

# Development of mobile application to handle e-books for Malmö City Library

Elise Ellerstedt & Miriam Tisander

---

2015

Master's Thesis

Department of Design Sciences  
Lund University



# Acknowledgments

We would like to thank our supervisor at Jayway Jonas Flodqvist, for his support; Andreas Lund, for all the iOS help; Per Sigurdson and Andreas Kristiansson for your feedback; and Niklas Lundberg, for explaining the inner workings of Elib.

Special thanks go to Mimmi Widner from Malmö City Library for all your helpful feedback, comments and explanations about the library field and its bureaucracy limitations.

We want to express our gratitude to our supervisor at Lund Institute of Technology Joakim Eriksson and our examiner Kirsten Rasmus-Gröhn.

Our thanks go to all the volunteers who participated in our user studies.

Finally, we would also like to thank our friends and family for being there.

*Malmö, Jayway, June 2015*

*Elise Ellerstedt & Miriam Tisander*

# Abstract

The number of people who choose e-books rather than reading traditional paper books are increasing as a result of our world becoming more digitized. Libraries need to keep up with this trend and modernize its lending procedures for their digital products. The purpose and goal of this Master thesis were to answer the questions how e-books can become more attractive to those who have not yet dared to read them and how to make the whole experience more user friendly. There already exists a couple of solutions for borrowing e-books today, but the ones recommended by the public libraries are unnecessarily complicated which discourage users with lower technical confidence.

This Master thesis explored solutions for these problems and showed an alternative approach where both the borrowing and reading features were supported in the same application. The development process was divided into four parts conducted over 20 weeks. The project started with mapping desired functionalities through evaluation of the current solution offered by the public library, conducting user surveys and by analyzing existing e-book applications. A design concept was then formed based on these mapping results by making a Lo-Fi prototype and then later a Mid-Fi prototype. These prototypes became the groundwork for the result of the thesis project, an executable Hi-Fi prototype application with a simulation of a loan. The final prototype was then evaluated through user studies, tests and compared to the current solution. No general results could be drawn from the user studies since only a small group could participate. For future development, tests should be conducted with the primary target group, consisting of persons already reading e-books who are also interested in borrowing. Such tests will result in more valuable feedback.

The conclusion of this thesis is that there are a number of obstacles ahead before a satisfactory solution is found. User friendly design is important, but in the end reading functionality and a sufficient enough selection of books are the most important aspects. Limitations due to the DRM protection and the bureaucracy rules of the libraries cause a lot of problems in making this a reality. To get around these restrictions and finding alternative solutions is in the end a question of getting more financing for the libraries.

**Keywords:** e-books, libraries, DRM protection, user friendly design, DL Reader, prototyping

<b>ACKNOWLEDGMENTS .....</b>	<b>2</b>
<b>ABSTRACT .....</b>	<b>3</b>
<b>1. INTRODUCTION .....</b>	<b>7</b>
1.1. JAYWAY.....	7
1.2. GOAL .....	7
1.3. RELATED WORK .....	8
1.3.1. Kaliber .....	8
1.3.2. Datalogics DL Reader .....	10
1.3.3. Survey by Stockholm Public Library.....	10
1.3.4. Article on design preferences for e-books.....	11
1.3.5. Study from the Chinese university in Hong Kong .....	11
1.4. Work process.....	12
1.5. Work distribution.....	13
<b>2. THEORY .....</b>	<b>14</b>
2.1. DEVELOPMENT METHODS.....	14
2.1.1. Scrum .....	14
2.2. DESIGN THEORY .....	15
2.2.1. User centered design .....	15
2.2.2. Accessibility.....	15
2.2.3. Prototyping.....	15
2.2.4. Choice of Mid-fi development program.....	16
2.3. TECHNICAL BACKGROUND .....	18
<i>The following development tools and languages were studied in this thesis work.....</i>	<i>18</i>
2.3.1. Elib .....	18
2.3.2. Objective-C.....	18
2.3.3. JSON .....	18
2.3.4. REST API.....	19
2.3.5. Git.....	19
<b>3. SPRINT 1: "ANALYSIS AND DESIGN PHASE" .....</b>	<b>21</b>
3.1. CURRENT SOLUTION.....	21
3.2. SURVEY.....	21
3.2.1. Results from the survey.....	22
3.3. REVIEWING EXISTING DIGITAL MEDIA BROWSERS .....	25
3.3.1. The demo application of DL Reader .....	25
3.3.2. Bluefire Reader .....	26
3.3.3. Adlibris Mondo.....	27
3.3.4. Kindle .....	28
3.3.5. Netflix and Spotify.....	29
3.3.6. Bokon .....	31
3.4. FEATURE LIST.....	32
3.5. PROTOTYPING .....	32

<b>4. SPRINT 2: “ SETUP AND BASE DESIGN PHASE”</b> .....	<b>35</b>
4.1. IOS DEVELOPMENT STUDIES .....	35
4.2. INSTALLATION OF DL READER CODE .....	35
4.3. REDESIGNING THE INTERFACE OF DL READER .....	35
4.4. SPRINT 2 CONCLUSIONS .....	38
<b>5. SPRINT 3: “SYNCHRONIZATION PHASE”</b> .....	<b>39</b>
5.1. SYNCHRONIZATION WITH ELIB .....	39
5.1.1. <i>RestKit vs JSONModel</i> .....	39
5.1.2. <i>SDWebImage vs AsyncImageView</i> .....	40
5.2. EXPANDING THE FEATURES .....	40
5.3. SPRINT 3 USER STUDY .....	43
<b>6. SPRINT 4: “FINAL DESIGN PHASE”</b> .....	<b>45</b>
6.1. SIMULATED LENDING PROCEDURE .....	45
6.2. FINALIZING DESIGN CHOICES .....	46
6.2.1. <i>Navigation and Positions</i> .....	46
6.2.2. <i>Constraints</i> .....	47
6.2.3. <i>Terminology and Font</i> .....	48
6.2.4. <i>Buttons, Icons and Badges</i> .....	49
6.2.5. <i>Colors</i> .....	51
6.3. FINAL USER TEST .....	52
6.3.1. <i>Part 1: Analysis of the current system</i> .....	52
6.3.2. <i>Part 2: User scenarios</i> .....	53
6.3.3. <i>Part 3: Debriefing</i> .....	53
6.3.4. <i>Part 4: Survey</i> .....	53
<b>7. DISCUSSION</b> .....	<b>56</b>
7.1. IMPLEMENTATION PROBLEMS .....	56
7.1.1. <i>DL Reader and its installation</i> .....	56
7.1.2. <i>Xcode and iOS version changes</i> .....	57
7.1.3. <i>Git synchronization with Xcode</i> .....	57
7.2. EVALUATION OF METHODS USED .....	57
7.3. IS THE ERA OF BORROWING E-BOOKS AHEAD? .....	58
<b>8. CONCLUSION</b> .....	<b>60</b>
<b>9. REFERENCES</b> .....	<b>61</b>
<b>APPENDIX</b> .....	<b>65</b>
A.1 SURVEYS .....	65
A.1.1 <i>Survey version 1</i> .....	65
A.1.2 <i>Survey version 2</i> .....	71
A.2 FEATURE LIST .....	79
A.3 THE MID-FI DESIGN MADE IN BALSAMIQ .....	84
A.4 FINAL USER STUDY .....	88

<i>A.4.1 Part 1 - Analysis of the current system .....</i>	<i>88</i>
<i>A.4.2 Part 2 - User scenarios .....</i>	<i>89</i>
<i>A.4.3 Part 3 - Post-test debriefing - Interview.....</i>	<i>89</i>
<i>A.4.4 Part 4 - Survey .....</i>	<i>91</i>

# 1. Introduction

Our world gets more digitized every day. New solutions and technologies are presented to simplify our everyday life. People are constantly on the move and to be able to reach information wherever you are is an attractive possibility. Some businesses are however hard to contemporize. Printed books have existed since before the fifteenth century and reading is an extracurricular activity many people have in common. The question is how can you modernize this traditional hobby? Since there is such a high quantity of literature published today, many readers prefer to lend books from the public libraries rather than always open up their wallets. E-books have become even more popular over the last few years with the rise of technology. According to a survey conducted by the public library of Stockholm every fifth borrower obtains a library card only with the mere purpose of the digital selection of the libraries [1]. The readers want to be able to find, lend and read books anywhere.

This Master thesis work was introduced by the company Jayway after a request from the public library of Malmö and Stockholm. The libraries wanted a mobile application in which the users can find, lend and read e-books all in the same application. There already exist a number of different solutions handling e-books that Jayway has developed, e.g. Dito, the e-book service of Bokus; and Elib, a portal for libraries used to borrow books. Unfortunately, none of the applications on the market at the moment have an integrated overall solution supporting both lending and reading. Today the users have to download a separate application to read the book. For borrowers with scarce technical experience these solutions provide low usability and a complex lending process.

## 1.1. Jayway

Jayway is a software innovation consultancy and development partner with offices in Stockholm, Malmö and five other cities around the world. Jayway helps their clients with all steps required for creating digital products (such as applications and websites) and service experiences, from the idea stage to a finished product through a close creative collaboration with their customers.

They are also keen on sharing their knowledge and skills by regularly writing posts on their blogs and by having technical seminars at their offices that are open for the public to observe and participate in. Even more notable is the fact that they organize an annual international developer conference called Øredev with about 1500 unique participants throughout the week it takes place [2].

## 1.2. Goal

The purpose of this project was to investigate how the libraries should attract more borrowers to their digital products. The current solution used for borrowing consists of many steps, which

may make the users less inclined to try it. A new technical solution that simplifies the user experience, with an interface that is self-explanatory and congenial, should therefore be researched. This solution is intended to become the proof-of-concept for a national e-book technology offered by Swedish public libraries.

The main goal was to develop a Hi-Fi prototype for mobiles that offers functionalities for both lending and reading e-books. Usability has been an important aspect since the application is aimed at a large target group. Both young as well as old book lovers should easily be able to find the book they are longing to read. Because of that a large part of the Master thesis was dedicated to designing. Since the assumption is that users will mainly read their e-books on tablets, the design was mostly directed to iPads but can function as a responsive layout and therefore adapt to smaller screens.

The long term goal, to attract more loyal users who return to the application often, can be achieved by introducing social aspects. The users should be able to share what they are reading and get inspired by other users. The greater vision is that the finished product, which is suppose to be based on the prototype presented in this report, will become a successful book version of Spotify or Netflix.

### 1.3. Related work



Figure 1.1 The pilot study.

#### 1.3.1. Kaliber

Jayway conducted a pilot study (see figure 1.1) in 2013 for Malmö City Library with support by Skåne Regional Council to find out how to make it easier to borrow e-books and what obstacles that exist today. The study involves a technical investigation of the cost and possibilities of making such an application a reality and a concept demo (see figure 1.2 and 1.3) with the Flinto prototyping tool [3]. The demo is a light Hi-Fi prototype with only a partial coverage of the desired functionality and does not include any functionality or implementation of any kind. The



result of the pilot study was that it is possible for this kind of application to be implemented at a reasonable cost and that the technology for it exists [4].

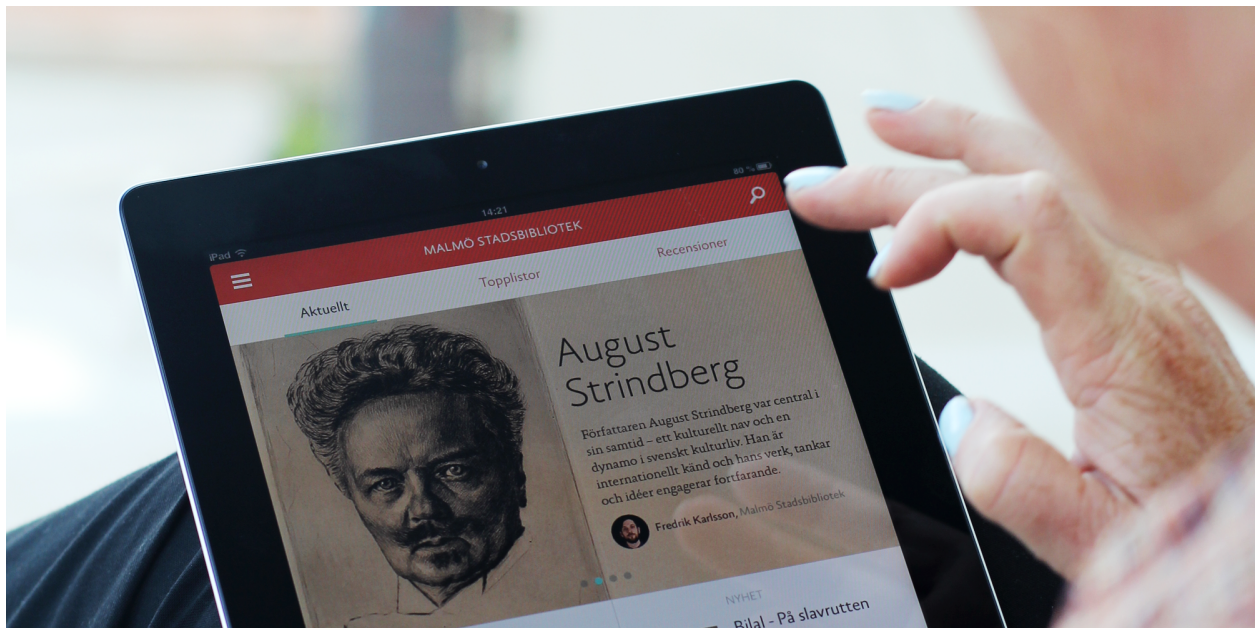


Figure 1.2 The Kaliber demo application.

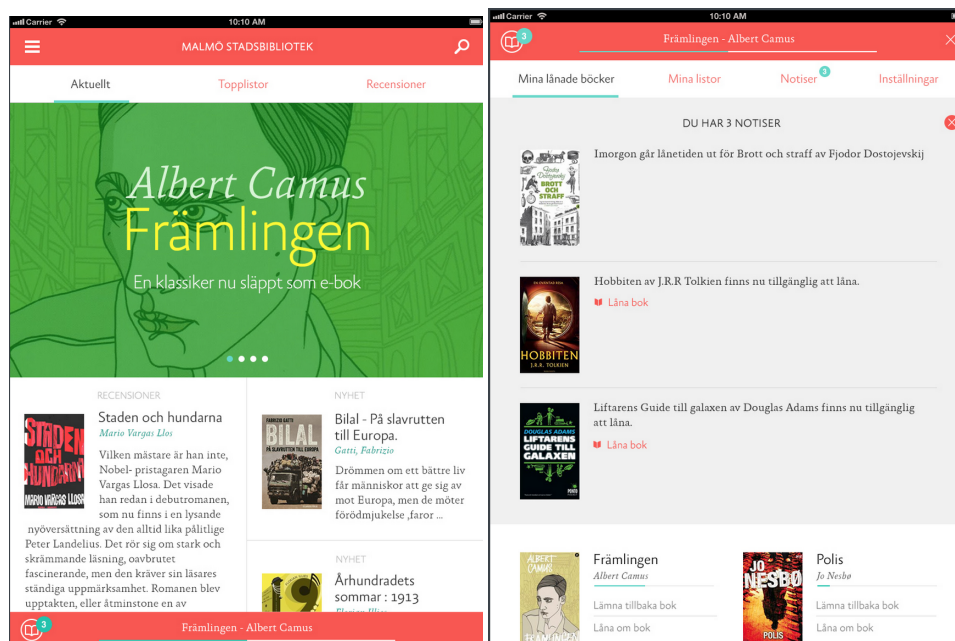


Figure 1.3 The Kaliber demo application

### 1.3.2. Datalogics DL Reader

Datalogics has developed an application called DL Reader that is a customizable e-reading application that publishers, libraries and e-book stores can use to modify, customize and further develop to make their own and branded e-book viewer. The application can with the help of Adobe technology open and download Digital Rights Management (DRM) protected epub files. This technology is therefore needed to be able to borrow digital books from a library [5].

### 1.3.3. Survey by Stockholm Public Library

The public library of Stockholm performed an evaluation in which they analyzed how borrowers relate to e-books. The survey brought up several questions and aspects concerning both reading and borrowing of digital literature [6].

Most participants were positive, 8 of 10 believe that they will borrow more digital literature in the future. However, the limitations of how many books that may be borrowed each month, and the range of books provided were however criticized. There were also concerns about the fees of the publishers on e-books for both libraries and consumers; and the fact that readers have to wait until newly published books are available as e-books because of the quarantine that publishers apply to new books [7].

To visualize their analysis Stockholm Public Library compiled a set of diagrams. The two that were taken in consideration during this thesis are presented below. Figure 1.4 show that 46% of the persons studied read e-books on their iPads. Figure 1.5 presents a description of the book (81%) and a cover image of the book (52%) were both desired functionalities.

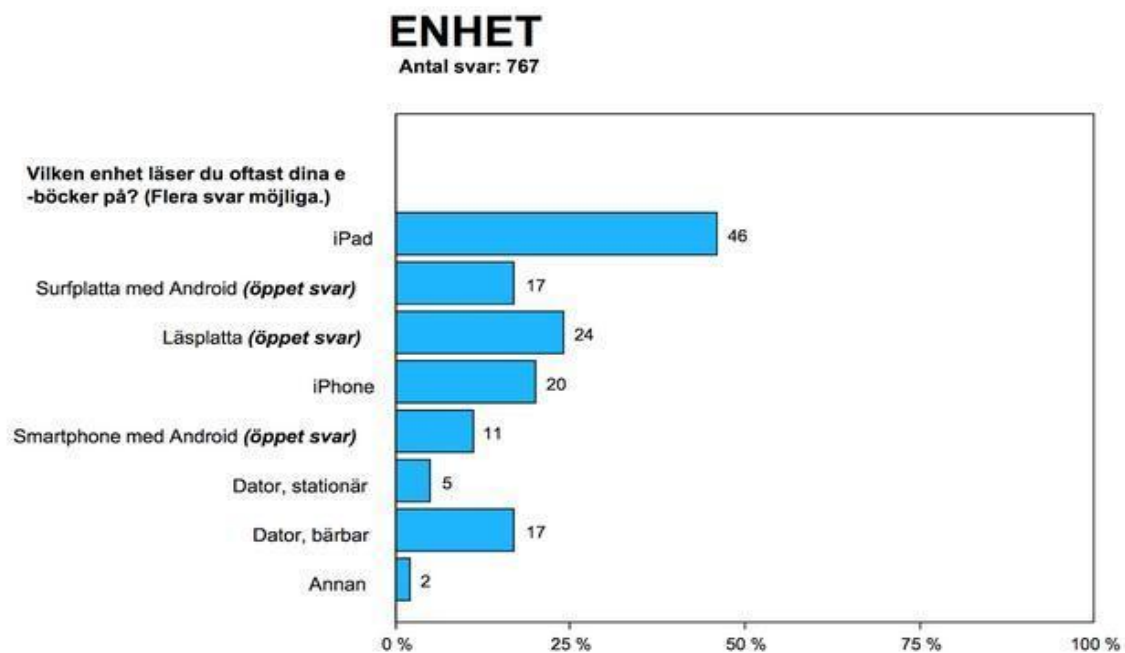


Figure 1.4 "Which device do you use to read your e-books?" Options from top to bottom read: iPad, Android tablet, E-reader, iPhone, Android smartphone, Desktop computer, Laptop, Other.

# Borrowing and reading habits

Number of answers: 750

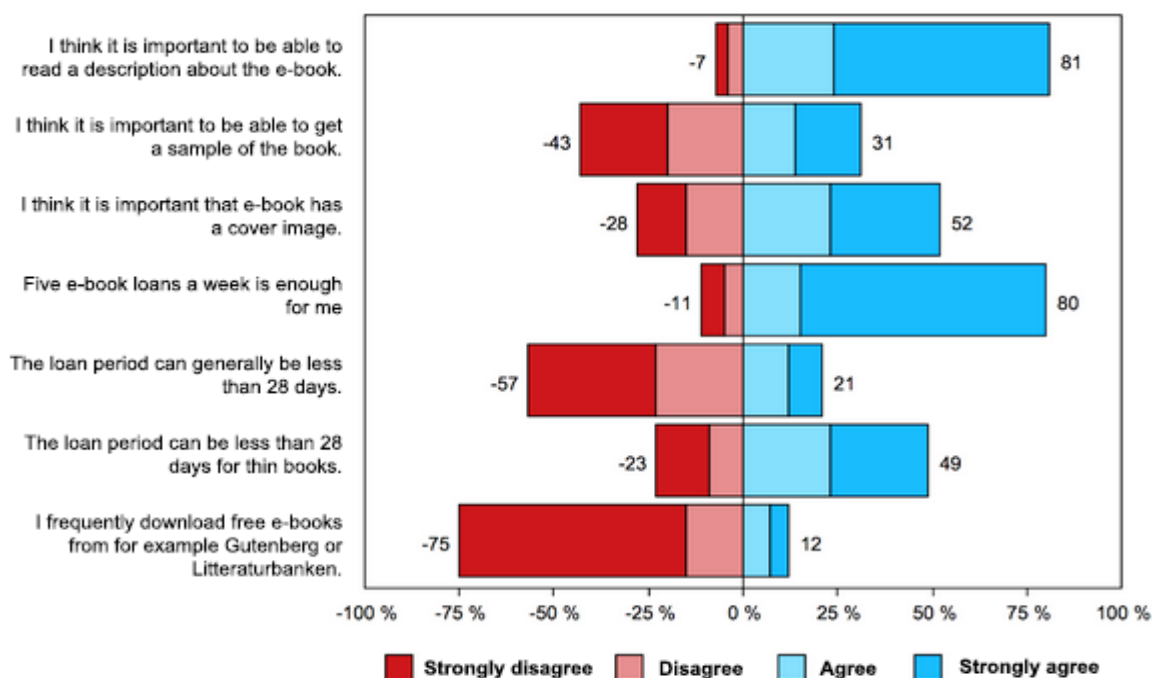


Figure 1.5 "Borrowing and reading habits for e-books".

## 1.3.4. Article on design preferences for e-books

Three Malaysian students made a research and wrote an article about how users prioritize certain layout and design choices for e-books [8]. The design of three different e-books were compared and according to this article functionality like being able to highlight text and annotation was very important, especially for academical books. Additionally, navigation functionality like marking the user's place in an e-book and showing page numbers were stated as something many e-books were lacking. Users rely on being able to track their progress, to be able to backtrack and return to certain sections, this should therefore be a must. Another necessary functionality mentioned was the ability to search in the text. All this combined results in an overall ease of use that is highly important for a successful reading application.

