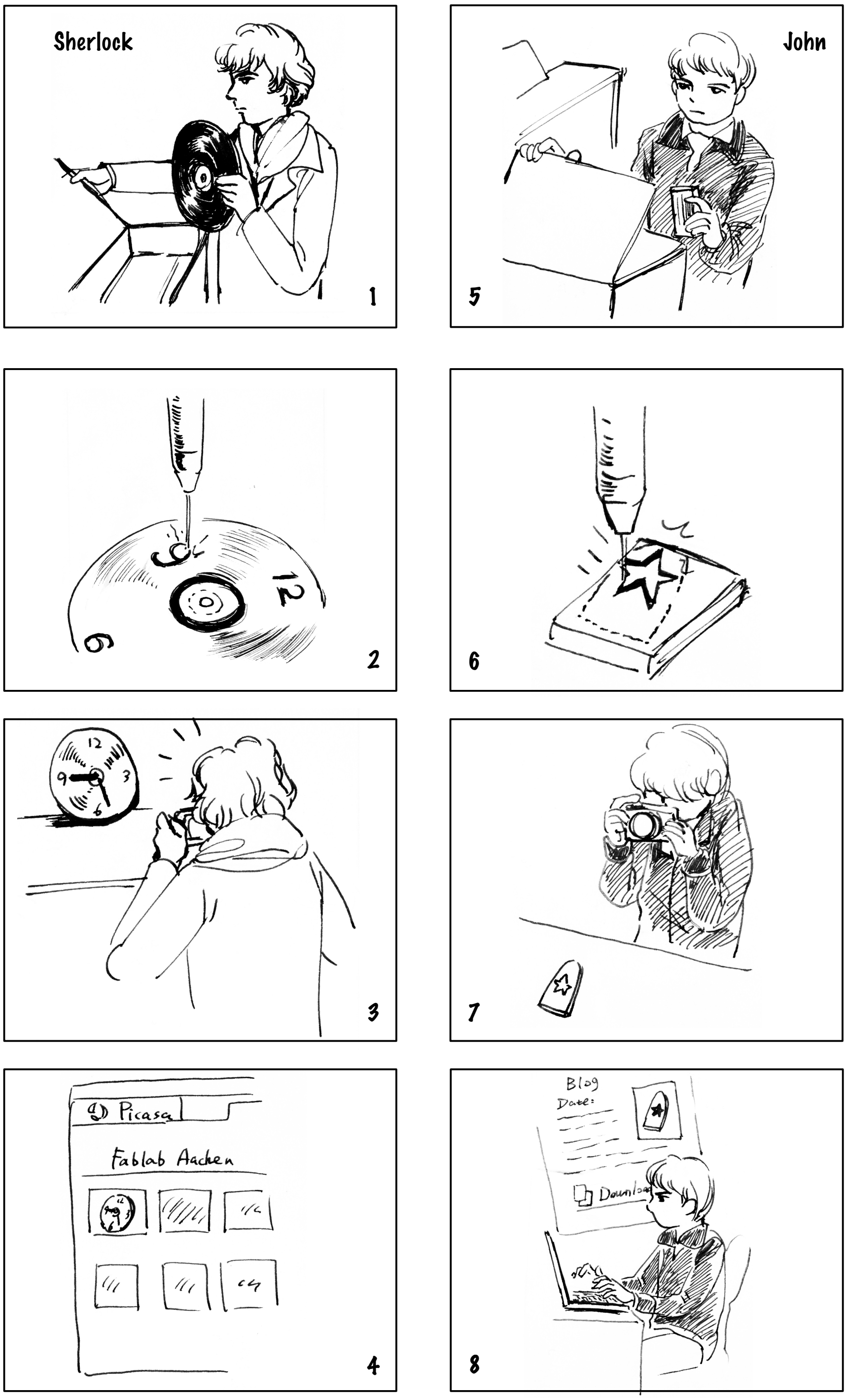

Figure B.1: Storyboard: Without Fabiji, Sherlock and John work at the fab lab on different days. John takes a picture after finishing cutting his record, and upload the picture to the fab lab web gallery. John also takes picture and post the picture to his blog. They don't know each other's projects because they share projects at different places.