## 1.3.5. Study from the Chinese university in Hong Kong

Hong Kong has one of the highest rates of ownership of mobile devices in the world. So digital literature like newspapers are greatly valued. However, reading digital texts are not as spread in the academic world. A study was performed at the Chinese university in Hong Kong during a period between 2006-2008. Twelve students from different educations, divided into two groups, were asked to participate. The first group got an introduction to e-books and later reported their experiences with the technology. The other group was monitored during a twelve week period by the research team, while each reading a different e-book.

The results from the study showed that digital literature could be beneficial to the students' learning process. However, some problems need to be handled to increase the user experience for long term readers. The students belonging to the first test group were more positive to the technology than those of the second group who used e-books for a longer time. It also became obvious that students of science or engineering were more comfortable with the technology. Some students thought that a couple of features were lacking, like being able to highlight text. Even though, the advantages of e-books were mentioned. Most students read their course literature commuting to and from the university, so the perks of not having to carry heavy books were highly appreciated [9].

#### 1.4. Work process

The process of this Master thesis was divided into sprints according to the agile project method Scrum (further explained in paragraph 2.1). To get an overall perception of the work progress a Scrum board was created. The sprints were divided into separate Scrum boards with help of the program Trello (paragraph 2.1.1), in order to get a better insight of the upcoming tasks. A whiteboard was also used for scheduling the sprints.

The difference between these was that the whiteboard also showed a calendar view with all the weeks and days of the Master thesis. For that reason, and that the board could easily be viewed by other co-workers and thesis workers in the office, made it a good complement to Trello. A daily diary of the work was written in a Google doc throughout the entire thesis with a short summary of each task done throughout the day. The diary was of great use when writing this thesis report.

The work process consisted of four sprints (see figure 1.6). During the first sprint a mapping of desired functionalities and design concepts was performed. The initial Lo-Fi prototype was sketched and transformed into a Mid-Fi prototype using the tool Balsamiq. These prototypes became the groundwork for the Hi-Fi prototype that was developed during the three subsequent sprints. The prototype was accurately evaluated through user studies. More information can be read in the chapter of each sprint.

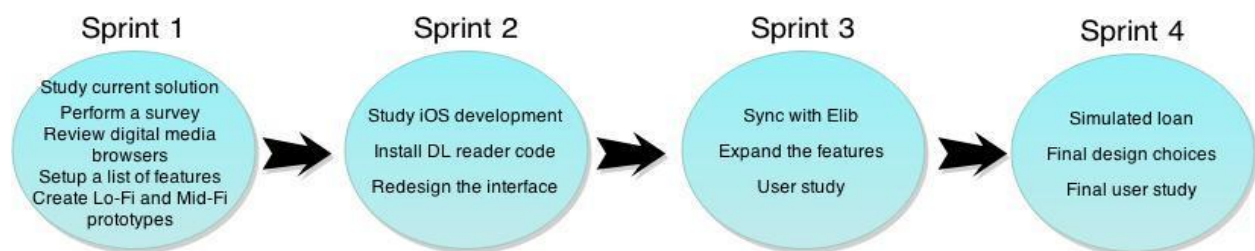


Figure 1.6 Work process

## **1.5. Work distribution**

The work was evenly divided between the thesis workers. Elise focused on and implemented the code for the Search view and the floating Sort and Grid buttons, while Miriam wrote the code for the Malmö City Library page and the book information window. In turn, Elise designed all the Mid-Fi images in correspondence to discussions and decisions made by both workers. In the meantime, Miriam learned about REST and was the one in charge of the synchronization against the Elib APIs. The rest of the project, like testing and writing this report, was all made together.

## 2. Theory

Before the sprints mentioned above could take place an overall knowledge of the development environments and its procedure was studied. The similar solutions out on the market was also compared and researched.

### 2.1. Development methods

The following development methods were used in this thesis work.

#### 2.1.1. Scrum

The agile method Scrum is the most popular method for software developers around the world. It can be combined with other methods such as Kanban and XP. Its main focus lies in the features that are going to be implemented, as opposed to how to implement them. Business advantage is also an important aspect in Scrum. Instead of a static requirement specification list a backlog is used. A backlog consists of a dynamic priority list, which makes it possible for requirements and desired features to be removed from the list and therefore also the product (if their priority descend) or added during the development process. The group consists of a product owner, a Scrum master and the team. The product owner represent the business advantage side of the project. He or she gives the desired features their different priorities. The Scrum master makes sure that the project moves forward and that the team are not held back or blocked by any obstacles. The team decides who does what and how different tasks are going to be solved.

Each morning a daily Scrum meeting is held. Each team member informs the others what he or she has done yesterday, what is planned to be done today and what obstacles he or she faces. The development is divided into several sprints, which span over two to four weeks. At the end of each sprint a demonstration is held where the developers present what they have implemented so far [10].

#### *Trello*

Trello is a tool recommended for constructing Scrum boards. It can be accessed on both Android and iOS devices. After signing up, the user can create different boards in which several lists can be produced. A Scrum board usually contains the lists “Backlog”, “To Do”, “In Progress”, “Under Testing” and “Done” (see figure 2.1). Each task is described within a card. To distinguish between categories of tasks, colored labels are provided. These task cards are placed in the list where they currently belong and moved when their status change [11].

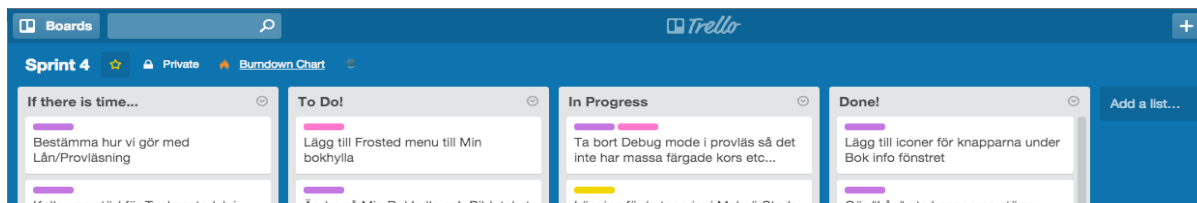


Figure 2.1 Trello boards.

## **2.2. Design theory**

The following design methods and design choices were studied in this thesis work.

### **2.2.1. User centered design**

Developing in a user centered perspective implies that the interface of a website or an application are designed so that it can be understood by a human as opposed to focusing on business goals where the users are presumed to learn the system. The development process is often iterative with the users constantly involved. User centered design results in usable and efficient products that attract loyal users and customers [12].

### **2.2.2. Accessibility**

To be able to reach a great target group of many ages as defined in this project, accessibility must be considered. Apple offers several tools in iOS for those with all types of disabilities so that the applications developed will become usable even for those with special needs [13].

Users with visual impairments are most likely to be affected by problems using book readers. Voice control is highly appraised, but for those who only require large text fonts the possibility to zoom can be used within every application from the App Store. If the reader supports change of font types the font Tiresia could be used. Tiresia is a font that is designed to make life easier for the visually impaired, formed and developed by the British organisation Royal National Institute of Blind People [14].

For users struggling with dyslexia the open source typeface Open Dyslexic can for some users be helpful and could be an alternative font type in all applications, which require reading abilities [15]. Other aids for hearing loss like Mono Audio (one-sided deafness) and Switch Control for motor disabilities are also handled in iOS.

### **2.2.3. Prototyping**

To create a simpler version of an application, prototyping is often used. It helps visualize the features that should be included and how the design could look like. A prototype can either have fewer details and many functions, then called “Horizontal” or be “Vertical” with more details and less functions (see figure 2.2).

The draft should be shown to stakeholders and customers during an early stage of the development, so that they are encouraged to leave their opinions before too much time and money are invested. It is both time consuming and expensive if changes are made in the end of the implementation. The prototypes can be divided into three categories, depending on their complexity and the cost to implement them, “Low-fidelity” (Lo-Fi), “Medium-fidelity” (Mid-Fi) or “High-fidelity” (Hi-Fi) [16].

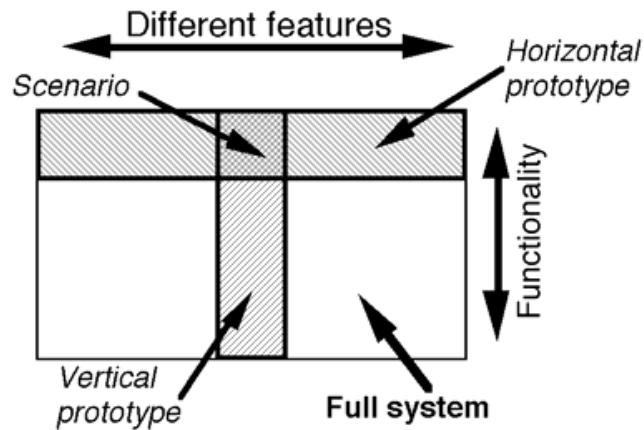


Figure 2.2 Vertical vs. Horizontal prototype

### **Low-fidelity prototypes**

The prototypes developed in low-fidelity are handmade, mostly paper-based and do not include any user interactions. They are easy and quick to create and enable an early visualization of different design solutions. At the beginning of a software development it is beneficial to show these simpler sketches to users since they feel more comfortable giving criticism in this stage.

### **Medium-fidelity prototypes**

Navigation, labels and interaction flow are included in a medium-fidelity prototype. However, it does not convey the final appearance. The focus of this prototype is to display sizes and positions of the elements within each view. Placeholders for images and example texts (Lorem Ipsum) are applied in a high extent.

### **High-fidelity prototypes**

Prototypes in high-fidelity are computer-based and show an almost true version of the user interactions. It contains many features and details. Because of the high complexity of these prototypes much time is spent producing them and therefore is it unlikely that the developers want to change much. If design choices have not been discussed earlier, reviewers tend to comment on small graphical details as oppose to the content. Since these prototypes are highly extensive they make a great tool for performing user tests and evaluate performance data later in the development process.

#### **2.2.4. Choice of Mid-fi development program**

The design choice of using the mockup application Balsamiq for the Mid-Fi design was first made after a few other candidates was compared and researched. Below is the comparison of these programs.



#### **Fluid UI**

One of the first programs considered was Fluid UI. They advertise their mockup tool as an effortless method for making prototypes. Fluid UI is a good choice when a lifelike prototype is desired. Fluid



UI prototypes look very close to the real thing (see figure 2.3), but this might confuse users testing the application and could make them think that all the design and its features are already implemented. Which in turn might censor their honest opinions about the design choices made. Also, the free prototype mode of Fluid UI only allows 10 slides. Accordingly the choice of opting out Fluid UI from the list of candidates was easy [17].

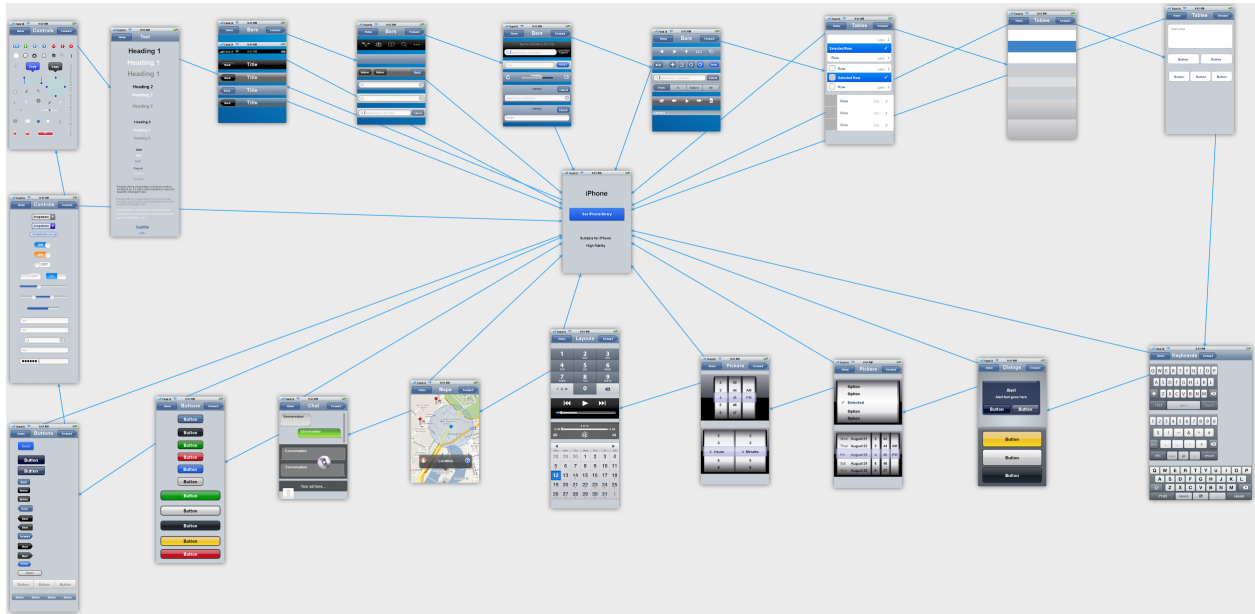


Figure 2.3 Fluid UI



### ***Invision and Flinto***

Another choice could have been to use either Invision or Flinto. Both programs support a more unique design (not necessary in iOS standard) which can be created in non-mockup programs like Photoshop or Illustrator. This is possible since Invision and Flinto makes a wireframe with clickable transparent parts on top of a chosen background image, like for example a photoshop created image of the layout. In this project these functions were not needed and would instead have made the design process slower and these programs were therefore not chosen [18, 19].



### ***Balsamiq***

Instead, Balsamiq was the tool chosen (see paragraph 3.5). The program is free to use for one week, per computer. This was enough with the combination of both a private and a borrowed computer.

Balsamiq was also a good choice because of its discrete colors, sketchy look and feel that makes users prone to give their views and ideas of the design without feeling as “troublesome” since the design is clearly not finished [20].

## 2.3. Technical background

The following development tools and languages were studied in this thesis work.

### 2.3.1. Elib

Elib is the leading producer and distributor of digital books in Scandinavia, with formats including PDF, Epub, Mobipocket and Microsoft LIT in collaboration with digital rights management of e-books such as Adobe Content Server, Mobipocket and Microsoft DAS server. Elib is the provider and distributor that Malmö City Library currently uses.

Their solutions include DRM protection with copy protection, digital watermarking and time-limited DRMs. “Copy protection” prevents users to copy and redistribute their downloaded books. “Digital watermarking” is used to mark a book so that, if needed, it can be backtracked to the user who downloaded it from the server. A time limit on the DRM is essential for libraries that only allows a user to read their distributed books for a certain period of time [21].

### 2.3.2. Objective-C

Objective-C is an object-oriented programming language used by Apple for its operating systems OS X and iOS and their application programming interfaces (APIs) Cocoa (AppKit frameworks) and Cocoa Touch (UIKit frameworks). It is a superset of the C programming language and inherits syntaxes, primitive types and flow control statements of C. It also adds syntax for defining classes and methods and runs on dynamic runtime [22].

The choice of using Objective-C in favor of the newer iOS language Swift was made since Objective-C has a more documented use and more available tutorials as of now.

### *Xcode*

Xcode is an integrated development environment (IDE) and a tool set for building Macintosh programs and applications for OS X and iOS. It includes a built-in Interface builder for building Graphical User Interfaces (GUIs) [23].

### 2.3.3. JSON

The data-interchange format JavaScript Object Notation (JSON) is a language that is easy to write, read and also easy for computers to parse and generate. The text format is language independent, but it can be associated with languages in the C-family, C, C#, C++, etc. [24]. The four primitives strings, booleans, nulls and numbers can be represented by JSON. Objects and arrays, which originate from JavaScript, are used to represent structured types. An object is an unordered collection of name/value pairs in which the name is a string and the value can be either type. A pair can be a record, struct, dictionary, keyed list, hash table or associative array. Whereas arrays are ordered lists of values [25].

Strings are sequences of Unicode characters. The numbers are presented in base 10 using decimal digits. For negative numbers the integer is initiated with a minus sign. Leading zeros

are however not allowed. Exponents and fraction parts may be used. An object consists of curly brackets that enwrap the names of the objects and their values. After a name comes a colon, used to separate, followed by the value. At the end of each value a comma is used to distinguish between different name/value pairs. All names within an object must be unique. An array is built with two square brackets in which elements can be represented. The values in an array do not have to be of the same type. The following example shows an object with an array containing three additional objects [26].

```
    {"employees":[
      {
        "firstName": "John",
        "lastName": "Doe"
      },
      {
        "firstName": "Anna",
        "lastName": "Smith"
      },
      {
        "firstName": "Peter",
        "lastName": "Jones"
      }
    ]}
```

Figure 2.4 JSON object with an array

#### 2.3.4. REST API

API stands for Application Programming Interfaces and REST stands for Representational State Transfer and is a maintainable software architecture style that is a simpler alternative to Simple Object Access protocol (SOAP). REST is a protocol for exchanging information in web services in computer networks. It is made up by guidelines and constraints for creating and designing components for scalable web services (usually communicating over HTTP). A client uses a REST API, the server has published, to communicate with the server. This API is open so that other clients or systems can connect to the application [27].



#### 2.3.5. Git

Git is a distributed revision control system, which means that a central repository is missing. Anyone can create their own repository copy that can handle both small and large open source projects. In addition, it also has software configuration management (SCM) tools that support local branching, full-fledged repository with complete history, full version-tracking capabilities, staging areas and multiple non-linear workflows. Git originally

uses a command-line interface (CLI) but since it has become the most commonly used version control system for software development, multiple Graphical user interfaces (GUI) applications are now available for use as well [28].

### ***GitHub***

GitHub is an open source sharing collaboration site and web hosting service for both private and open software projects using Git. GitHub supports online code reviews, overviews and code management. GitHub is free for open source use and offers a paid subscription for private use. It is since may 2011 the most popular web hosting service for open source projects [29].

## 3. Sprint 1: “Analysis and design phase”



The purpose of the first sprint was to collect information about how the current solution for borrowing e-books at Malmö City Library works. User needs were charted to find out how the number of borrowers of e-books can be increased. Current reading applications that are used today were analysed. Their advantages and drawbacks were studied and noted. This information formed a feature list with desired functionalities and the first prototypes were created.

### 3.1. Current solution

The current solution was studied by carrying out the loan procedure and downloading different applications that supports DRM protected e-books and comparing these to each other.

The first element needed for being able to borrow and download a book, is a library card. This is accomplished by visiting the Malmö City Library and applying for a card. When a library card and a PIN code are acquired, a book can be borrowed the same day by visiting the home page of Malmö City Library [30]. There, a book is chosen, the pin code is entered and the book is then (if all the loan criteria are met) downloaded to the user’s smartphone or computer.

In addition to a library card an Adobe ID is needed to be created before the user can read DRM protected e-books. After this a DRM supported application is downloaded and logged into using the Adobe ID. The recommended application from Malmö City Library, that supports DRM protected e-books, is Bluefire Reader. This application was therefore studied and compared with other reading applications on the market. To read more about this comparison go to paragraph 3.3 in this report.

### 3.2. Survey

One of the most important question formulations of the Master thesis was how to attract more people into borrowing and reading e-books from a public library. To collect and organize answers for this, a questionnaire was formed and published on Facebook, Twitter, book forums and blogs. The questionnaire was composed into two initial questions about the participants’ previous borrowing experiences (if they had borrowed and read e-books from a library before or not). Depending on which answer the participants chose they were forwarded to another set of questions. Those who answered that they had not borrowed or read any e-books yet had to explain why this was. Whilst the rest received six follow up questions that looked into their experience with the lending procedure. These questions concerned their technological knowledge; earlier borrowing experience; what would make them borrow more often; and which application they used for reading.

The questions were either combined with a couple of optional answers or constructed like a statement followed by a four-point scale in which the participant needed to select its level of agreement. The reason for using a four-point scale is that the participants are forced to choose a number that leans to one side or the other. In a five-point scale the middle answer can easily be chosen, but it does not give much information about the participants true thoughts to the interviewers.

Since a few of the answers indicated that some of the questions were a bit poorly formulated an updated version was made. The first version and the second updated version of this questionnaire can be found in Appendix A.1.

### 3.2.1. Results from the survey

#### Har du någon gång lånat och läst en e-bok från ett bibliotek?

Svarade: 57 Hoppade över: 0

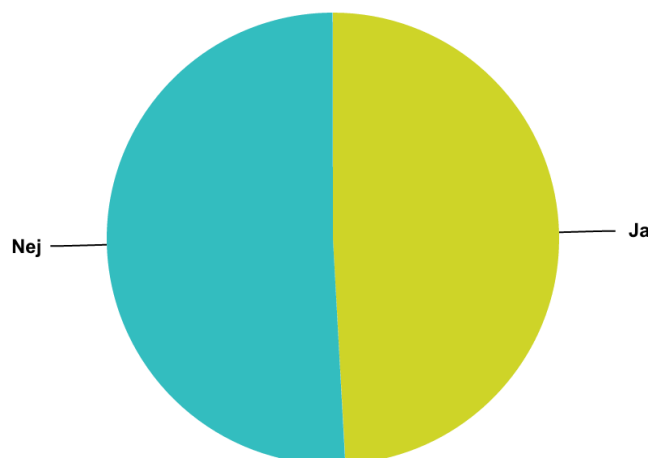


Figure 3.1 Version 1: "Have you ever borrowed and read an e-book from a library?". "No" in blue to the left, "Yes" in yellow to the right.

In the first version of the survey 57 people participated, almost 50% of the persons answering had borrowed e-books from a library before (see figure 3.1). A common reason for not borrowing e-books was that some did not even know that it was possible (see figure 3.2). The second most common reason seems to be that people do not like reading e-books and that they prefer traditional books. The problem mentioned above occurred when the most common reason was studied, almost as much as 60 % chose the option "other" (figure 3.2). This answer did not give much information for the statistics of the study. Therefore, in the second version of the survey more reasons were given.

### Vad är anledningen till att du inte har testat?

Svarade: 28 Hoppade över: 29

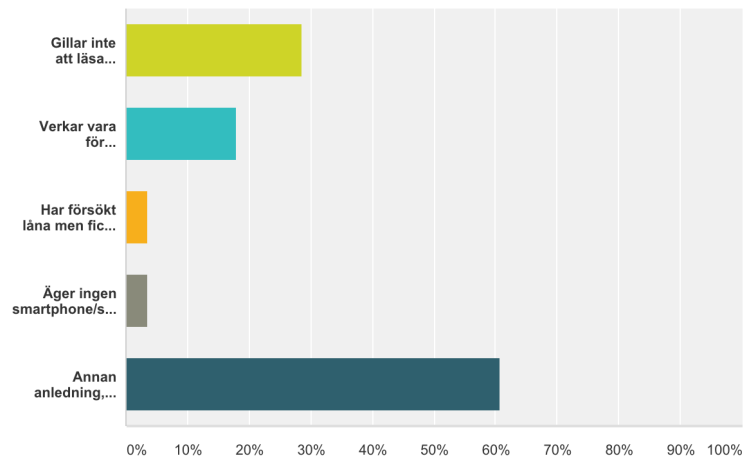


Figure 3.2 Survey version 1: “Why have you not borrowed e-books?”. Options from top to bottom read: “Do not like to read e-books”, “Seems to be too complicated”, “I have tried but I did not get it to work”, “I do not own a smartphone/tablet or other device to read it on.”, “Other..”.

Most of the participants in the first survey were students of engineering answering on Facebook thus the technical knowledge was higher than average. Therefore, the amount of participants finding the borrowing procedure easy was very high (see figure 3.3). The second version (with 17 participants), which was aimed for people with less technical knowledge, resulted in more people finding the lending procedure difficult.

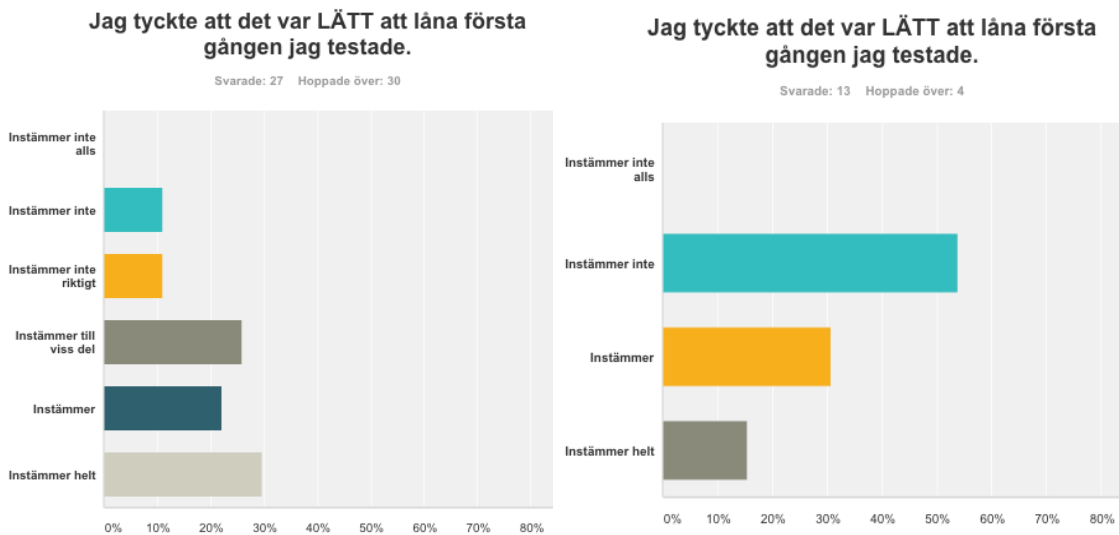


Figure 3.3 “I thought it was EASY to borrow the first time I tried.”. Survey 1 to the left versus survey 2 to the right. Options in scale “Strongly disagree” (top) to Strongly agree (bottom).

Another question that got very different results after updating the survey was the question concerning the issue with using two separate applications for lending and reading. The first version of the question was “I like having a separate app for reading the book that I have borrowed” and the second version was “I would love to have an app that supports both reading and borrowing procedure in the same application”, both had a positive tone that leads user to choose the more positive answers (see figure 3.4). In the second versions all the participants wanted an application with both functionalities, and even though a big percentage answered in the first study that they liked the fact that there was a separate application a lot of users still did not.

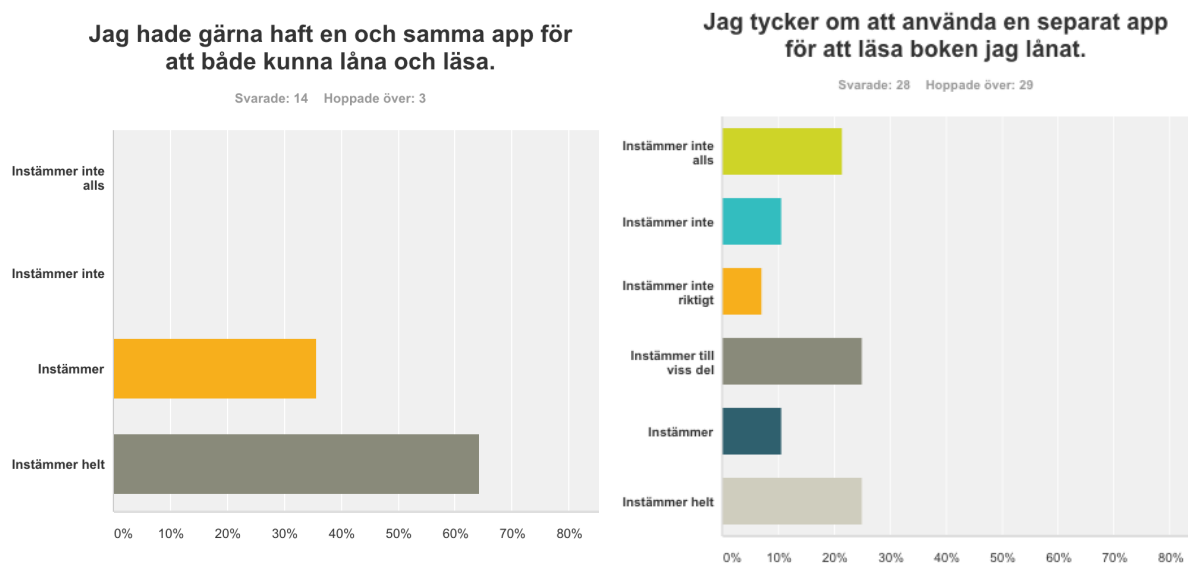


Figure 3.4 survey 1 “I like having a separate app for reading the book that I have borrowed” to the left vs survey 2 “I would love to have an app that supports both reading and borrowing procedure in the same application” to the right. Options in scale “Strongly disagree” (top) to Strongly agree (bottom).

**More statistics:**

- Almost **65 %** of the people answering said that they would borrow more often if there was a bigger and better selection of e-books available. **14 %** if it was easier to borrow, almost **29 %** if the reading experience was better and almost **29 %** if it was easier to find books to read.
- Almost **70 %** said that they use Bluefire reading for their borrowed e-books, which is not surprising since this is the application recommended by the City Library.



### 3.3. Reviewing existing digital media browsers

To create an application that offers appealing design, inspiration was taken from existing mobile applications and media services. Finding an application equivalent to the one developed in this project was not easy. It is possible that a solution like this is available in other countries, but due to limited linguistics none was found. The applications evaluated were instead e-book readers and media services used by Swedes.

The applications were compared to find flawed features, features that were missing or design choices that did not fit the requirements of usability. These parts were documented and avoided in the design of this application. Below are evaluations of some of the media services on the market today.



#### 3.3.1. The demo application of DL Reader

The DL Reader application that is available at the App Store is a demo application used as a downloadable demonstration of the basic functionality that the unbranded reader supports. The application code can then be purchased from Datalogics for use in implementing an own customized reading application (more about this in 1.3.2).

The reader is one of only a few applications that can read Digital Rights Management (DRM) protected e-book files. The application supports features such as bookmarking pages; highlighting text; text search; multiple language support; support for both Adobe ID and Vendor ID accounts; and compatibility with Adobe-based bookstores, libraries and more [31].

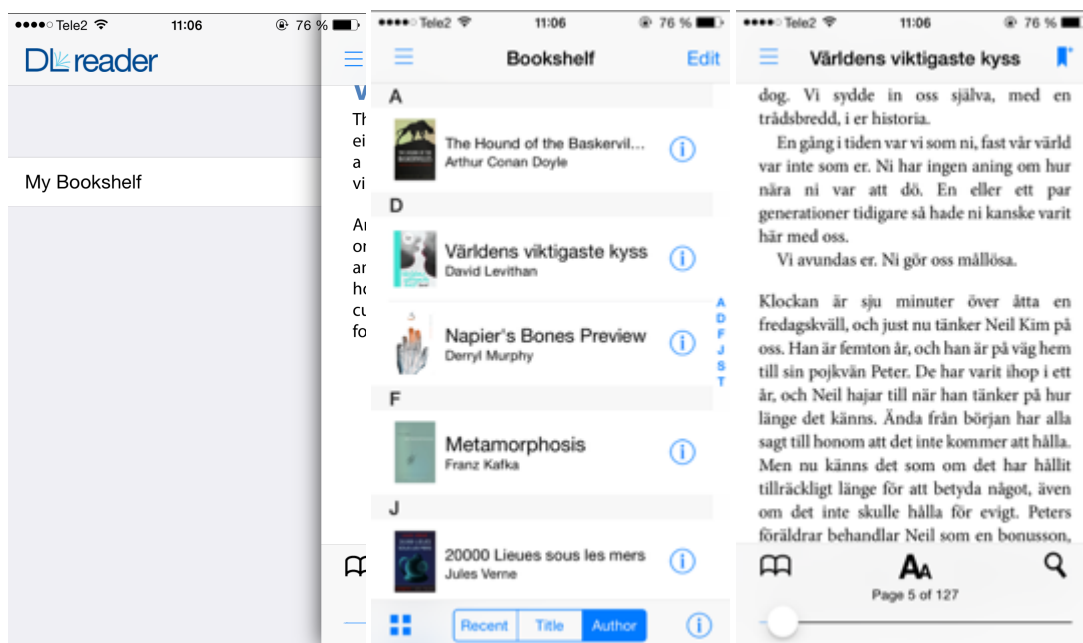


Figure 3.5 DL Reader App.

The reader functionality has the basic settings like changing font size, see chapters, highlighting, bookmarks and a simple text search (see figure 3.5). These are all stated as a must according to the article mentioned in paragraph 1.3.4. However, a lot more functionality is desirable for a long time use. Another functionality missing and a big concern for future usage of borrowed books is that the application does not have any indication or information showing the amount of days left on the loan.



### 3.3.2. Bluefire Reader

Bluefire Reader is another application that can read DRM protected e-books borrowed from libraries. Its features include highlighting; bookmarking pages; looking up the definition of a word; share excerpts (via email, Facebook and Twitter); and organize books into collections etc.

Bluefire Reader is the application that Malmö City Library is recommending their iOS users to download and use for reading their books. It was therefore essential to investigate this application. [32]

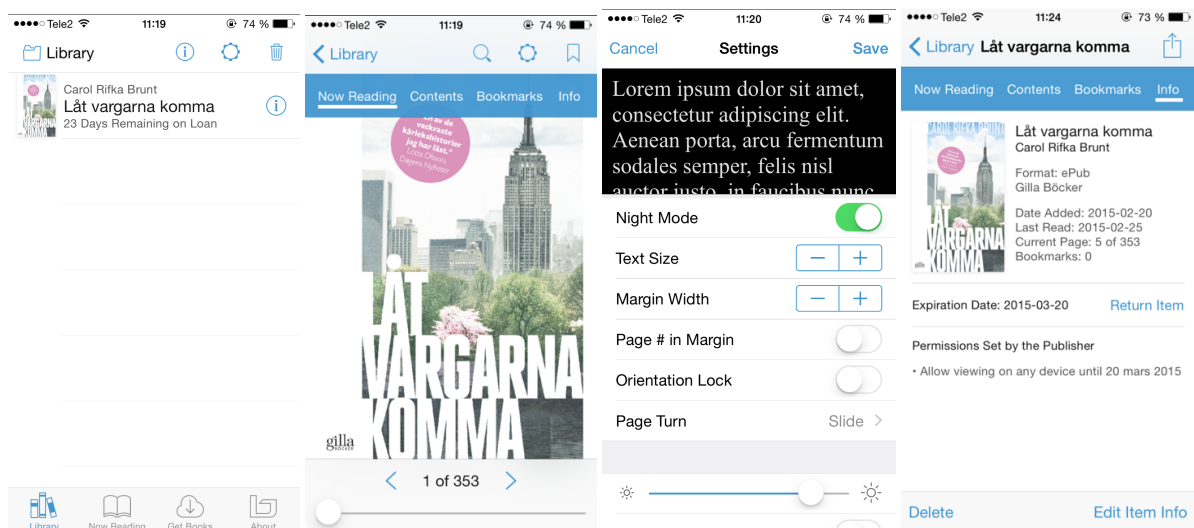


Figure 3.6 Bluefire Reader App

After testing the application a few concerns came up. The settings in the reading mode were not always satisfying. It did have an extensive list of settings available, which was good, but the font and its color do not always correspond to the chosen settings and when applying night mode the background became gray instead of black for some of the tested library books.

Another big concern, since this is the application that Swedish users are recommended to use, is the fact that the language in the application is English and the settings to change language does not seem to exist.

However, the application did have a clear indication about the number of days remaining on the loan. It also shows the expiration date of the borrowed book in the information section (figure 3.6). Another good feature is the synchronization between devices, a user can start reading on their iPad and then continue from the same spot on their iPhone.



### 3.3.3. Adlibris Mondo

Adlibris Mondo is both an online store and an application where you can buy, collect and read e-books, audiobooks etc. [33].

The difference is that the purchase and the reading experience is separated, with the store only available on the web page (figure 3.7) and the reading only in the application (figure 3.8). What instead makes Mondo more interesting is their social features with their review system, where buyers can rate and review books on the web page and share their reading experiences.

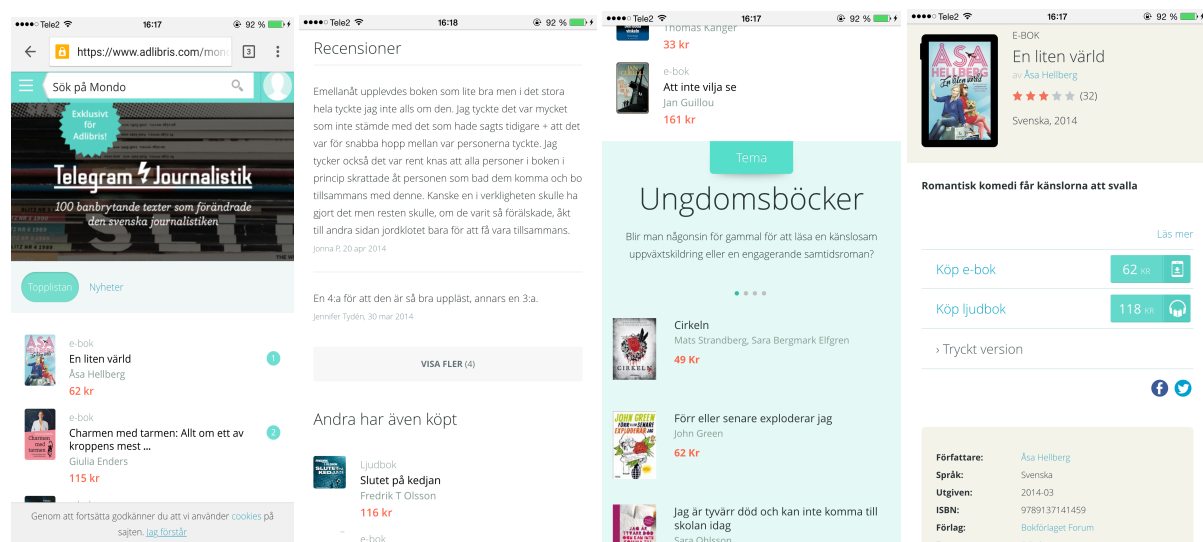


Figure 3.7 Webview: Buying, finding and rating/reviewing books.

The web page is consistent with its design and quite easy to follow. The store also has a nice feature where certain books from a popular genre are displayed together in a group (see figure 3.7) and where the different books of a genre can be explored by sliding the window event scroll left or right to change the chosen genre. This was later used in the view for Malmö City Library in the Mid-Fi prototype (see Appendix A.3).

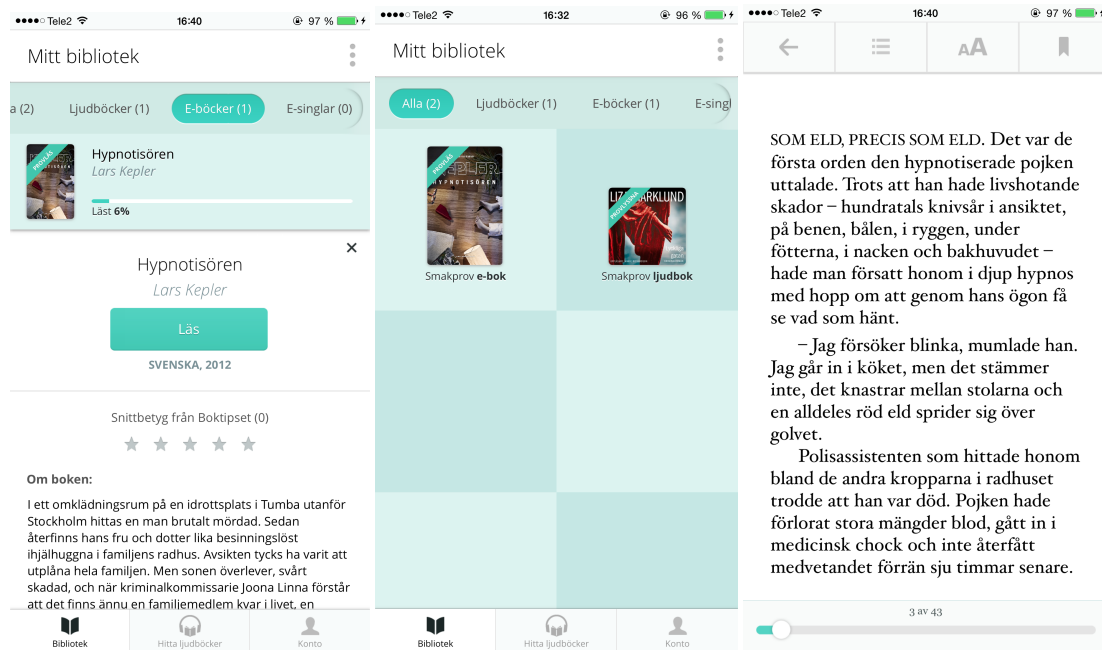


Figure 3.8 Mobile app view: Reading, collecting books.

The reading application features a simple grid or list view with the books the user has bought and a pretty standard reading functionality with settings for screen brightness, font size, type and reading mode. A clear indication of the percentage of how far the reader has read is shown both in the list and the grid mode (see figure 3.8).



### 3.3.4. Kindle

Kindle is an application for reading purchased books from the website of Amazon [34]. The application automatically synchronizes last page the user reads, bookmarks, notes, and highlights across the user's devices. The application also supports accessibility features for blind and visually impaired users with VoiceOver. The reading experience can be customized with settings for brightness, font size, font type, background color and margin size (see figure 3.9). Another nice feature is the "amount of pages read" indication in the list-view for the books.

The interface is quite straight forward and easy to use. The application has a tasteful modern design, and the dark gray background color is easy on the eyes, and makes the covers "pop out" more. In similarity to the DL Reader a button for switching between the grid- or the list-view and a button for sorting modes is located at the bottom of the toolbar. A clear "Sample"-label is given to each book sample.

The design includes a lot of simple and precise interaction choices and was therefore a great inspiration for the Mid-Fi prototype.

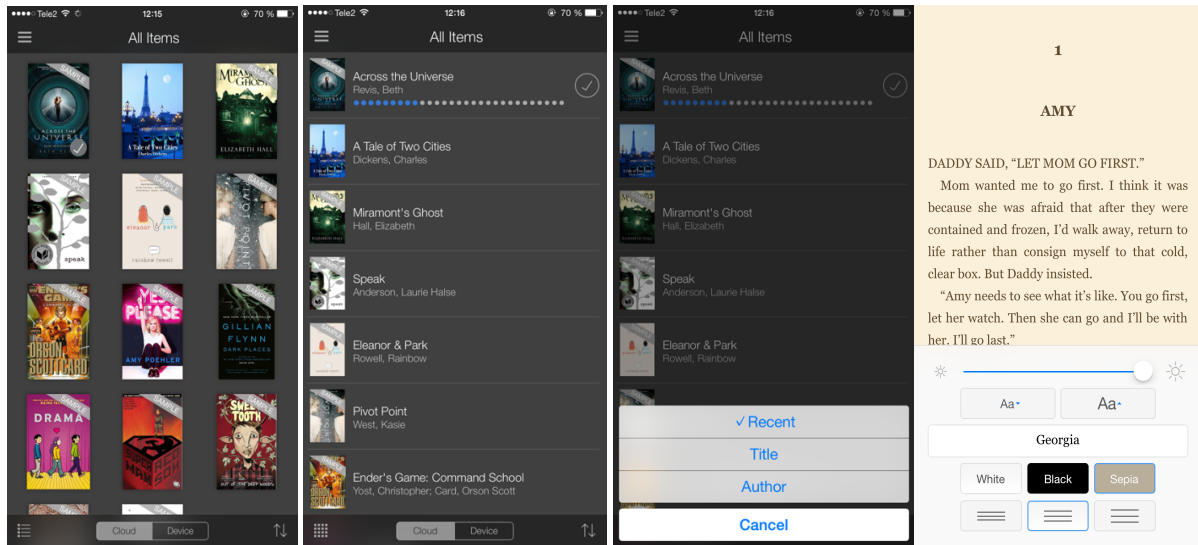
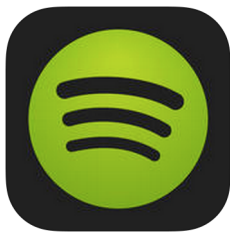


Figure 3.9 Pages from the Kindle App.



### 3.3.5. Netflix and Spotify

Since the music and movie streaming services Spotify and Netflix have been very successful their interface were carefully studied. The characteristics of their interfaces are subtle backgrounds where covers and titles are in focus. Both services promote music and movies into different categories and themes which makes it easier for users to find what they are looking for (see figure 3.11). This was taken in consideration when designing the interface of the library application.



To flexibly switch between different views within an application a navigation menu is required. In addition to that, the interface should not be cluttered. A great way to display extensive information is to let it be disclosed upon a click of a button. The navigation of the prototypes was influenced by the slide menu in Spotify (compare figure 3.10 with 6.3 and 6.6) [35].