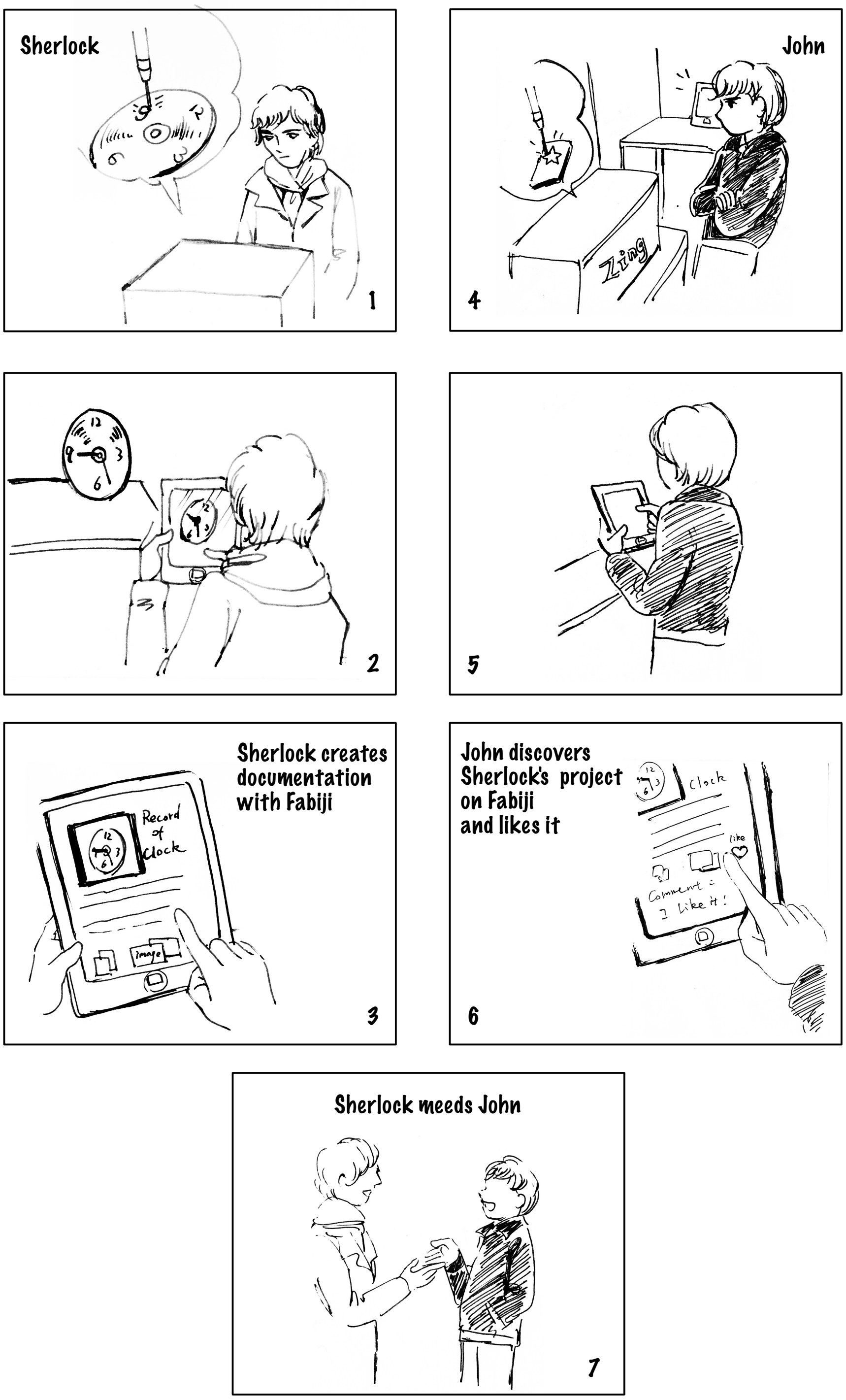

Figure B.2: Storyboard: With Fabiji, John discovers interesting projects created by Sherlock, and they decide to arrange a meeting to discuss collaboration possibilities.

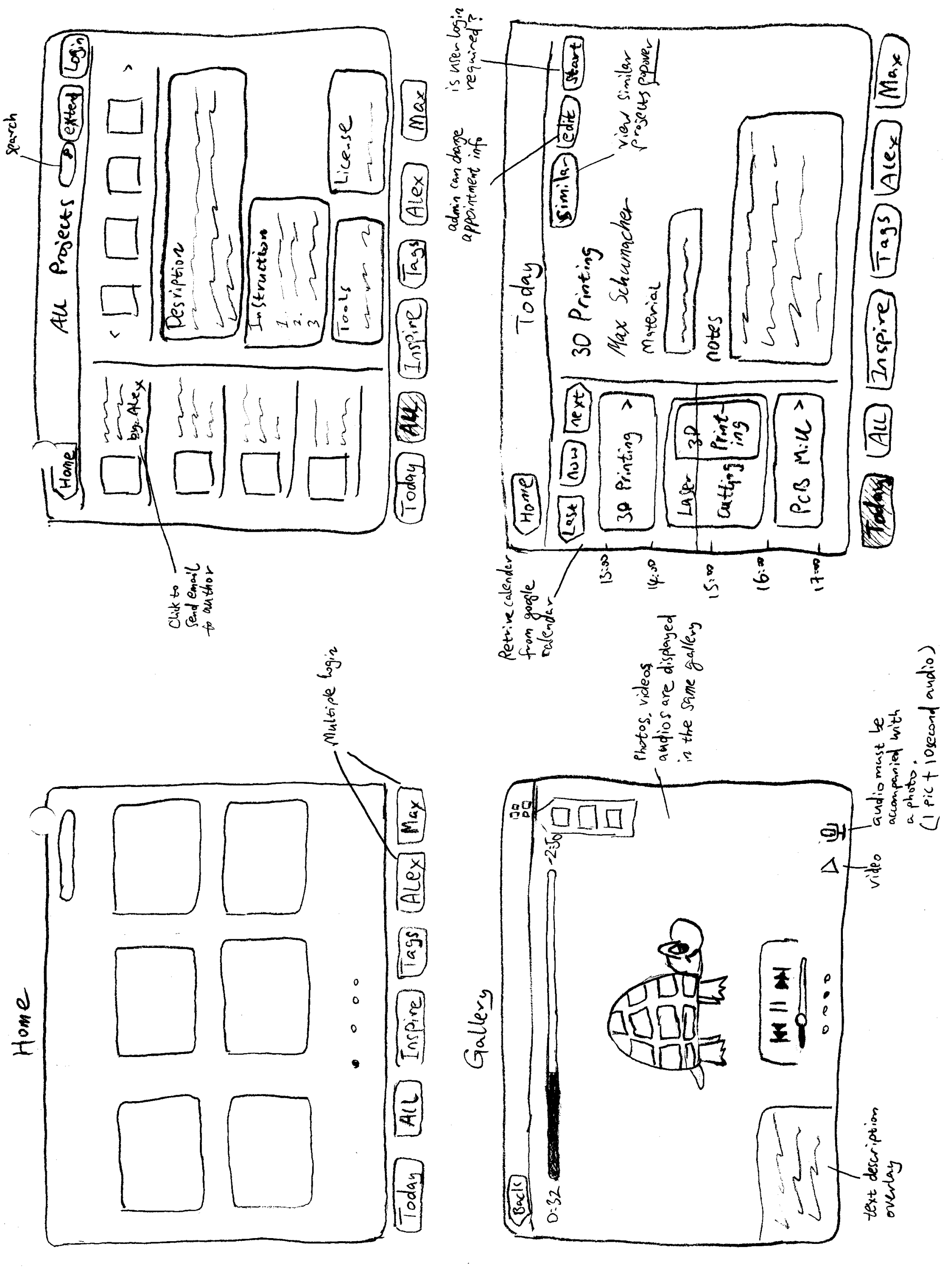

Figure B.3: Paper Prototype, page 1 of 4: Home view, all projects view, gallery view, today view