A book has an extensive amount of data, which may be significant for some users when deciding over which book to borrow. However, since the library has many books to offer, the detailed information must be displayed in an obvious yet stylish way. On Netflix and on similar sites, information about a movie is revealed in a popover window which glides in over the screen from the right, only covering a part of the screen (see figure 3.12) [36]. This design solution was used in the prototypes (see figure 5.3).

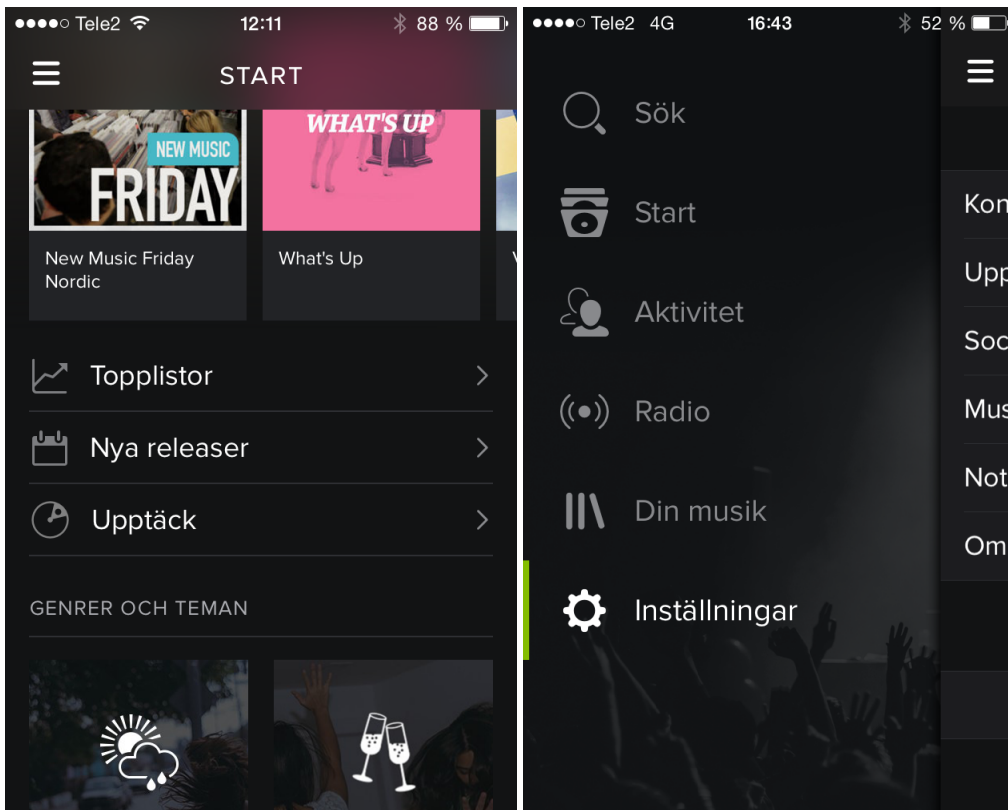


Figure 3.10 Spotify start page and side menu.

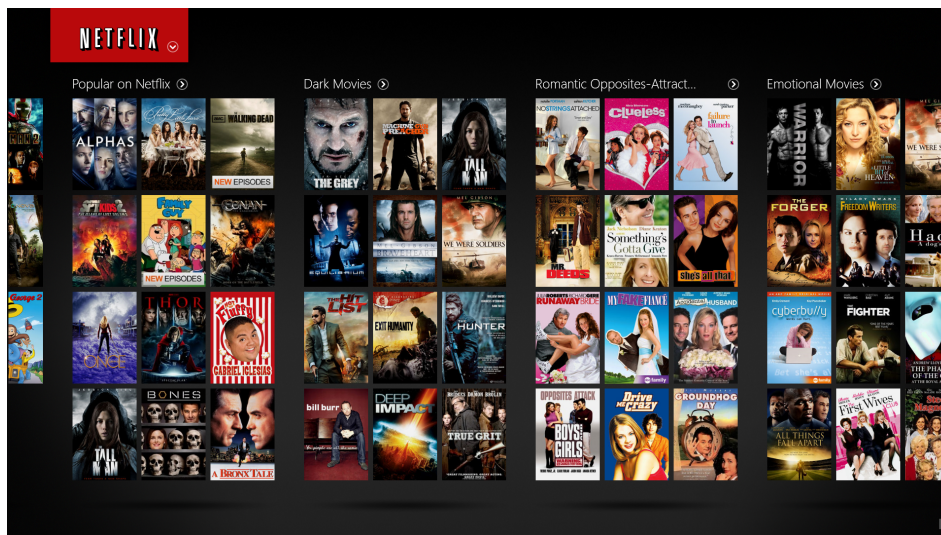


Figure 3.11 Netflix movie categories.

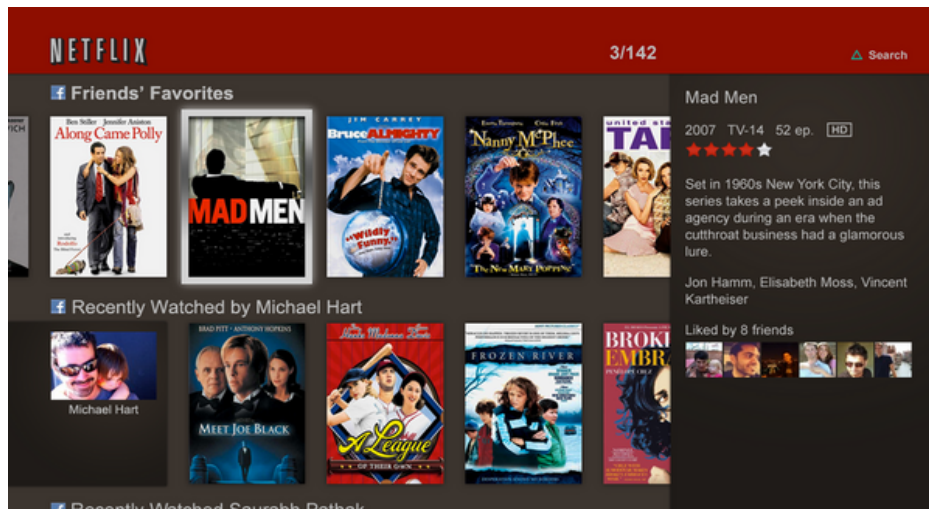


Figure 3.12 Netflix popover window with information.



### 3.3.6. Bokon

Bokon is an online e-book store that has its own application with functionalities for both purchasing and reading e-books. The application is automatically synchronized from the Bokon library [37].

The application uses a design with horizontal scrolling displays (figure 3.13) for the different books on the front page of the store (similar to the design of Netflix and Spotify). The application also uses a top featured spot at the top of the start page with extra space for certain books that they want to give extra attention to. The application of this project used a similar solution for displaying new and popular books and also for showing and giving attention to events held at the library (see figure 3.13 and 5.2).

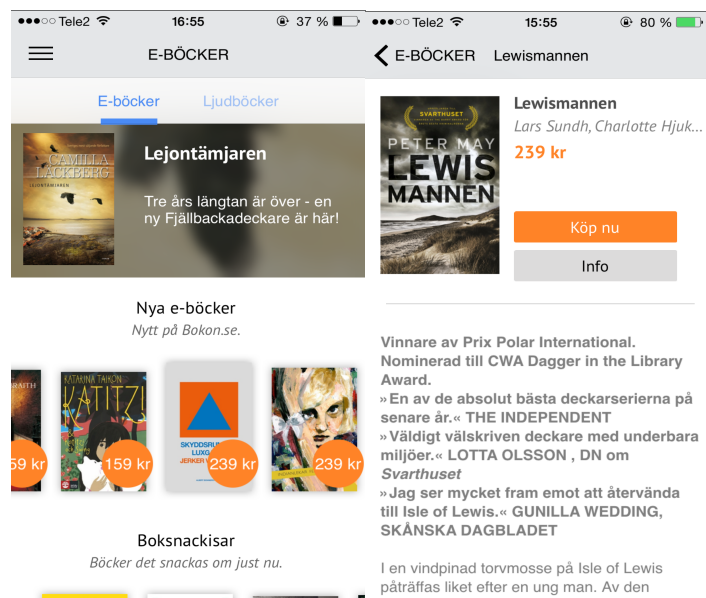


Figure 3.13 Bokon

### 3.4. Feature list

After studying existing applications and media services as well as the answers from the surveys, a feature list was created at the end of week two. It lists the desired functions of a finished product. The features were sorted into appropriate groups and given a degree of priority to make it clearer which functions were most important in this limited project. This process was continued and finished at the beginning of week three, by adding and changing features along the way.

The social features are believed to increase loyal users and make the application more desirable. Unfortunately, they had to be set aside due to the limited project scope. The picture below (figure 3.14) is a selection from the feature list presenting what type of information about a book users should be able to retrieve. Some inspiration was drawn from the survey conducted by Stockholm Public Library (see figure 1.5) which established that a description of the book (81%) and a cover image of the book (52%) were both desired functionalities. The complete feature list can be viewed in Appendix A.2.

Book Information		
See title	The user should be able to see the book's title	1: Must have
See author(s)	The user should be able to see who wrote the book	1: Must have
See cover	The user should be able to see a picture of the cover of the book	3: Would like to have
Read description	The user should be able to read a description of the book's story/plot	1: Must have
See nbr of pages	The user should be able to see how many pages a book has	2: Would love to have
See ISBN	The user should be able to see the book's ISBN number	1: Must have
See publication date	The user should be able to see the date the book was published	2: Would love to have
See translator	The user should be able to see who translated the book, IF the book was translated	2: Would love to have
See original title	The user should be able to see book's original title, IF the book was translated	3: Would like to have
See publisher	The user should be able to see the publisher of the book	1: Must have
See language	The user should be able to see the language the book is written in	1: Must have
See categories	The user should be able to see which categories the book belongs to	1: Must have
See series order	The user should be able to see what order in a series a book has, IF its part of a series	4: If there is time
See series	The user should be able to see the other books in the series that it belongs to, IF it belongs to any	4: If there is time
Read sample	The user should be able to read a small part of the book before deciding to borrow it	4: If there is time
Add to TBR list	The user should be able to add a book to the TBR-list that the user wants to read in the future	3: Would like to have
See borrowing period	The user should be able to see how long a book can be borrowed	3: Would like to have
See similar books	The user should be able to see a list with similar books	4: If there is time

Figure 3.14 Selection from the feature list concerning information about a book.

### 3.5. Prototyping

Based on the feature list and studies of existing applications, ideas of how the interface would work and look started to form. As the survey from Stockholm Public Library (figure 1.4) presented, most e-books were preferred to be read on a tablet (46% on iPad, 17% on Android).



Therefore the focus of this thesis project was designed for iPads throughout the whole process of prototyping. The initial thoughts were sketched on paper, making it easy to present several solutions. Below (figure 3.15 and 3.16) are some pictures of the Lo-Fi prototype procedure.

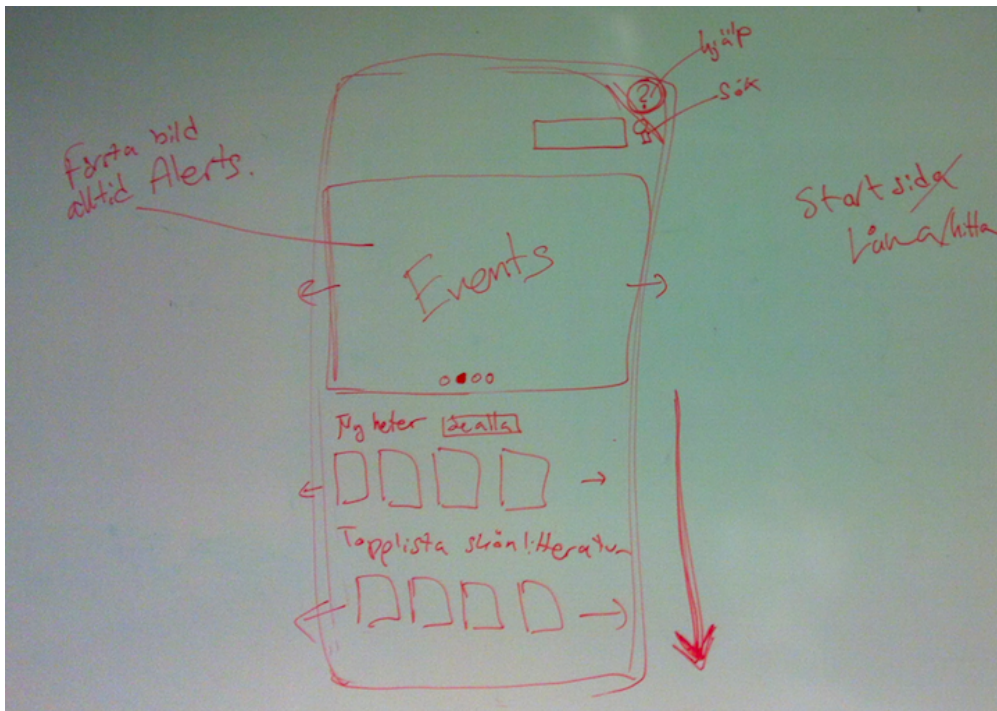


Figure 3.15 LO-Fi sketch

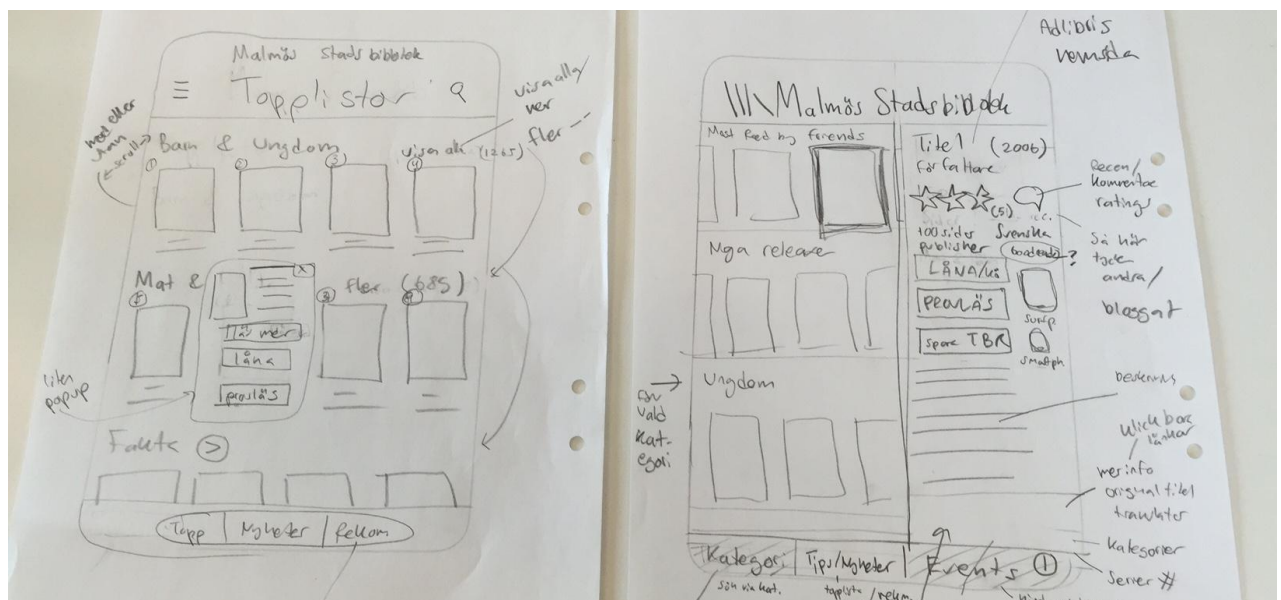


Figure 3.16 LO-Fi sketches

A more specific Mid-Fi prototype was then developed with the mockup and wireframing tool Balsamiq (see figure 3.17). The program contains many complete elements that can be dragged and dropped onto a canvas. The elements can then be edited in desired styles. Both prototypes were made vertical, meaning that they present many details and fewer functions (see paragraph 2.2.3). The Mid-Fi prototype was later used as a template when implementing the Hi-Fi prototype in Xcode. For the full documentation of the Mid-Fi prototype look in Appendix A.3.

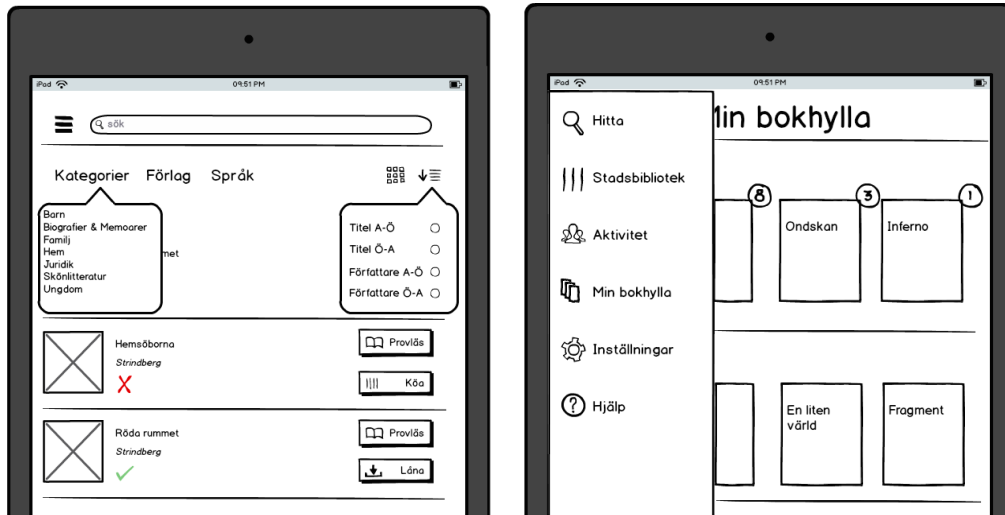


Figure 3.17 Mid-Fi prototype

## 4. Sprint 2: “ Setup and base design phase”



In sprint 2 the implementation of the application began. Knowledge about iOS development was obtained and the foundation of the application, based on DL Reader, was installed and stored using GitHub. Transformations of the interface according to the Mid-Fi design begun.

### 4.1. iOS development studies

Since the Hi-Fi prototype was developed for iPads the primary programming language used for writing software for iOS, Objective-C, was studied and learned before the implementation started. The Objective-C training was conducted by following an easy step-by-step tutorial for building a calculator in the book “iPad and iPhone App Development” [38], viewing beginner tutorials for Objective-C on YouTube [39] and following other various tutorials online by for example RyPress [40]. The usage of Xcode was learned in correlation with the studies of Objective-C and its tutorial lessons since all the programming was implemented in Xcode.

### 4.2. Installation of DL Reader code

The core foundation of the Hi-Fi prototype was based on the DL Reader code provided from Datalogics. To install the code an extensive guide had to be followed. This procedure was fairly complex. Once completed, the code installed includes support for reading DRM protected books with a simple reader, a web view for their Book Club page, a grid/list view for the books downloaded to the application, and support for the Adobe activation models, Adobe ID and Vendor IDs. The code provided was over 400 000 lines long. Much time was therefore dedicated to reviewing the code and trying to understand how it worked. A document was kept to note which classes and methods controlled different parts of the interface, e.g. the side menu. This project used the private GitHub repository of Jayway to store the code and its modifications. Source Control Handling in Xcode was used for synchronizing.

### 4.3. Redesigning the interface of DL Reader

Since the study of iOS and installation of DL Reader took longer time than expected the redesign of the interface had to be shortened in this sprint. The interface of DL Reader was slowly transformed based on the Mid-Fi prototype. It took time to detect and change the right objects because of the large amount of code in the project. The main focus was to change headings, buttons and colors. To determine which colors were best suitable was difficult and therefore not finalized until the last sprint.

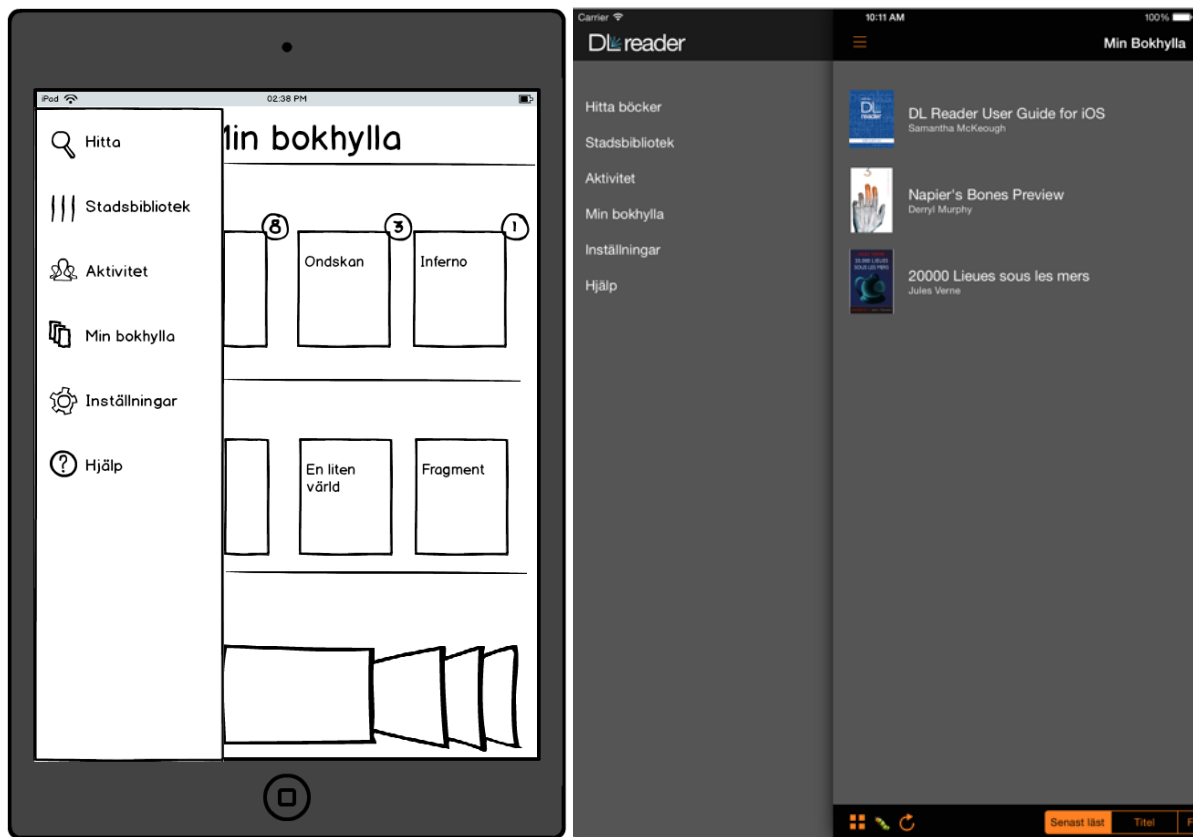


Figure 4.1 Side menu. The Mid-Fi prototype to the left and Hi-Fi prototype to the right.