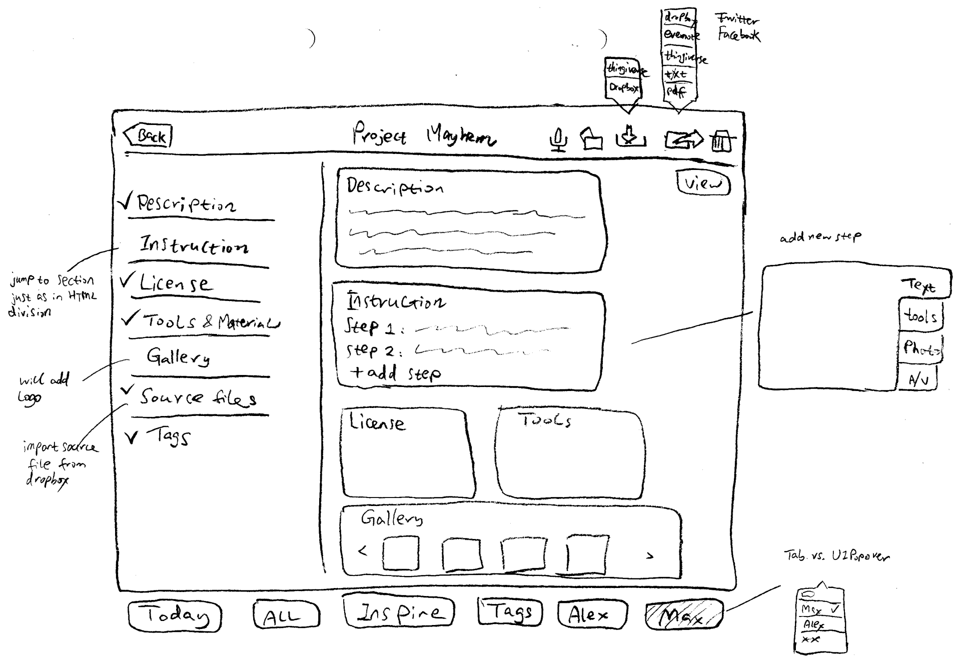

Figure B.4: Paper Prototype, page 2 of 4: Project editing view

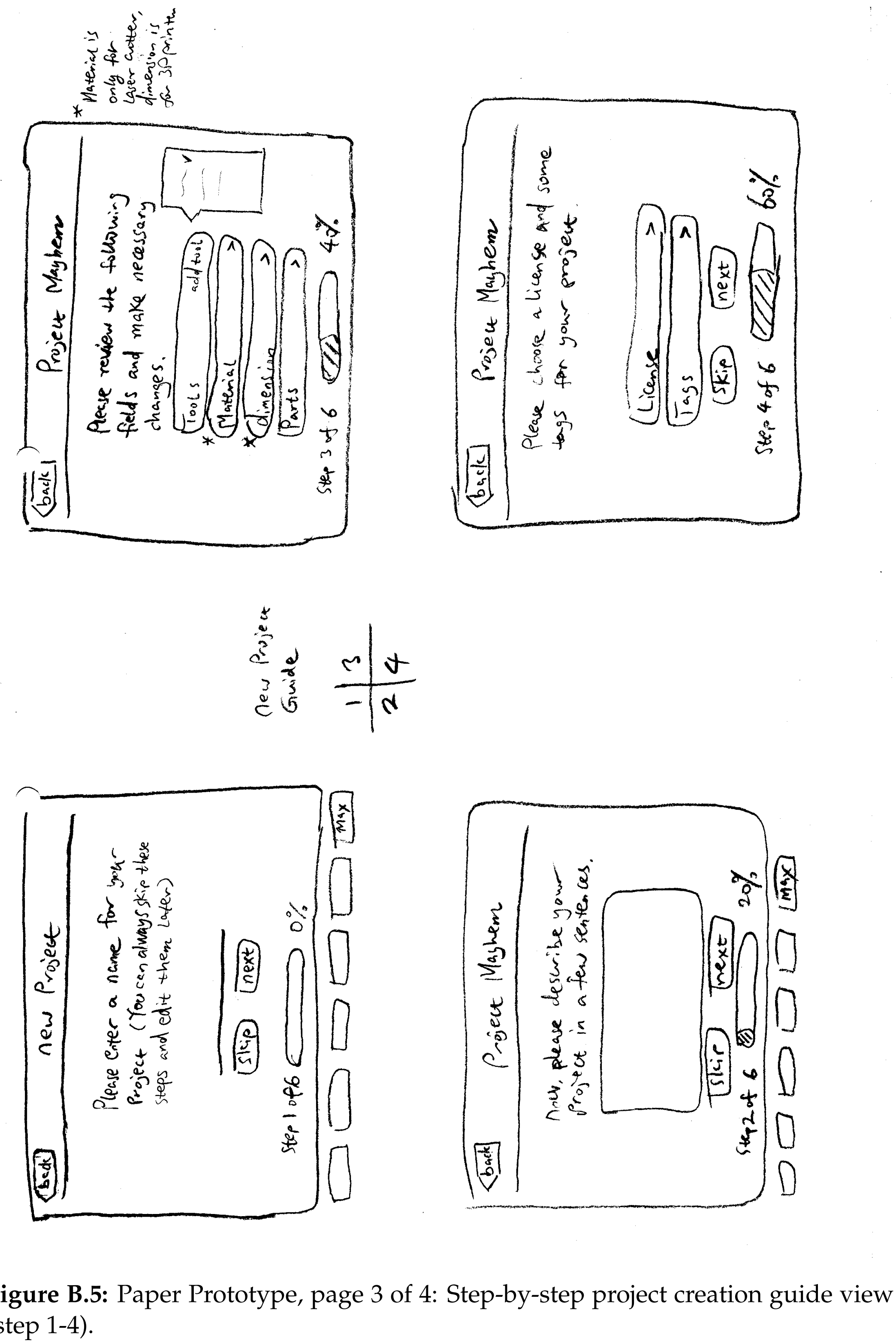

Figure B.5: Paper Prototype, page 3 of 4: Step-by-step project creation guide view (step 1-4).