More menu options and pages were added to the side menu (see figure 4.1 above). Some changes were made to the “My Bookshelf” page, like updating the information that was shown in the popover revealed when the info button (see figure 4.2 below) was pressed. This information was changed in an HTML document which the popover was initially programmed to display. The original “Book Club” page from the DL Reader code, which was linked to a Datalogics book club page, was removed and a new webview with the webpage of Elib (see paragraph 2.3.1) was created and displayed at the end of the sprint.

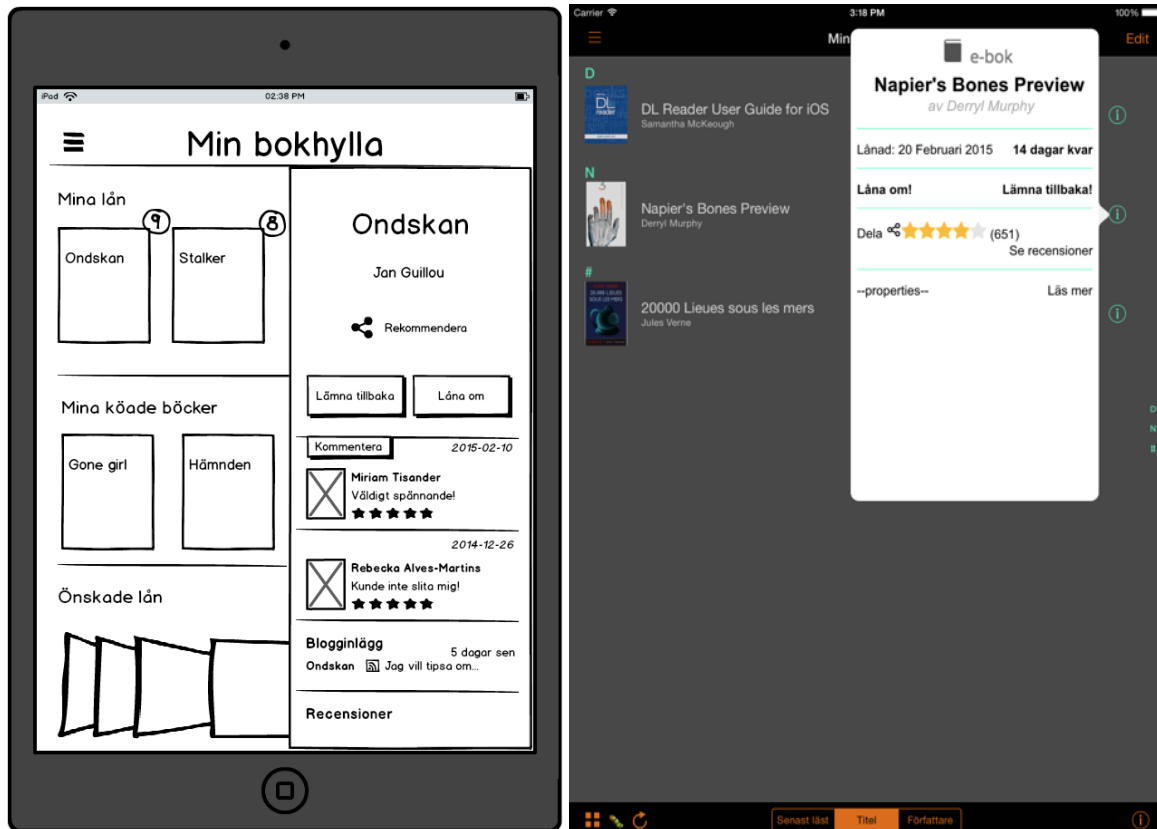


Figure 4.2 “My Bookshelf” popover info. The Mid-Fi prototype to the left and Hi-Fi prototype to the right.

A view for searching for books to borrow was also created at the end of this sprint (see figure 4.3 below). The view contained a search bar, but since there were limited data to interact with, searching for books was not yet possible. The buttons for switching to grid-view, sorting, reading a sample, borrowing and queuing also lacked interaction and functionality. Three filters presenting options for categories, publishers and languages were displayed with popovers containing an extensive list of the options retrieved from a hard coded property list saved directly in Xcode. A table view was presented containing a static table cell with information about a book (like cover image, title and author), which was also fetched from the property list and was implemented to visualize how a search result would be displayed in a later stage of the implementation.

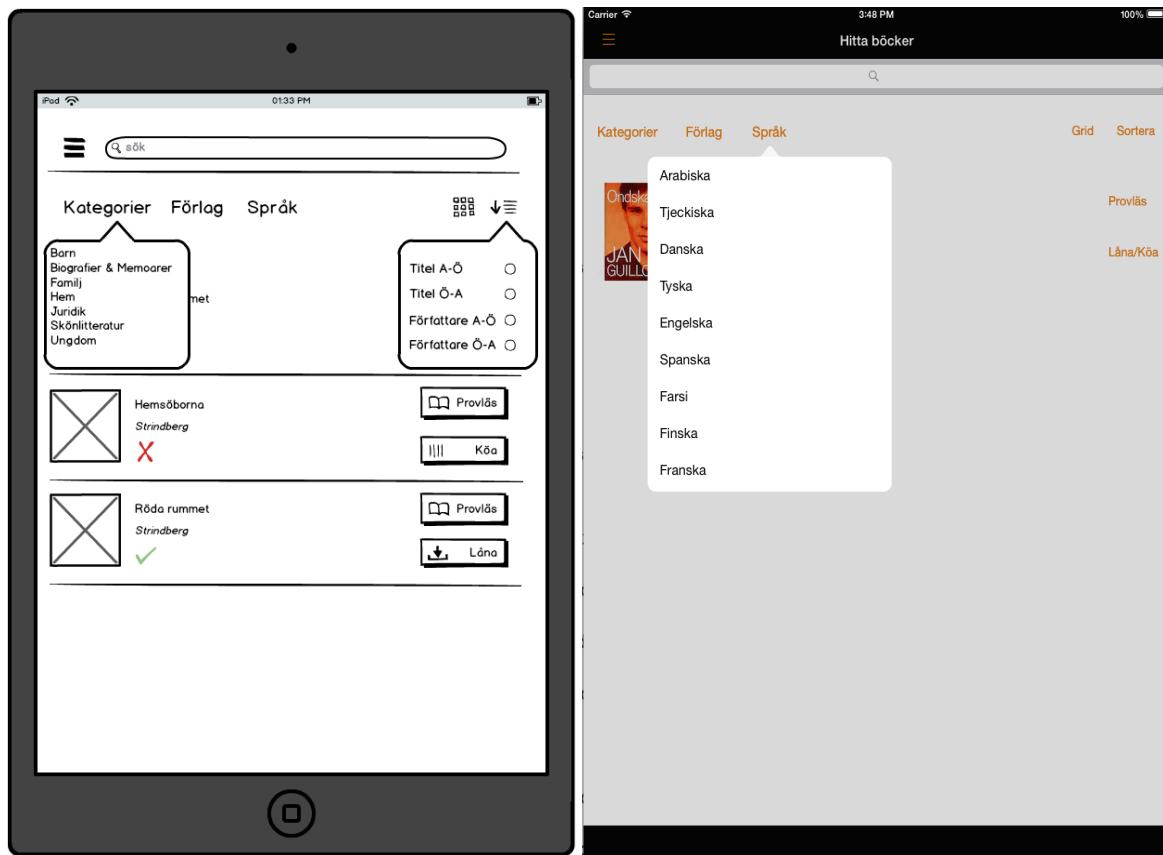


Figure 4.3 Search view with filters. The Mid-Fi prototype to the left and Hi-Fi prototype to the right.

#### 4.4. Sprint 2 conclusions

At the end of the second sprint a demo presentation was held where the current solution was presented for the team at Jayway. Valuable feedback was obtained and after discussions it was decided to remove the filters in the “Search view” (see figure 4.3). All forms of searches are already supported in the search field which makes the filters redundant. The team also recommended to keep the interface simple. Many advanced features will satisfy experienced users, but they are also likely to daunt users with less technical knowledge. These features could instead be more hidden, e.g. with links named “see reviews” or “read more”.

## 5. Sprint 3: “Synchronization phase”



The third part of the Master thesis was dedicated to connecting the application to Elib (see paragraph 2.3.1) which was accomplished by using third party frameworks. After receiving data from Elib to interact with, additional features could be implemented. To establish how users interpreted the first version of the interface user studies were conducted at the end of this sprint.

### 5.1. Synchronization with Elib

To be able to display actual data and actual books from the library, the application needed to be connected to Elib through a REST API (see paragraph 2.3.4). Since this development only concerns a prototype the data was retrieved from a test account at Elib in contrast to connecting to the actual database. The data from Elib was reviewed as JSON code (see paragraph 2.3.3). To simplify the REST communication third party frameworks was used, below are four frameworks which were examined and compared during this Master thesis.

#### 5.1.1. RestKit vs JSONModel

RestKit is an Objective-C framework that is a combination of HTTP request/response API and an object mapping system [41]. RestKit was the first framework used in the project studies in sprint 2 when learning how to handle REST communication with the server of Elib [42].

RestKit promised to make the handling with JSON and REST simple and also promised to reduce the lines of code required. Unfortunately, it did not make coding simpler. The usage was instead a bit complicated and no apparent solution for using RestKit with unnamed objects in an array (see figure 5.1), could be found. A few days were given to solve this issue without result. Therefore this framework was chosen not to be used and instead JSONModel was chosen. The problems mentioned and the rest of the RestKit implementation were solved in a matter of hours after switching to JSONModel instead.

```
"Authors": [
  {
    "FullName": "Ika Johannesson",
    "FirstName": "Ika",
    "LastName": "Johannesson"
  },
  {
    "FullName": "Jon Jefferson Klingberg",
    "FirstName": "Jon",
    "LastName": "Jefferson Klingberg"
  }
],
```

Figure 5.1 JSON array with objects without a name.

JSONModel is also a data modelling framework used for iOS or OSX applications that supports built-in networking. Using this framework made the creation of JSON data model classes much easier and reduced the amount of code needed to be written. JSONModel automatically syncs the user's model classes and the structure of the user's JSON input validation; type casting; class cascading and more [43].

### 5.1.2. SDWebImage vs AsyncImageView

Since the application will load an extensive amount of images from the server of Elib, the images needed to be loaded and displayed asynchronously (independently of the main flow), so that the downloading would not freeze the interface. This was achieved by using a third party framework for asynchronous image loading.

The project first used the framework SDWebImage [44] but after encountering some problems with loading the framework correctly into Xcode and to the existing DL Reader code a slightly more user friendly version, AsyncImageView, was used [45].

## 5.2. Expanding the features

In Sprint 3 a page for the Malmö City Library was created (see figure 5.2). This page contained three parts; a horizontal scrollable view presenting informative images about events related to the library; a list of newly added books to the e-book selection of the libraries; and a top list of the most popular titles that people have borrowed.

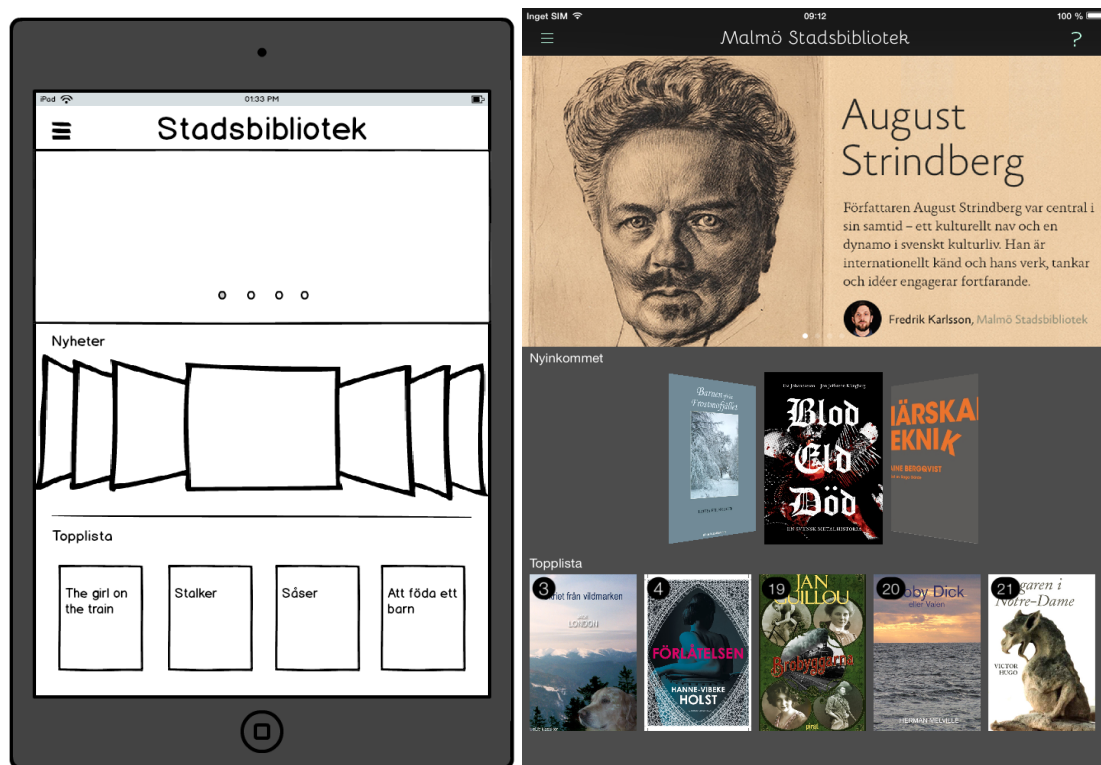


Figure 5.2 The Malmö City Library page.  
The Mid-Fi prototype to the left and Hi-Fi prototype to the right.



A side view window with information and details about a book was implemented (see figure 5.3 below) according to the Mid-Fi design. This information is shown when a book is tapped in the Malmö City Library page and in the “Search for books” page. A button with the purpose of reading sample books was positioned in this view. A temporary solution for this feature was implemented (to read more about this go to paragraph 6.1).

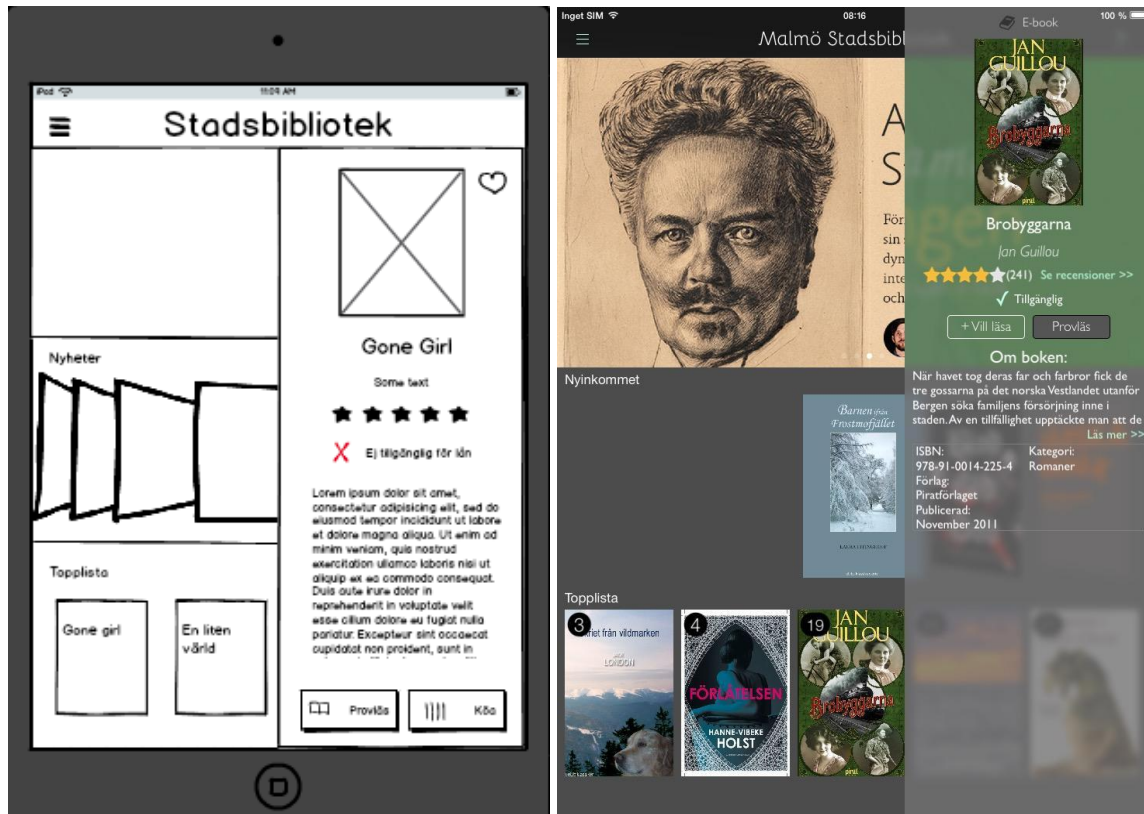


Figure 5.3 The side view of the “Malmö City Library” page.

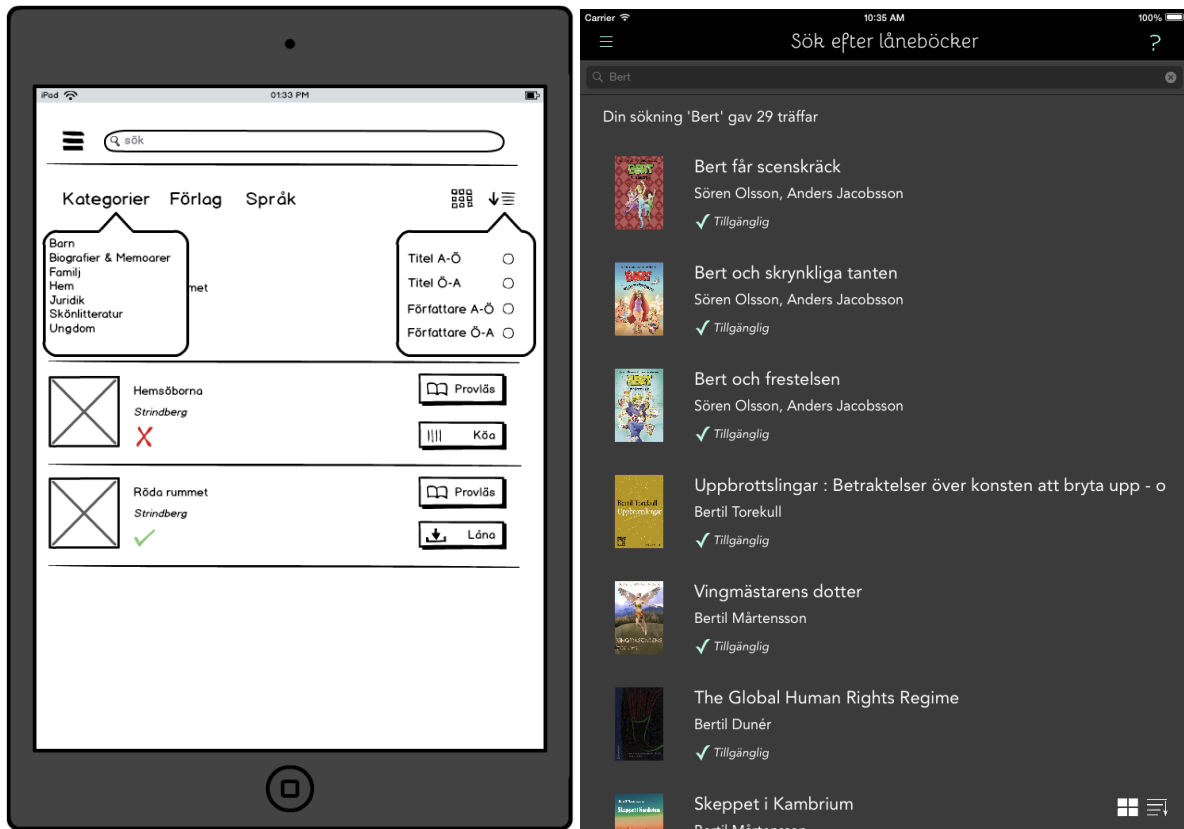


Figure 5.4 The search view: the Mid-Fi prototype to the left and Hi-Fi prototype to the right.

The “Search for books” view from sprint 2 was further developed (see figure 5.4). When typing a word in the searchfield the result was fetched from Elib. The text below the search bar presented the number of hits. In each cell of the table a book cover, title and author were shown. To give users feedback if the book was available for borrowing a check mark and the short text “Available” was used. The users were able to scroll through the table, but the text with the number of hits and the two buttons in the bottom right corner were permanent. When making a long press on the button with an arrow a popover with sorting options (title, author, etc.) was shown. As the user picks different options in the list the search result in the table was sorted accordingly. A radio button in selected state gave the user feedback on which sorting was currently used. By simply tapping the same button the direction of the sorting was changed (e.g. from A-Z to Z-A) and the icon of the button was rotated indicating that a change had been made (figure 5.5). When pressing the button for grid-view the listed search result was instead displayed as a collection of book covers (see figure 5.6).



Figure 5.5 The sort button changes.

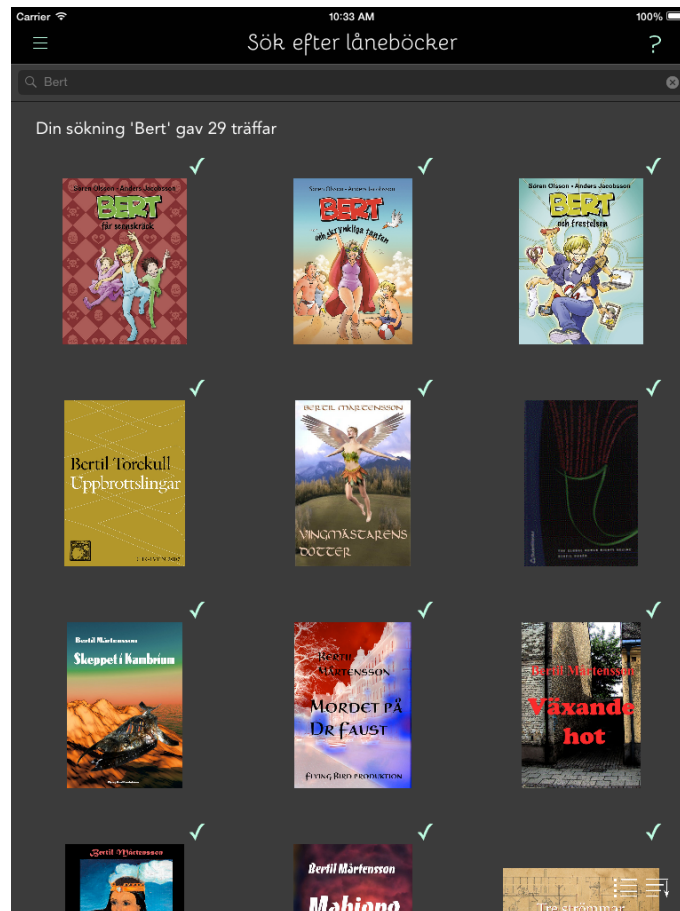


Figure 5.6 Searches in the grid view.