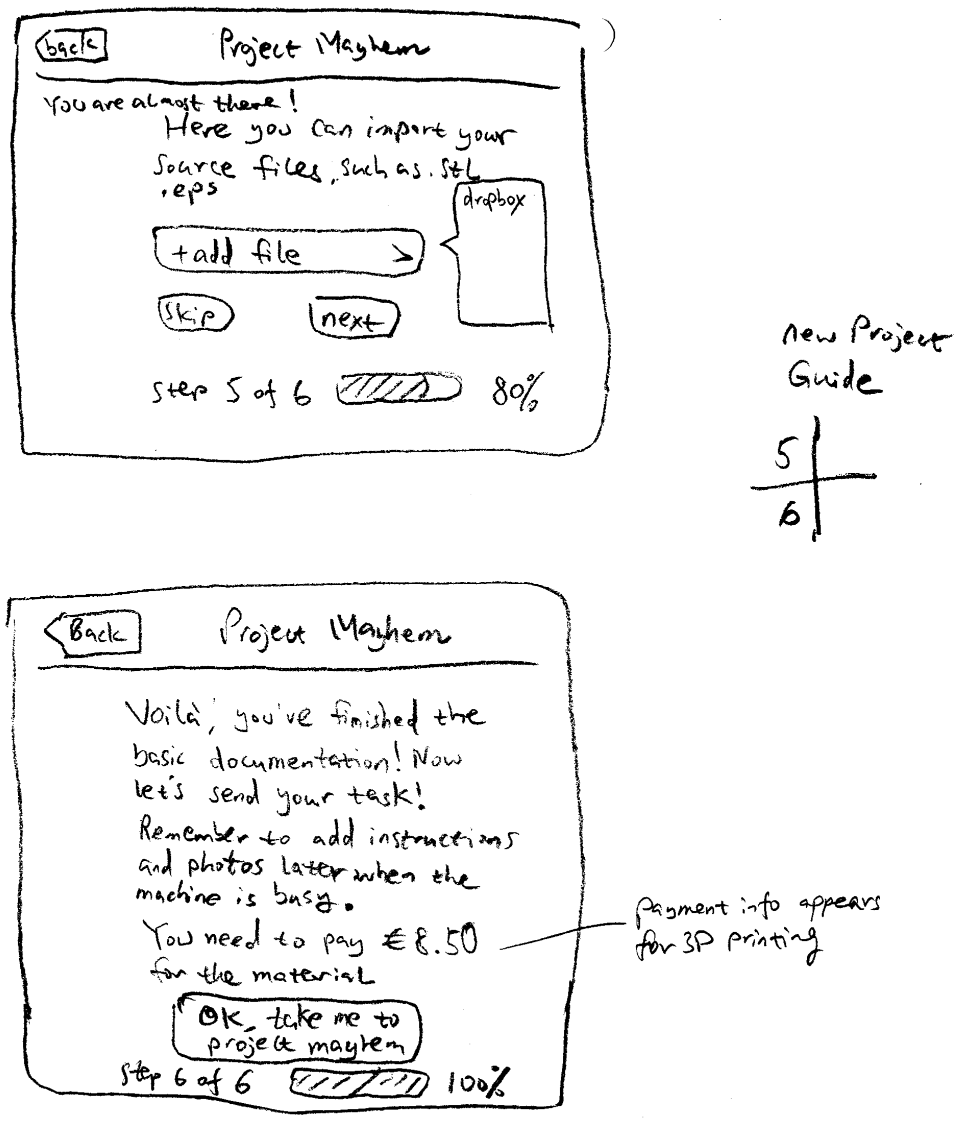

Figure B.6: Paper Prototype, page 4 of 4: Step-by-step project creation guide view (step 5-6).

Appendix C

Images

C.1 Screenshots

This section includes five screenshots form the final version of the iPad application.