### 5.3. Sprint 3 User study

To evaluate how users interacted with the application a user study was performed. Six students from LTH and two representatives from Malmö City Library were asked to participate. The following scenarios were created and the participants were asked to complete the task with minimum help from the test leaders.

1. "Find the release date of the book which is placed as number four in the top list of Malmö City Library."
2. "You want to borrow a thriller whose title you cannot remember, however you know that the last name of the author starts with the letter "Ö". What do you do?"

The hypothesis was that users would have problems with separating the view for borrowed books and the view for the City Library due to their similar appearance. This was confirmed. It also became obvious that the rest of the interface was not as self-explanatory as anticipated. Most participants got stuck on the same problems. To find the fourth book in the top list of the public library view was not difficult. However, a long press upon the cover of the book was needed to display the side view window containing information about the book. None of the participants realized that they had to press that long.

To find the book in task two was even more difficult. First of all, few understood that they could search for thrillers in the search field and therefore only typed “Ö” whereupon a huge amount of hits was displayed. Due to this problem all participants needed help from the test leaders to be able to continue. After a successful search the participants needed to sort the hits after Author from Ö-A. The sort button contained two events. The popover containing sort choices appeared after performing a long press on the button and a short tap changed the sort direction. These were not natural button events for the participants and the long press action was once again not something the users were familiar with.

The user study was successful since the participants could carry out the scenarios and only needed help from the test leaders when they did not understand that they could search for categories. Even though, the results from the study implicated a few changes in purpose to enhance the user experience. The long press in the view of the Malmö City Library was greatly shortened, the two events of the sort button was reduced to one. Only a single tap is now required to make the popover appear and the choices include both directions “A to Ö” and “Ö to A”. The different kind of searches allowed are clarified with a placeholder text in the search bar (see figure 5.7).



*Figure 5.7 Search bar with an explanatory placeholder text*

Afterwards an updated version was presented at a demonstration for the team at Jayway. It became obvious that the text in the search bar was hard to read when using a device in dark surroundings. The font size was therefore increased and the text color brightened.

## 6. Sprint 4: “Final design phase”



The focus of the last sprint was to complete the design of the interface and a few more features to be able to perform proper user studies. Finalizing design choices were made and applied to the interface. Since an actual lending procedure could not be implemented, due to faults in the DL Reader code provided from Datalogics, a simulated loan was chosen instead. The loan procedure was needed and used during the final user studies.

### 6.1. Simulated lending procedure

The support of an actual loan was not working properly in the DL Reader code. For some reason a link to an .epub or .acsm file could not be opened directly in the application. Therefore, to be able to test how the process of borrowing will feel like, a simulated version was implemented.

When a “Borrow” button was pressed a pop up window appeared with a request for the user’s PIN code (see figure 6.1). This password was never tested against an actual account in the simulated version. Instead a link to a book sample file was displayed in Safari, where the choice of opening the file in the application could be made (see figure 6.2). In that way the file was automatically added to the “My Bookshelf” page. This solution was made to save a lot of time and for the benefit of being able to perform user tests of the borrowing process without having to deal with the highly complex protection procedure of Adobe ID.

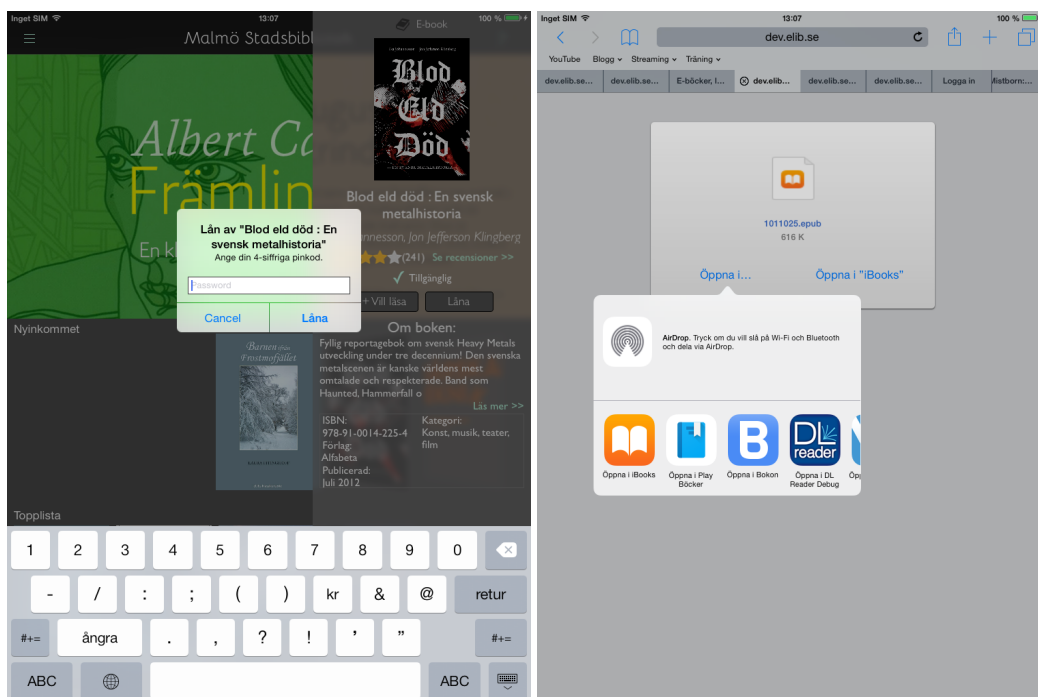


Figure 6.1 Simulated loaning procedure. Figure 6.2 The links to reading samples in Safari.

## 6.2. Finalizing design choices

During the last sprint the final design of the interface was implemented. All views were reviewed and updated to further follow the Mid-Fi prototype. Listed below are the finalized design choices and the reason for choosing them.

There are several aspects to keep in mind when designing iOS-applications. Most importantly the different views should be consistent. It will confuse users or make them question the purpose of change if the appearance or position of elements differs between views. The documentation, Human Interface Guidelines, has been set up by Apple in order to help designers with some of these aspects and has been considered during the design process [46].

### 6.2.1. Navigation and Positions

The navigation of an application should be self-evident, entailing that users should always know where they are and how they can get to another view. The Navigation bar is a standard element in iOS-applications (excluding games) and is positioned at the top of the screen. It is a great asset for presenting the title of the current position and how to navigate to other views. This validates the positions of the side menu and the title of the view at the top of the screen [47]. The navigation bar of this application contains the icon for the side menu in the left corner, the title of the view in the middle and the icon for the help menu in the right corner (see figure 6.3 below).



Figure 6.3 Navigation bar

A rule that should be exercised in interfaces is to place important elements in the top left corner of the screen. The less important elements should be positioned closer to the lower right corner. This basically means that the importance decreases along the diagonal from top left to bottom right [48]. Since the side menu will be frequently used, every time the user wants to switch between views, it has the optimal position of the top left corner making it most accessible (see figure 6.3 above).

To even more enhance important content within an application negative space is used [46]. It makes the interface focused and composed. The negative space also makes the interface more efficient since elements are easier for the user to find. To avoid a cluttered interface, plenty of negative space is used in this application (see figure 6.4 for an example).

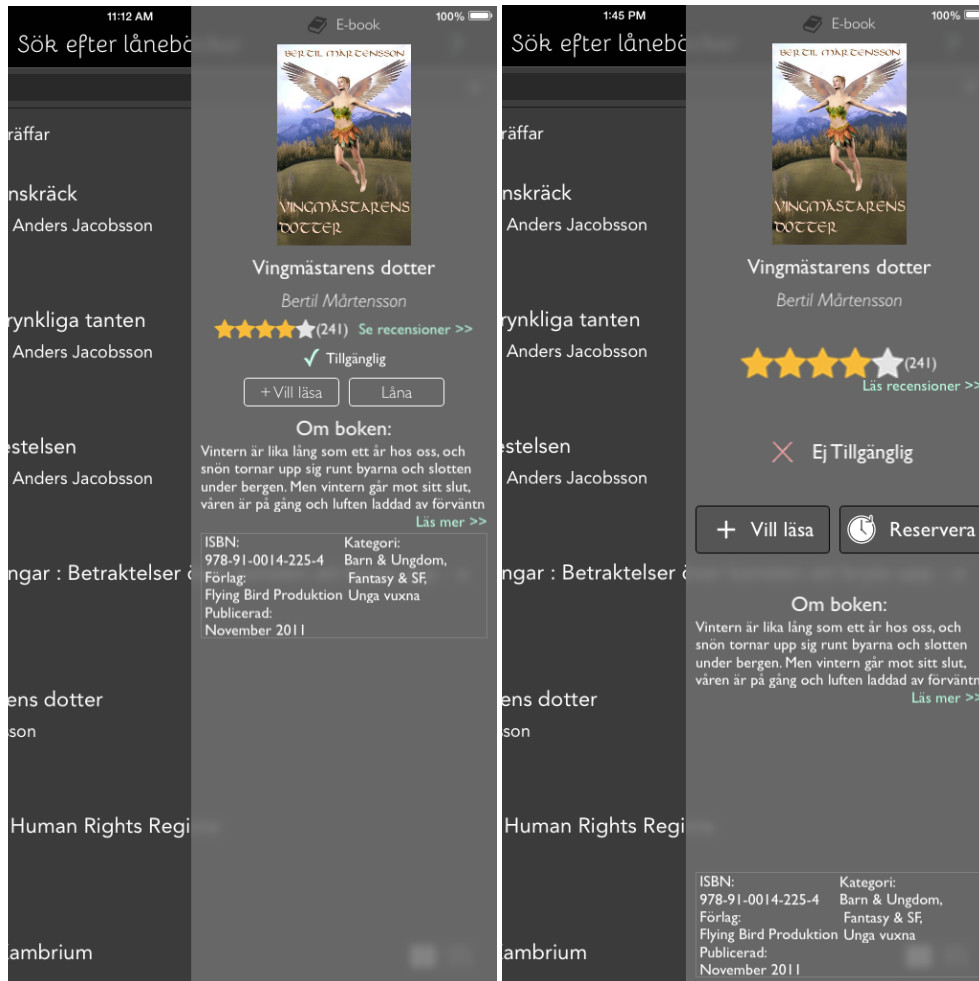


Figure 6.4 Book info window: Old version to the left vs. the new version with more negative space to the right.

## 6.2.2. Constraints

The focus was kept on making the interface as simple as possible with fewer elements to interact with [49]. Since the application is aimed at such a large target group it is hard to create an extensive design, with many features, that will appeal to all.

An apparent constraint can be found when clicking on a book in the view for the "Malmö City Library" page. Depending on whether a book is available for borrowing or not the right button changes between "Borrow" and "Reserve" (see figure 6.5). Choices that can not be made at the

moment are consequently eliminated, making it obvious to the user which options they can select.

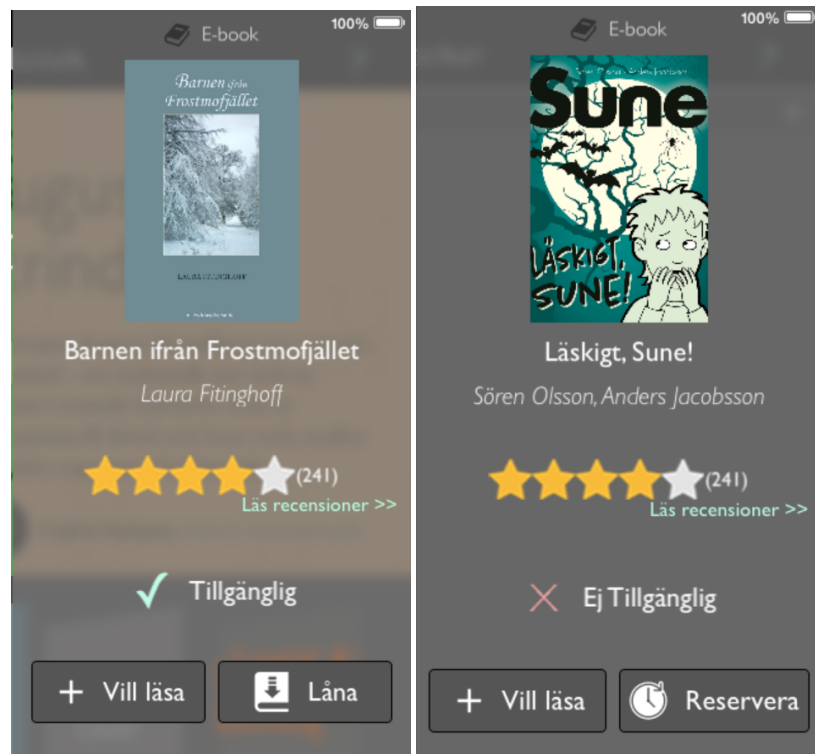


Figure 6.5 The “Borrow” and “Reserve” buttons.

### 6.2.3. Terminology and Font

When choosing the titles within the application it is vital that the users understand them. It is also important that they do not contain unnecessary words, making the title too long or complicated [50]. Icons can be used to further enhance the meaning of a title, especially when dealing with a short word. Words with all-capital letters can be used to draw attention in some cases but should be avoided otherwise. The most important titles of this application are the ones in the side menu (see figure 6.6). They have to be informative, but due to the lack of space, they also had to be shortened.

Since the target group consists of borrowers from Swedish public libraries, and that this application only is a prototype the whole content is in Swedish. Though it should be an alternative to give the users the possibility to change language in the actual finished product.

Fonts can be either serif or sans-serif which basically means that the letters have feet or not. Serifs make it easier for the users to recognize signs and follow rows, which is preferable in printed texts. However, computers cannot show as many points as a printed text, which makes



small font sizes hard to read on screens [51]. It can be a good idea to use one font for the headings, which are often bigger and to use a sans-serif for the text body.

The main titles, displayed in the Navigation bar of each view, use the font family Bellota (see figure 6.3 of the Navigation bar). Even though the font is a fairly simple sans-serif with subtle curves at the end of some letters, it gives the application a more unique look that users hopefully will recognize and associate with this library application. Avenir is used for the rest of the text, which is a legible sans-serif that resembles the popular font family Arial.

#### 6.2.4. Buttons, Icons and Badges

Buttons and clickable items should have a target of 44x44 points or more to make them as easy to hit as possible [48]. The size of the buttons in the application is therefore of at least this amount. The purpose of the buttons “Borrow” and “Reserve” are enhanced with icons (see figure 6.5). Icons are to be preferred together with a short title, as mentioned above. The icons must be well understandable or else it eliminates their purpose. Using metaphors as icons increases usability since they visualize a concept that the users are familiar with, as in the common case of the use of a magnifying glass for search.

Icons were added to the different options in the side menu (see figure 6.6). A magnifying glass used for the search page; a pillar building for the Library page to make the user associate with old school library buildings; books for the “My Bookshelf” page; a person talking for the Social page; and the commonly used gear for Settings.



*Figure 6.6 Icons and titles for the pages in the side menu.*

Besides the button change mentioned under constraints (see paragraph 6.2.2), the user needs clear information if a book is available for loan. This is accomplished with an icon in the form of a “check mark” (see figure 6.7). Since the mark can be interpreted differently between countries, e.g. in English speaking nations, it stands for affirmation in contrast to Sweden and Finland, where it sometimes meant “wrong” (usually when used in old school systems for correcting homework or tests). Younger users are probably more familiar with the meaning of the English version than older users. The purpose of the icon is further enhanced with the color green,

which is commonly known as “go”. On the other hand, if borrowing is unavailable a red cross, representing stop, is shown. An alternative to the check mark icon could have been to use a “thumbs up” icon or a green circle (see figure 6.8).

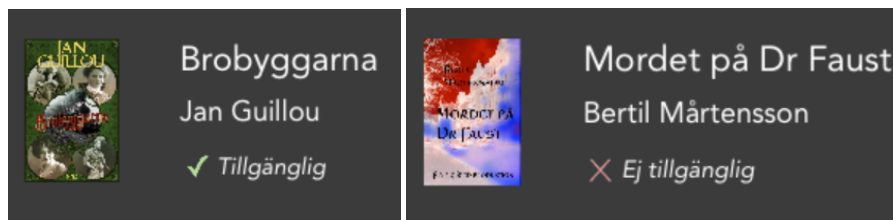


Figure 6.7 To the left - checkmark icon for available book. To the right - cross for unavailable.

An altogether different approach could have been to simply not show books that are unavailable. One problem with that solution could be when a user searches for a specific book and does not find it. She will probably believe that the book does not exist among the e-books that the library offers while in fact it is just not available at the moment.

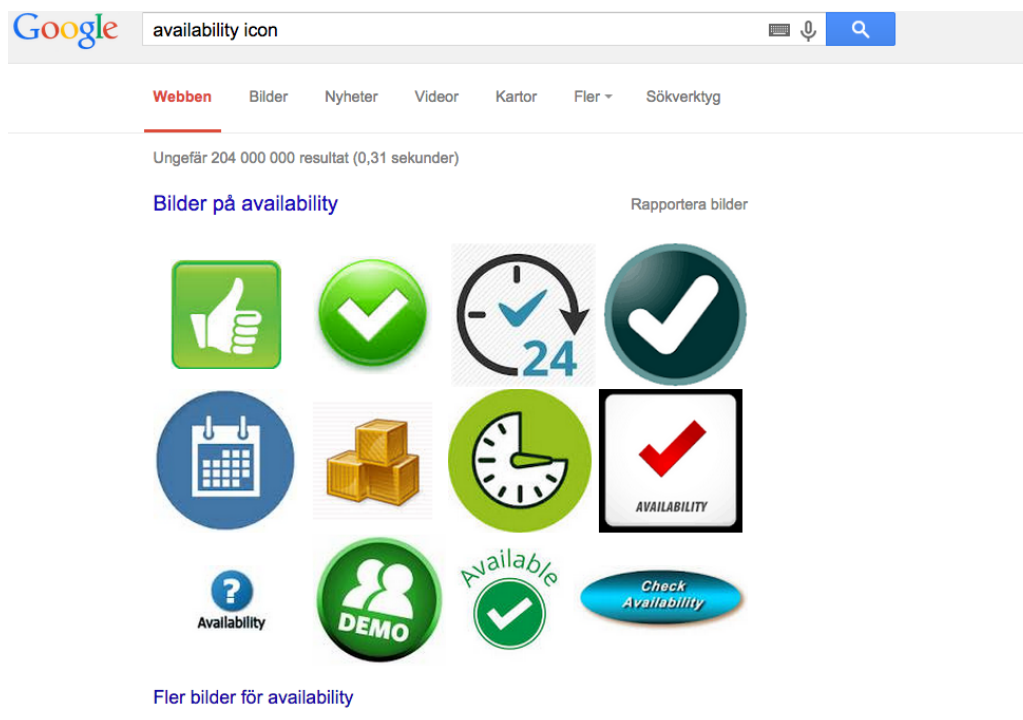


Figure 6.8 Google search for availability icons.

Another way to enhance information is to use badges on top of images, in this case the cover of the book. In the “My Bookshelf” view badges, in the right hand bottom corner (see figure 6.9), was used to either indicate that a book was only a sample or if the book was a borrowed book, visualize the amount of days left on the loan.



*Figure 6.9 Read sample badge in the “My Bookshelf” page to the left vs. “amount of days left” for borrowed books to the right.*

### **6.2.5. Colors**

The color scheme of the interface was important. The application needed a modern and sophisticated design where Spotify and Netflix are two great examples of applications that have beautiful interfaces. A dark background with white titles complement the colorful book covers that make them stand out and become more in focus. However, there are both advantages and disadvantages of using dark backgrounds.

Studies of whether dark or light backgrounds are best to use have been conducted since the 80s. Since white color reflects more light, white text on black screens reflects the words and they become harder to read. So in the case of applications which require focused reading, black text on white is recommended [52], this is also mentioned in the article about preferred e-book design [8]. Thus, this is the standard setting on the Reading mode. However, since some people like the opposite an option to change these colors is preferable.

### 6.3. Final user test

At the end of the fourth sprint, a test plan was conducted [53]. The goal of the test was to check the usability of the prototype and to answer questions regarding reading settings and social features. The test also analyzed if a new solution is necessary or if the current version is enough.

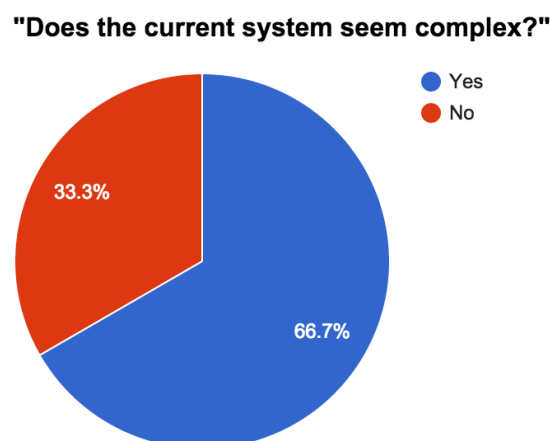
The test was constructed of four parts:

1. **Analysis of the current system:** a presentation and an explanation of the current solution followed by questions of its complexity.
2. **User scenarios:** four scenarios testing the prototype.
3. **Debriefing:** an interview with each participant.
4. **Survey:** a checklist where the participants rated the importance of reading and social functions.

The tests were all performed by six volunteer students of high technical background and the test is therefore a so called “Best Case Testing”. Meaning that the participants are expected to understand and interact with the application more easily than an average user and that the results should have an extra high success rate.

#### 6.3.1. Part 1: Analysis of the current system

After introducing and demonstrating the current system to the testers, each participant received two questions: “Does the current system seem complex?” and “Does this intimidate you from trying the process of borrowing e-books?”. A majority thought the system was complicated (see figure 6.10), and half of the testers thought that this would withhold them from using the system.