C.2 Photos taken from iPad

This section includes a comparison of photographs taken from the iPad.

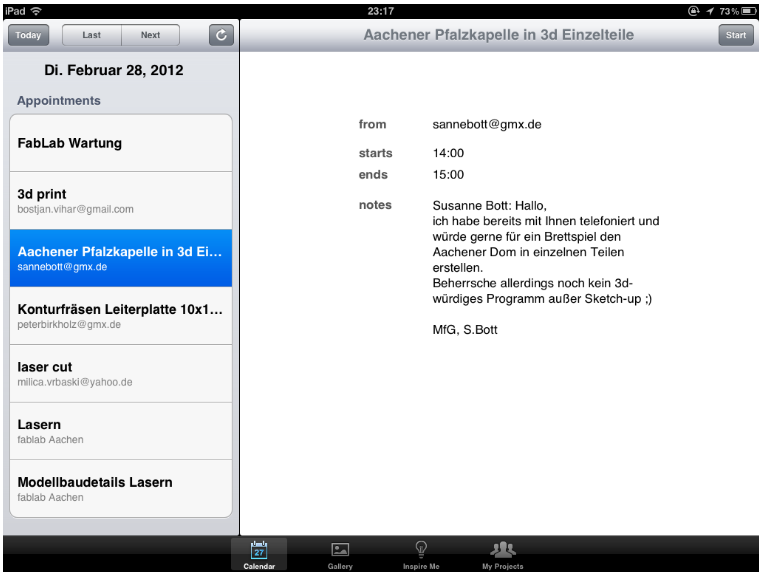

Figure C.1: Screenshot of Calendar tab from the iPad application.

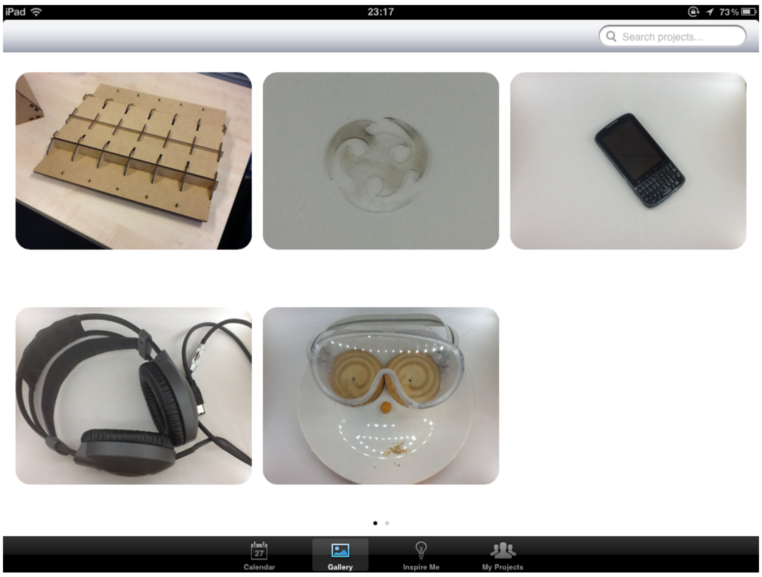

Figure C.2: Screenshot of Gallery tab from the iPad application (project documentation created by test users).

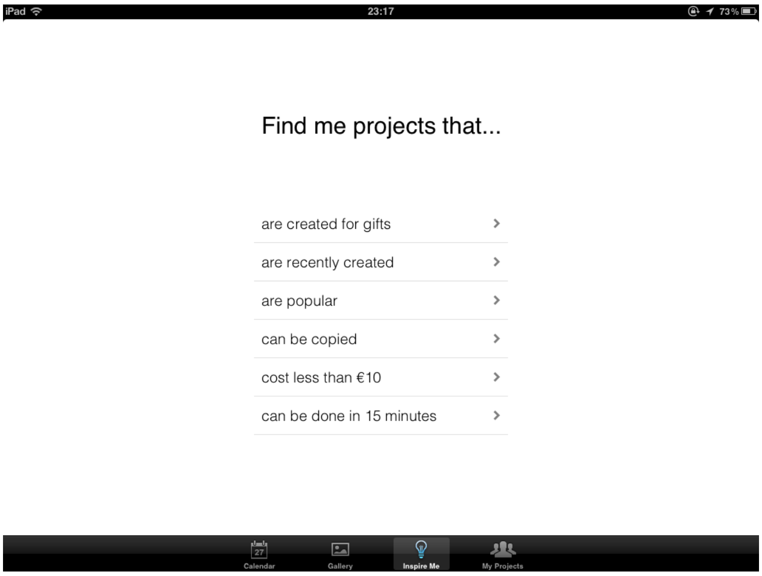

Figure C.3: Screenshot of Inspire Me tab from the iPad application.