*Figure 6.10 Analysis of the current system.*

### **6.3.2. Part 2: User scenarios**

A significant majority of the users understood all parts of the design that the user stories covered, and the prototype therefore passed the user test. However, some concerns and useful feedback, like that the sorting buttons were small, should be considered.

### **6.3.3. Part 3: Debriefing**

It became clear under the debriefing interviews that the majority of the testers did not think that an e-book application could be as popular as Spotify or Netflix. Mostly since they believe that e-books are not as desirable as traditional books and because most people now a days listen to music and watch movies/tv-shows more frequently than they read. Another reason mentioned was that you cannot use the application simultaneously like when listening to music in Spotify and that it therefore would not be used as often.

One of the big questions in this study was to check if this new application and solution is worth its developing cost. Only 67 % said that they thought it would be, but most of the persons that were questioned did not read or like e-books in the first place. A more positive percentage, 83 %, said that they would download and try the application if it was released.

Analysis of the social aspect showed that 83 % thought that social media was important to them, but none of the test subjects thought that social features would make them use the application more often. Though, some believed that social aspects could increase the user experience for more experienced book readers.

A few of the participants mentioned that they would have liked if the search functionality would have been on the same page as The Malmö City Library page. Additionally, most of the user testers wanted a more clear indication where different book genres and categories were located. Both of these concerns are questions that have come up in discussions before and are already planned steps for the next version of the prototype.

### **6.3.4. Part 4: Survey**

The participants received a survey each, which they could look through individually and without the test leaders present. It contained two checklists, one concerning reading functions and the other social functions. They were rated depending on their importance for the application (see Appendix A.4.1.).

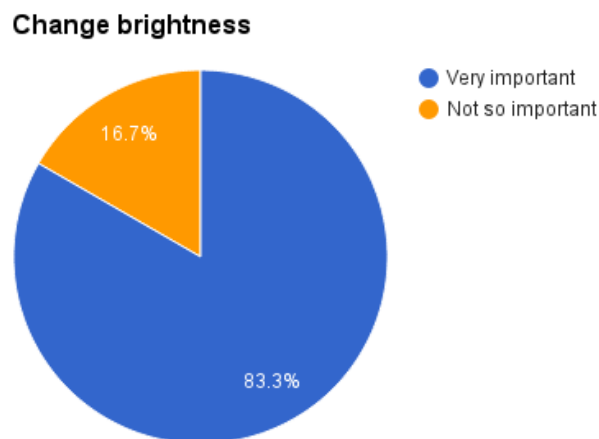
#### ***Reading features***

The checklist of suggested reading features contained the following functions:

- Change brightness
- Change to night mode

- Change margin/line spacing
- Change font
- Change text or background color
- Choose how to navigate between pages
- See percent read
- See reading rate

The only feature that was dominantly chosen as very important when reading an e-book was to be able to change the brightness (see figure 6.11). The remaining features were mostly considered as “nice to have but not important” for the reading experience.



*Figure 6.11 The importance of being able to change brightness*

### **Social features**

The five social features, believed to make the user experience more exciting were: ratings, sharing with friends, reading reviews, participate in book clubs and to be able to have challenges, e.g. reading goals. The majority of the participants liked the features, but they did not think they were a must to use the application (see figure 6.12-16). As presented in paragraph 6.3.3, 83% of the participants did not believe that social features would make them use the application more often.

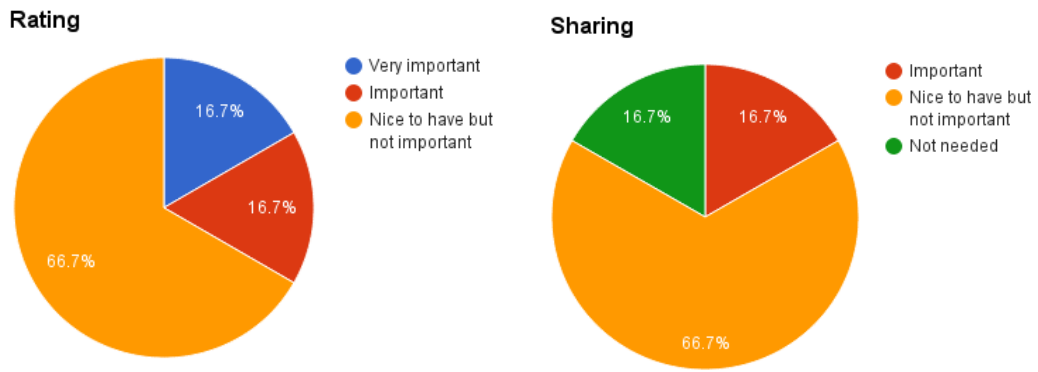


Figure 6.12 The importance of being able to rate books.

Figure 6.13 The importance of being able to share/recommend books to friends.

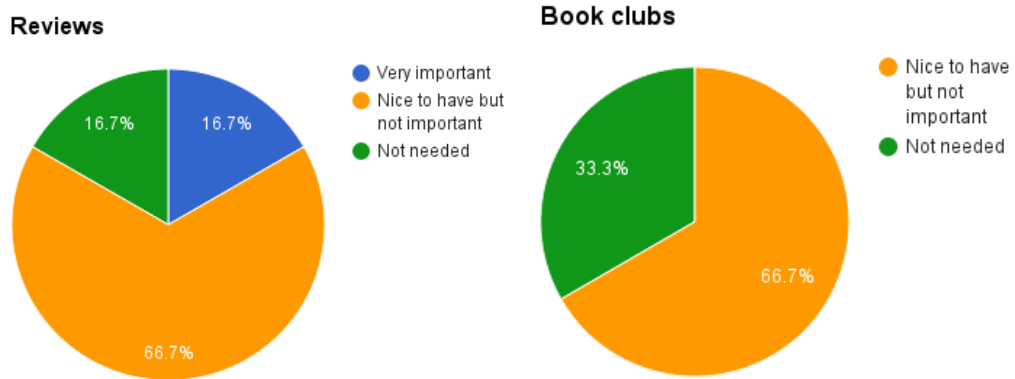


Figure 6.14 Being able to read reviews. Figure 6.15 Being able to partake in Book Clubs.

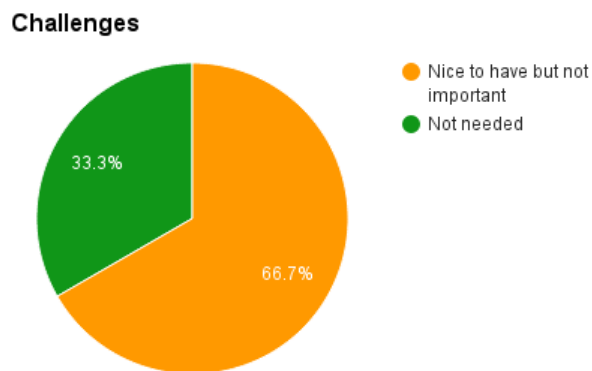


Figure 6.16 Being able to participate or make challenges.

## 7. Discussion

After the final user studies the work process of the thesis was evaluated. A couple of complicated aspects were encountered during the project. Below is a discussion of some of the problems that arose and some thoughts for the future.

### 7.1. Implementation problems

Implementing the perfect e-book application comes with a set of dependencies which makes the development process harder. If you do not want to spend time and money implementing an entirely new version that supports DRM protected files, the solution had to be based on an application like DL Reader. Another option could have been to make a whole new solution to protect files, which also would have been very pricey. In addition, all the actual book data has to be retrieved from Elib. All these dependencies takes time to integrate, which was proven during this Master thesis.

The first bottleneck of the project was waiting for the computers (used for development) to arrive, personal computers were used while waiting. After the first program settings and the unpacking of the new computers was completed the process of downloading and installing Parallels Desktop, Windows 10 for Mac and Visual studios started. This was recommended by an Elib developer and took a few hours, but was unfortunately never used. These and other unexpected encounters and problems took away valuable hours from the project, but these kind of delays are something all developers have to take into account when planning a project.

#### 7.1.1. DL Reader and its installation

The user guide and installation guide provided with the DL Reader code was not very satisfying. The whole installation process from start to finish took about two weeks because of the unclear instructions and trying to make sense of how to proceed with the build of all the provided files. Before the intended order to unpack all the file was discovered, the debug system in Xcode pointed out several issues and errors without an explanation how to repair them. Waiting for the installation to be completed resulted in a lot of time wasted on implementing code that the DL Reader already had.

This could have been avoided if a more careful examination of the DL Reader Demo (see paragraph 3.3.1) would have been studied before hand. All those hours spent was fortunately not a total time waster since more knowledge of both Objective-C and Xcode came as a result. The bigger problem was instead when small changes needed to be made in the 400 000 lines of code without a proper user guide with directions where these changes could be executed. Due to this and the lack of sufficient documentation in the actual code, simple modifications were very time consuming and complex.



Another bottleneck was waiting for the RMSDK certificate. Which was needed to be able to further implement the DL Reader code, this certificate had to be updated once every month. If this was forgotten, the implementation could not continue until a new certificate was obtained.

All these problems and the fact that the actual reason behind choosing DL Reader, the DRM protection support, did not work properly makes you wonder if another program or solution would have been better suited for this application.

### **7.1.2. Xcode and iOS version changes**

Working with Apple products entails that updates with new features are released frequently. New functionalities are often not the only updates in a new Xcode version, the layout of the program may also have been changed. Which means that menu options and buttons may have moved, disappeared or may have been re-named. New iOS versions with new design rules also cause elements that were used in a lot of tutorials to no longer be supported. As a beginner of Xcode a lot of time was spent looking for tutorials explaining certain functions and design implementations. Thus, finding suitable tutorials have sometimes become quite the hassle since they need to match the current version of Xcode to be fully understandable and to be able to follow the whole tutorial as intended.

### **7.1.3. Git synchronization with Xcode**

Another time consuming issue encountered during this project was the synchronization with GitHub (see paragraph 2.3.5). A lot of time was spent re-doing old work when code was accidentally removed since a commit was not properly saved. The reason for these problems was a result of scarce experience with git. The problems might have been avoided if git had been handled through the terminal instead of Xcode and most certainly if more experience had been possessed.

## **7.2. Evaluation of methods used**

The methods used in this thesis work were satisfactory. Working with Scrum in an Agile environment made the process run smoothly and having partial goals helped to get a better overview of the entire process. What could have been improved is the amount of research done in the beginning of the study. Even though no other solutions for library applications was found some international libraries could have been studied to grow a broader knowledge of the field and its current solutions.

Moreover, the test studies could have been more structured. It was difficult to coordinate test sessions and attract enough participants. The test subjects for the user studies of this thesis consisted of a small group, who all had an engineering background. A more extensive and varied crowd of people, with different levels of technical knowledge and more experienced e-book users should have been evaluated. To draw general conclusions from these insufficient test data was therefore not possible.

Some of the design choices implemented according to the prototype created in Balsamiq were removed when testing the Hi-Fi prototype, e.g. the filters in the Search view (paragraph 4.4). Implementation time could have been saved if these unsuitable design choices were discovered in earlier tests. In retrospect, tests on the Lo-Fi and Mid-Fi prototype should also have been conducted.

### **7.3. Is the era of borrowing e-books ahead?**

The purpose of this Master thesis was to evaluate how public libraries can attract more people into borrowing e-books and if it is possible to make the whole experience easier to interact with. Through surveys and user studies, it became obvious that the interest for e-books in general is increasing. Like the study from the Chinese university in Hong Kong stated (paragraph 1.3.5), not having to carry heavy books on commutes or trips is a major advantage. However, the future of borrowing e-books is more uncertain. There are several aspects that the public libraries have to consider if they want to attract more readers to their virtual bookshelves.

The questionnaire in paragraph 3.2. indicate that the biggest drawback is the limited selection of e-books that the libraries provide. If the goal of the libraries is to attract more borrowers they consequently need to have more books to offer. One reason for the limited amount of books is that publishers often apply a quarantine to newly released books. Many enthusiastic readers might not want to wait for the books to be released as borrowable books and a high rate will instead turn to bookstores. Only allowing two loans per month, like Malmö City Library does, is also a major drawback.

Other aspects concern the technical solution. The users need a tool which makes it easy to find e-books. It should not be a challenge to find a specific book which many borrowers perceive today. The reading process must provide a great user experience with many options so that they can adjust the text and background to fit their personal preferences. Today the pages of an e-book are sent one at a time as images, since they need to be DRM protected. The possibilities of changing the text layout would be close to endless if the e-books instead were read as .epub files. Accessibility would increase through optional fonts that can be read despite dyslexia, like Open Dyslexic (paragraph 2.2.2) or personal color choices which makes the text evident even for users with different types of color blindness.

During the development of this prototype it became obvious that the chosen target group is too wide. To create an application that satisfies both beginners and experienced e-book readers is hard. The average user might not care for social features or even find them overwhelming. On the other hand, the social aspects can be the future reason that experienced e-book readers chose this application over already established solutions like Bluefire Reader.

To evaluate this theory, the tests should also have been conducted on readers that already uses the current solution and who are positive to borrowing e-books. The test results could have been very different in that case. Therefore, knowledge regarding the interest of social media may depend hugely on the addressed target group. If the new application were of satisfaction for the already established users, then their utilization might attract a broader user audience in time.

However, in the end it still comes down to the reading experience which is the functionality users will spend most hours interacting with. If the reading features are inadequate [8] the application will not be used for reading and the main purpose is thereof lost.

## 8. Conclusion

After researching and developing this HI-FI prototype the conclusion is that there are still much work ahead before more people will start transitioning into borrowing e-books. To measure if this solution is indeed easier and more user friendly than the current solution is not easy. Furthermore, to answer if it all is worth the hassle and the high cost of the application being fully developed is an even tougher question to answer.

To base the application on a solution like DL Reader, which is needed to read DRM protected files, implicate a huge dependency. For future development a solution might be to further investigate the possibility of instead streaming e-books, as the media in e.g. Spotify or Netflix, or choose a more documented application base than DL Reader. To compare and reach for a popularity similar to Spotify is not fair, since the book market is very traditional and hard to change. In the end, overcoming the obstacles mentioned in this thesis, is only a question of financings.

The results from our final user studies indicate that this prototype and solution could be an easier and more user friendly alternative to the current solution if all complexity with both a library card PIN and an Adobe ID account could be combined in a one step solution.

## 9. References

- [1] A. Mebius mfl. (2013). Förlagen styr lån av e-böcker. (Online) *SvD*. Available: <[http://www.svd.se/opinion/brannpunkt/forlagen-styr-lan-av-e-bocker\\_8584272.svd](http://www.svd.se/opinion/brannpunkt/forlagen-styr-lan-av-e-bocker_8584272.svd)> (2015-01-30).
- [2] Jayway. (Home page) Available: <<http://www.jayway.com>> (2015-05-14).
- [3] Kaliber demo app. (Online) Available: <<https://www.flinto.com/p/bf71d81a>> (2015-01-28).
- [4] Kaliber. (Home page) Available: <<http://kaliberapp.se/>> (2015-01-28).
- [5] Datalogics. (Home page) Available: <<http://www.datalogics.com/>> (2015-05-12).
- [6] Stockholm Public Library (2014). Biblioteket\_Rapport\_140821\_ed21. (Online) Available: <[http://bloggar.biblioteket.se/digitalabiblioteket/wp-content/uploads/2014/12/Biblioteket\\_Rapport\\_140821\\_ed21.pdf](http://bloggar.biblioteket.se/digitalabiblioteket/wp-content/uploads/2014/12/Biblioteket_Rapport_140821_ed21.pdf)> (2015-01-30).
- [7] Stockholm Public Library (2013). Så talade e-bokslåntagarna (last updated 2013-10-10). (Online) Available: <<https://biblioteket.stockholm.se/inspiration/artikel/så-talade-e-bokslantagarna>> (2015-01-30).
- [8] Chong, P-F., Lim, Y-P. and Ling; S-W. (2009) On the Design Preferences for Ebooks, *IETE Technical Review*, 26:3, 213-222
- [9] Lam, P. Lam, S-L., Lam, J. and McNaught, C. (2009) Usability and usefulness of eBooks on PPCs: How students' opinions vary over time, *Australasian Journal of Educational Technology*, 25:1, 30-44
- [10] Schwaber, K. (2004). *Agile Project Management with Scrum (Developer Best Practices)*. Redmond, WA: Microsoft Press.
- [11] Trello. (Home page) Available: <<https://trello.com>> (2015-01-28).
- [12] Norman, D. and Draper, S. (1986). *User-Centered System Design: New Perspectives on Human- Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

- [13] Accessibility. (Online) *Apple*. Available: <<https://ssl.apple.com/accessibility/ios/>>. (2015-04-15).
- [14] Ingre J. (2013). Typsnitt för synskadade. (Online) *Cap & Design*. Available: <<http://capdesign.idg.se/2.990/1.485076/typsnitt-for-synskadade>> (2015-04-15).
- [15] Open Dyslexic. (Home page) Available: <<http://opendyslexic.org>> (2015-04-15).
- [16] Prototyping. (Online) *Usability.gov*. Available: <<http://www.usability.gov/how-to-and-tools/methods/prototyping.html>> (2015-02-03).
- [17] FluidUI. (Home page) Available: <<https://www.fluidui.com>> (2015-02-02).
- [18] Invision. (Home page) Available: <<http://www.invisionapp.com>> (2015-02-02).
- [19] Flinto. (Home page) Available: <<https://www.flinto.com>> (2015-02-02).
- [20] Balsamiq. (Home page) Available: <<https://balsamiq.com>> (2015-02-02).
- [21] Elib. (Home page) Available: <[https://www.elib.se/ebook\\_about\\_elib\\_eng.asp](https://www.elib.se/ebook_about_elib_eng.asp)> (2015-02-02).
- [22] About Objective-C. (Online) *Apple*. Available: <<https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>> (2015-01-28).
- [23] Xcode. (Home page) Available: <<https://developer.apple.com/xcode/>> (2015-01-28).
- [24] JSON. (Home page) Available: <<http://www.json.org>> (2015-02-02).
- [25] Crockford, D. (2006). The application/json Media Type for JavaScript Object Notation (JSON). (Online) *Json.org*. Available: <<http://tools.ietf.org/html/rfc4627>> (2015-02-02).
- [26] T. Bray, Ed. (2014). The JavaScript Object Notation (JSON) Data Interchange Format. (Online) *Google, Inc*. Available: <<http://tools.ietf.org/html/rfc7159>> (2015-02-02).
- [27] Richardson, L. and Ruby, S (2007). *RESTful Web Services*. Sebastopol, CA: O'Reilly Media Inc.
- [28] Git. (Home page) Available: <<http://git-scm.com>> (2015-01-28).
- [29] GitHub. (Home page) Available: <<https://github.com>> (2015-01-28).
- [30] Malmö City Library. (Home page) Available: <<http://malmo.elib.se>> (2015-01-28).

- [31] DL Reader. (Home page) Available: <<http://www.datalogics.com/products/ebook/rmsdk/dlreader>> (2015-02-02).
- [32] Bluefire reader. (Home page) Available: <<http://www.bluefirereader.com>> (2015-01-28).
- [33] Adlibris Mondo. (Home page) Available: <<https://www.adlibris.com/mondo>> (2015-02-02).
- [34] Kindle. (Home page) Available: <<http://www.amazon.com/Kindle-eBooks>> (2015-02-02).
- [35] Spotify. (Home page) Available: <<https://www.spotify.com>> (2015-01-28).
- [36] Netflix. (Home page) Available: <<https://www.netflix.com>> (2015-01-28).
- [37] Bokon. (Home page) Available: <<https://bokon.se>> (2015-02-02).
- [38] Steinberg, D. H. (2012). *iPad and iPhone App Development*. (e-book) *Stanford University CS193P: Dim Sum Thinking, Inc.* Available: App Store. (2015-01-28).
- [39] Mac and Computer Help. (2015). (Video files). Available: <<https://www.youtube.com/user/macandcomputerhelp>> (2015-01-29).
- [40] Hodson R. (2013). Ry's Objective-C Tutorial. (Online) *RyPress*. Available: <<http://rypress.com/tutorials/objective-c/index>> (2015-01-29).
- [41] RESTkit. (Home page) Available: <<http://restkit.org>> (2015-02-16).
- [42] Github for RESTkit. (Home page) Available: <<https://github.com/RestKit/RestKit>> (2015-02-16).
- [43] Github for JSONModel. (Home page) Available: <<https://github.com/icanzilb/JSONModel>> (2015-03-02).
- [44] Github for SDWebImage. (Home page) Available: <<https://github.com/rs/SDWebImage>> (2015-02-16).
- [45] Github for AsyncImageView. (Home page) Available: <<https://github.com/nicklockwood/AsyncImageView>> (2015-03-02).
- [46] Designing for iOS. (Online) *Apple*. Available: <<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG>> (2015-04-09).

[47] Navigation. (Online) *Apple*. Available:  
<[https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Navigation.html#/apple\\_ref/doc/uid/TP40006556-CH53-SW1](https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Navigation.html#/apple_ref/doc/uid/TP40006556-CH53-SW1)> (2015-04-09).

[48] Adaptivity and Layout. (Online) *Apple*. Available:  
<[https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/LayoutandAppearance.html#/apple\\_ref/doc/uid/TP40006556-CH54-SW1](https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/LayoutandAppearance.html#/apple_ref/doc/uid/TP40006556-CH54-SW1)> (2015-04-09).

[49] Rogers, Y., Sharp, H. and Preece, J. (2011). *Interaction Design: Beyond Human - Computer Interaction, 3rd Edition*. John Wiley & Sons, Inc.

[50] Terminology and Wording. (Online) *Apple*. Available:  
<[https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/FeedbackCommunication.html#/apple\\_ref/doc/uid/TP40006556-CH56-SW1](https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/FeedbackCommunication.html#/apple_ref/doc/uid/TP40006556-CH56-SW1)> (2015-04-09).

[51] Serif vs Sans the final battle. (2013). (Online) *Web Designer Depot*. Available:  
<<http://www.webdesignerdepot.com/2013/03/serif-vs-sans-the-final-battle>> (2015-04-07).

[52] Knight K. (2009). The Dos and Don'ts of Dark Web Design. (Online) *Web Designer Depot*. Available: <<http://www.webdesignerdepot.com/2009/08/the-dos-and-donts-of-dark-web-design/>> (2015-04-07).

[53] Chisnell, D. and Rubin, J. (2008). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, 2nd Edition*. Indianapolis, IN: Wiley Publishing Inc.