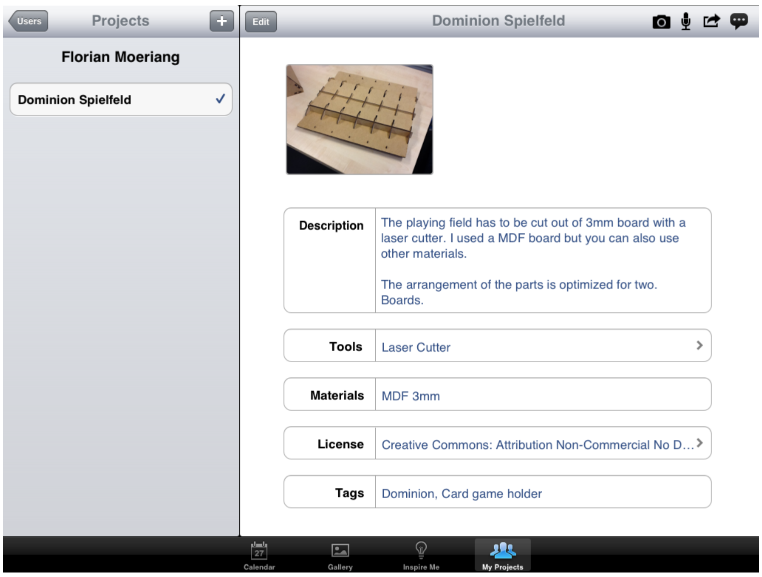

Figure C.4: Screenshot of My Projects tab from the iPad application (project documentation created by a test user).

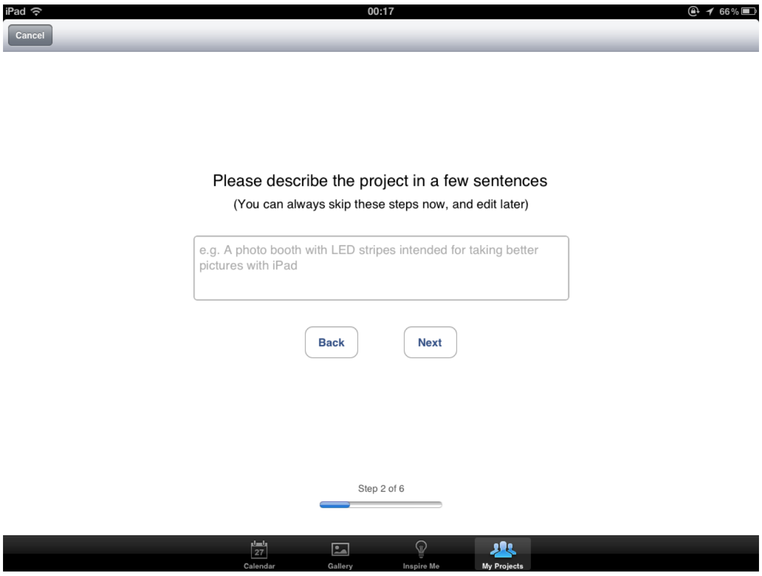

Figure C.5: Screenshot of the step-by-step project creation guide from the iPad application (project documentation created by a test user).

Figure C.6: Example photograph (unprocessed jpeg image) taken from the iPad using the photo kiosk, with white background and LED lighting.

Figure C.7: Example photograph (unprocessed jpeg image) taken from the iPad without using the photo kiosk, on the workbench with normal room lighting.

Appendix D

Source Files

The Xcode project containing all source code of the iPad application is available for download:

File: Fabiji[a]

[a]http://hci.rwth-aachen.de/~he/thesis/Fabiji.zip

The laser cut parts of the photo kiosk is available for download:

File: Lasercut[a]

[a]http://hci.rwth-aachen.de/~he/thesis/Lasercut.zip

Bibliography

A. Bangor, P. Kortum, and J. Miller. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, 2008.

J. Brooke. SUS: A Quick and Dirty Usability Scale. In P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, editors, *Usability evaluation in industry*. Taylor and Francis, London, 1996.

A. M. Burger, B. D. Meyer, C. P. Jung, and K. B. Long. The Virtual Notebook System. In *Proceedings of the third annual ACM conference on Hypertext*, HYPERTEXT '91, pages 395–401, New York, NY, USA, 1991. ACM.

V. Bush. As We May Think. *Atlantic Monthly*, 1945.

A. P. Conn. Time Affordances: The Time Factor in Diagnostic Usability Heuristics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '95, pages 186–193, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

R. T. Fielding and R. N. Taylor. Principled Design of the Modern Web Architecture. *ACM Trans. Internet Technol.*, 2(2):115–150, May 2002.

T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an Agent Communication Language. In *Proceedings of the third international conference on Information and knowledge management*, CIKM '94, pages 456–463, New York, NY, USA, 1994. ACM.

B.J. Fogg. *Persuasive Technology: Using Computers to Change What We Think and Do*. Morgan Kaufmann series in interactive technologies. Morgan Kaufmann Publishers, 2003.

N.A. Gershenfeld. *Fab: The Coming Revolution on Your Desktop–from Personal Computers to Personal Fabrication*. Basic Books, 2005.

M. Guzdial, N. Rappin, and D. Carson. Collaborative and Multimedia Interactive Learning Environment for Engineering Education. In *Proceedings of the 1995 ACM symposium on Applied computing*, SAC '95, pages 5–9, New York, NY, USA, 1995. ACM.

J. Gwizdka, J. Louie, and M.S. Fox. EEN: A Pen-Based Electronic Notebooks for Unintrusive Acquisition of Engineering Design Knowledge. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1996. Proceedings of the 5th Workshop on*, pages 40 –46, jun 1996.

R. Harper, L.A. Palen, and A.S. Taylor. *The Inside Text: Social, Cultural and Design Perspectives on SMS*. Kluwer International Series on Computer Supported Cooperative Work. Springer, 2005.

J. Hong, G. Toye, and L.J. Leifer. Personal Electronic Notebook with Sharing. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1995., Proceedings of the Fourth Workshop on*, pages 88 –94, apr 1995.

S. Kuznetsov and E. Paulos. Rise of the Expert Amateur: DIY Projects, Communities, and Cultures. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 295–304, New York, NY, USA, 2010. ACM.

C. Lewis and J. Rieman. *Task-Centered User Interface Design: A Practical Introduction*. 1994.

J. R. Lewis and J. Sauro. The Factor Structure of the System Usability Scale. In *Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International 2009*, HCD 09, pages 94–103, Berlin, Heidelberg, 2009. Springer-Verlag.

C. Lindinger, R. Haring, H. Hörtner, D. Kuka, and H. Kato. Mixed Reality Installation 'Gulliver's World': Interactive Content Creation in Nonlinear Exhibition Design. In Stefan Göbel, Rainer Malkewitz, and Ido Iurgel, editors, *Technologies for Interactive Digital Storytelling and Entertainment*, volume 4326 of *Lecture Notes in Computer Science*, pages 312–323. Springer Berlin / Heidelberg, 2006.

A. Määttä and P. Troxler. Developing Open & Distributed Tools for Fablab Project Documentation. In *OKCon*, 2011.

D. A. Mellis, D. Gordon, and L. Buechley. Fab FM: the Design, Making, and Modification of an Open-Source Electronic Product. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, pages 81–84, New York, NY, USA, 2011. ACM.

C. 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.

B. A. Myers. The Importance of Percent-Done Progress Indicators for Computer-Human Interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '85, pages 11–17, New York, NY, USA, 1985. ACM.

I. Posch, H. Ogawa, C. Lindinger, R. Haring, and H. Hörtner. Introducing the FabLab as Interactive Exhibition Space. In *Proceedings of the 9th International Conference on Interaction Design and Children*, IDC '10, pages 254–257, New York, NY, USA, 2010. ACM.

C. Snyder. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. The Morgan Kaufmann Series in Interactive Technologies. Morgan Kaufmann, 2003.

G. Toye, M.R. Cutkosky, L.J. Leifer, J.M. Tenenbaum, and J. Glicksman. SHARE: a methodology and environment for collaborative production development. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1993. Proceedings., Second Workshop on*, pages 33 – 47, apr 1993.

M. Weiser. The Computer for the 21st Century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, July 1999.

D. Wixon, K. Holtzblatt, and S. Knox. Contextual Design: An Emergent View of System Design. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, CHI '90, pages 329–336, New York, NY, USA, 1990. ACM.

Index