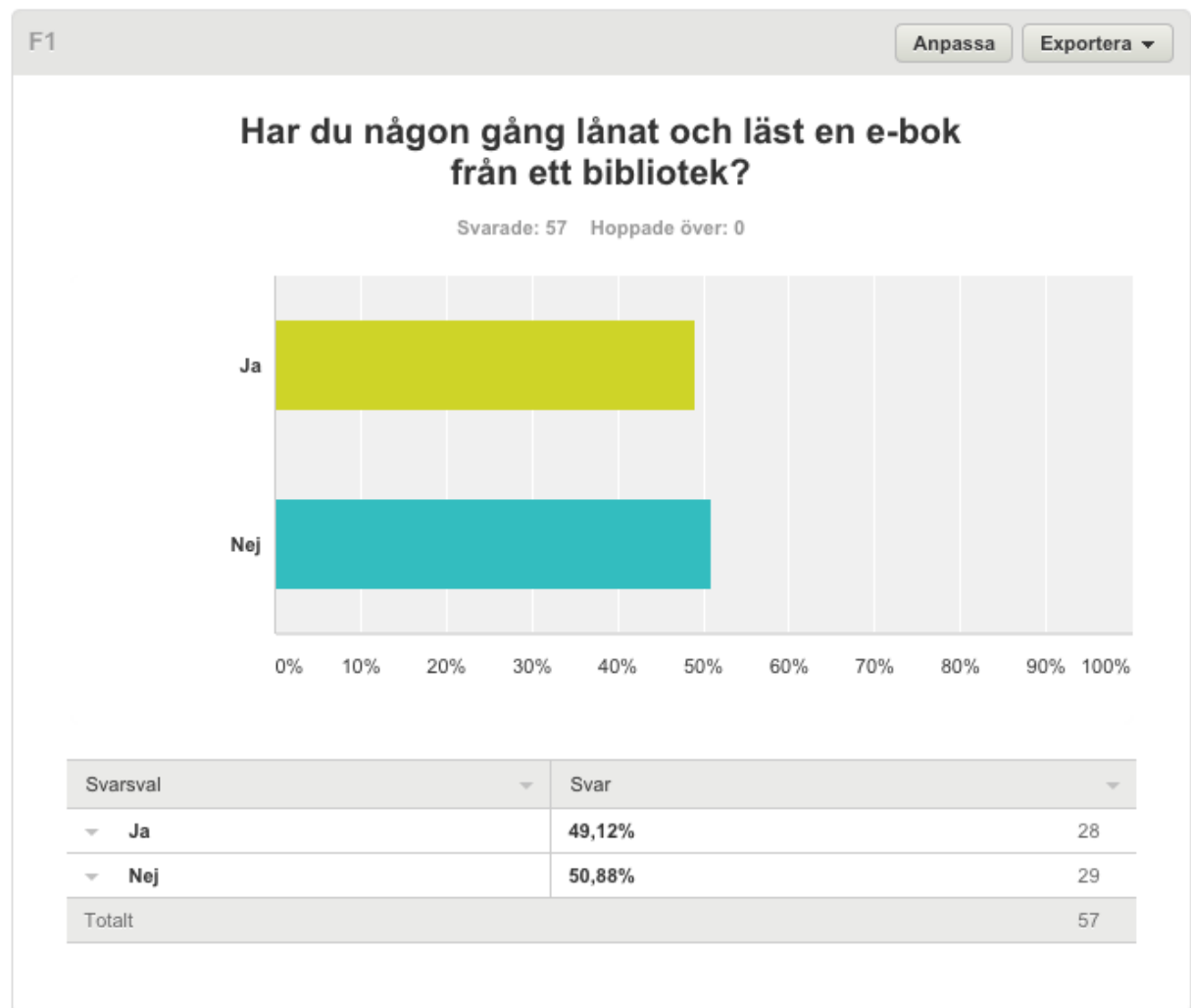


# Appendix

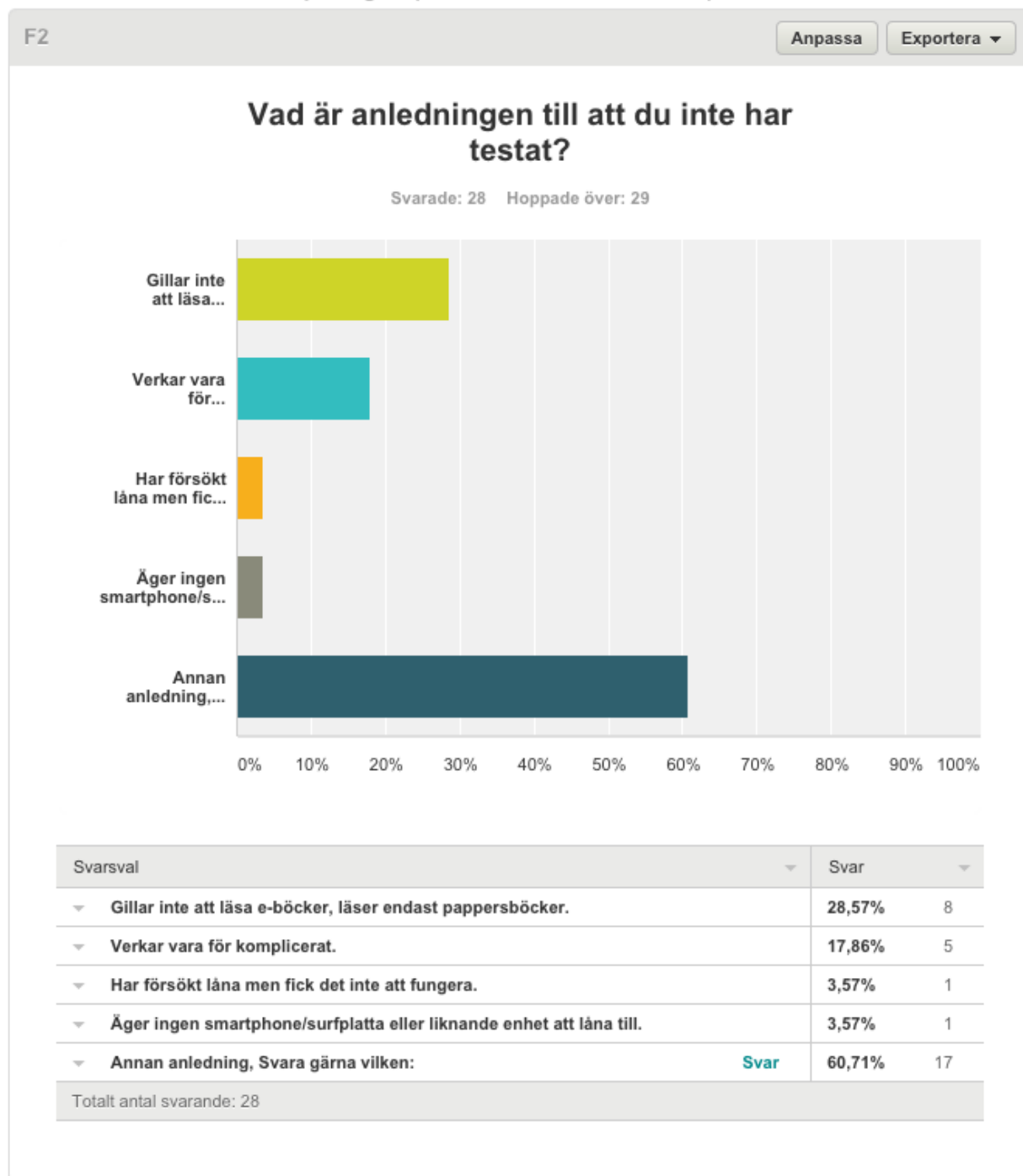
## A.1 Surveys

### A.1.1 Survey version 1

SIDA 1



SIDA 2: Ifall du svarade NEJ på fråga 1 (du har INTE lånat nån e-bok)



SIDA 3: Ifall du svarade JA på fråga 1 (du HAR lånat e-böcker)

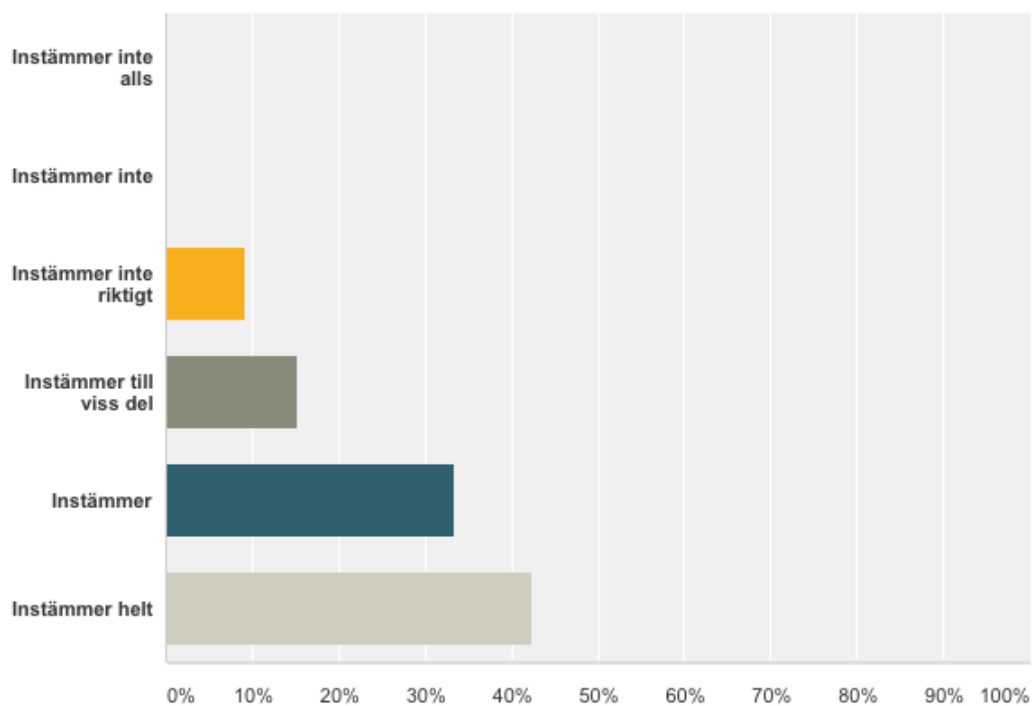
F3

Anpassa

Exportera ▼

### Jag har bred teknisk erfarenhet.

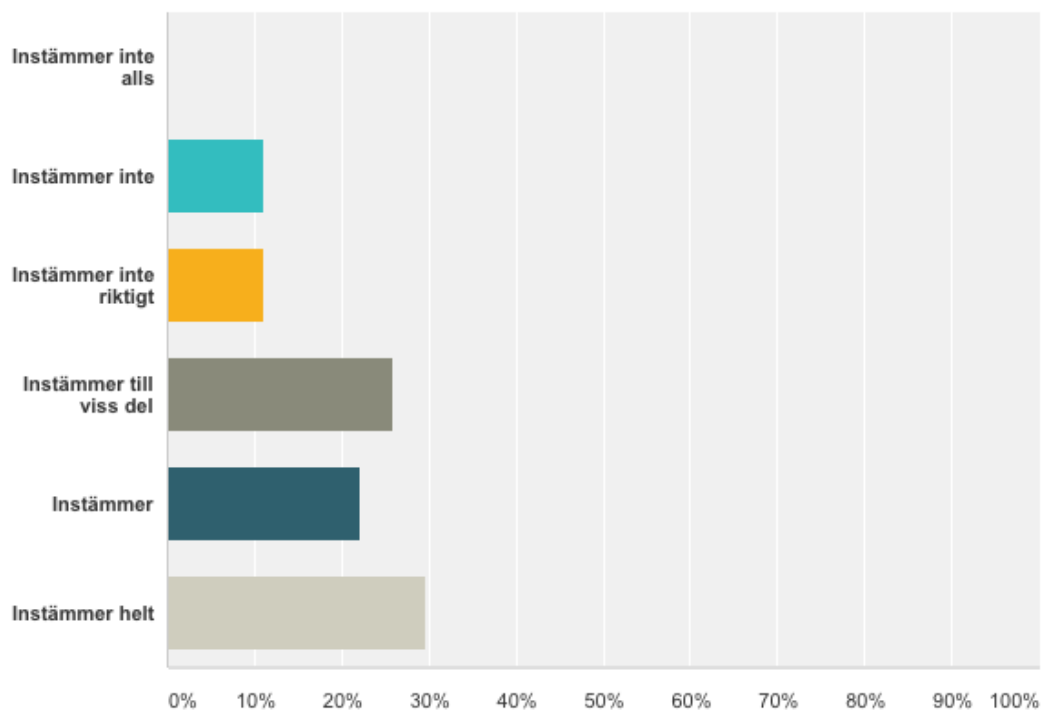
Svarade: 33 Hoppade över: 24



Svarsval	Svar
▼ Instämmer inte alls	0,00% 0
▼ Instämmer inte	0,00% 0
▼ Instämmer inte riktigt	9,09% 3
▼ Instämmer till viss del	15,15% 5
▼ Instämmer	33,33% 11
▼ Instämmer helt	42,42% 14
Totalt	33

## Jag tyckte att det var LÄTT att låna första gången jag testade.

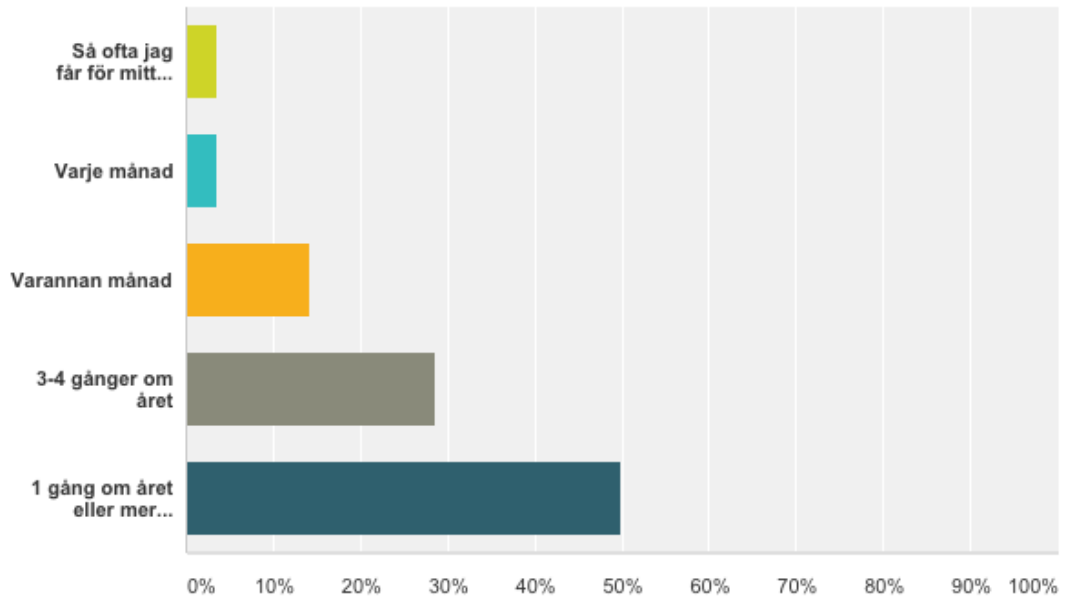
Svarade: 27 Hoppade över: 30



Svarsval	Svar
▼ Instämmer inte alls	0,00% 0
▼ Instämmer inte	11,11% 3
▼ Instämmer inte riktigt	11,11% 3
▼ Instämmer till viss del	25,93% 7
▼ Instämmer	22,22% 6
▼ Instämmer helt	29,63% 8
Totalt	27

## Hur ofta lånar du e-böcker från Biblioteket? (välj det du tycker passar bäst)

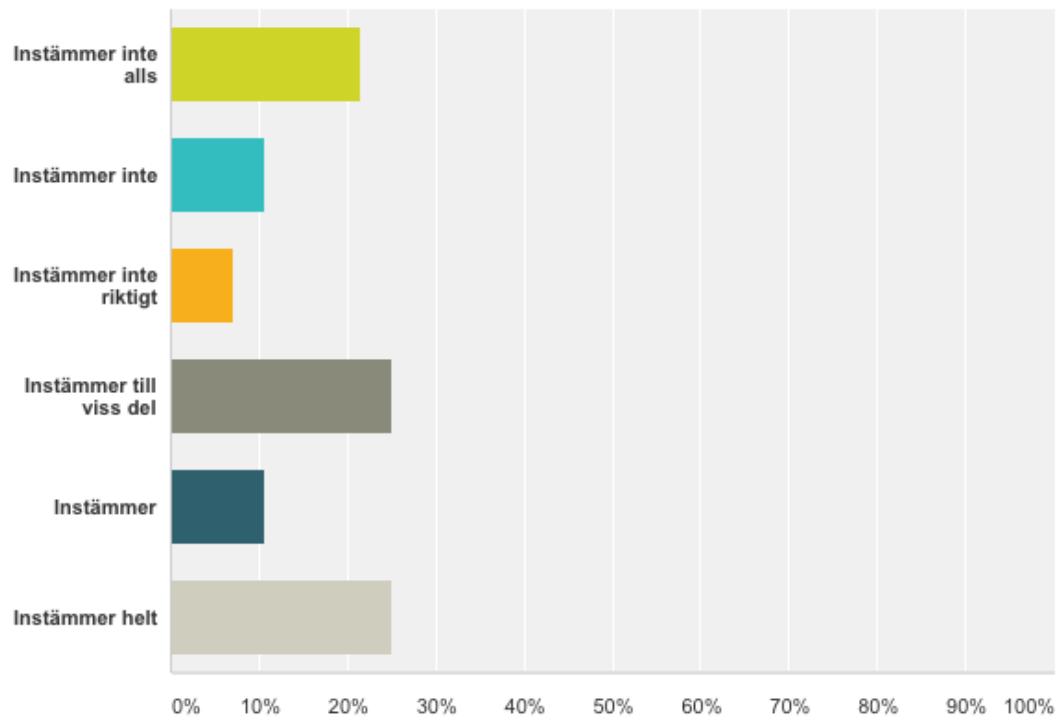
Svarade: 28 Hoppade över: 29



Svarsval	Svar
▾ Så ofta jag får för mitt bibliotek	3,57% 1
▾ Varje månad	3,57% 1
▾ Varannan månad	14,29% 4
▾ 3-4 gånger om året	28,57% 8
▾ 1 gång om året eller mer sällan	50,00% 14
Totalt	28

## Jag tycker om att använda en separat app för att läsa boken jag lånat.

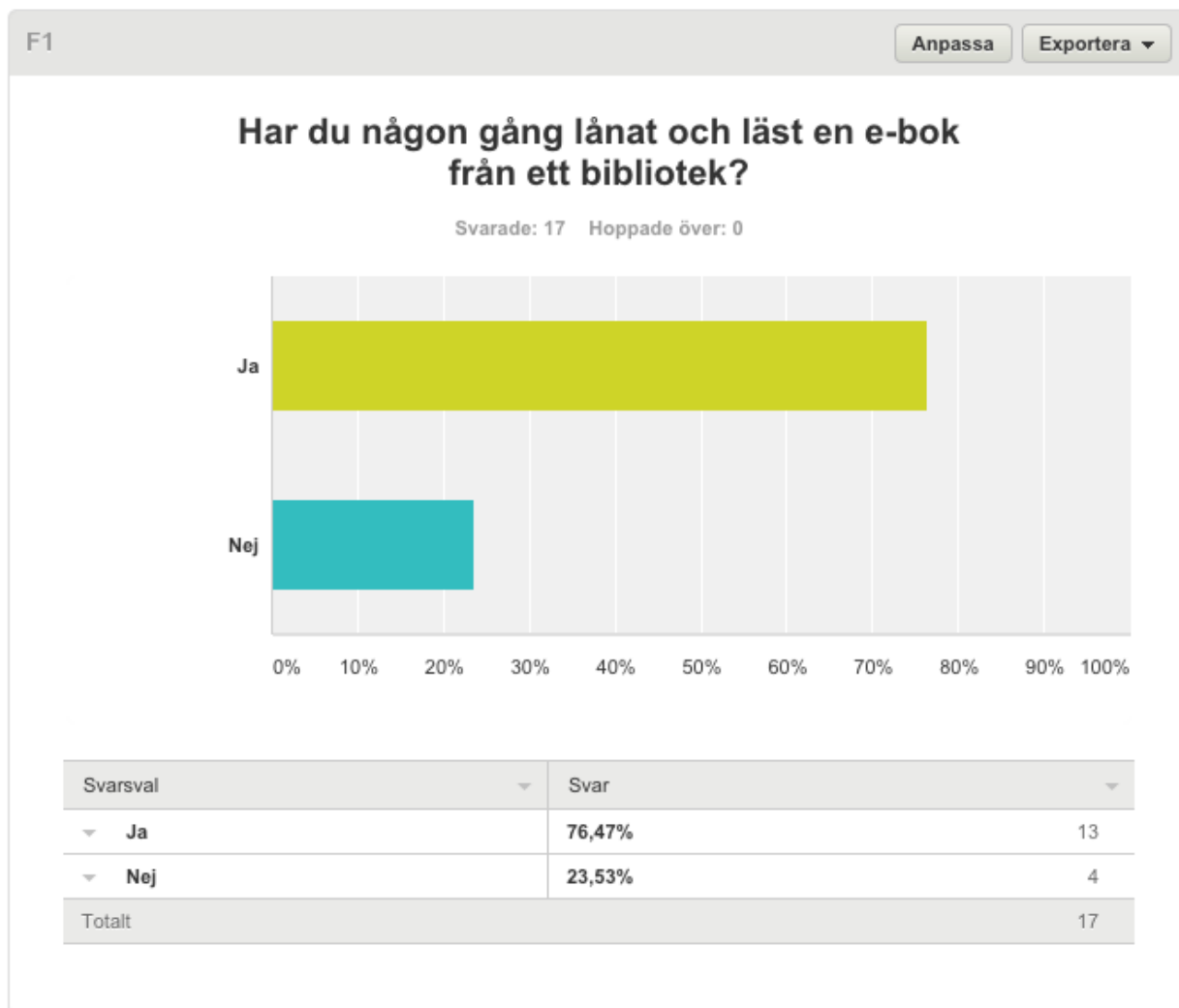
Svarade: 28 Hoppade över: 29



Svarsval	Svar
▼ Instämmer inte alls	21,43% 6
▼ Instämmer inte	10,71% 3
▼ Instämmer inte riktigt	7,14% 2
▼ Instämmer till viss del	25,00% 7
▼ Instämmer	10,71% 3
▼ Instämmer helt	25,00% 7
Totalt	28

## A.1.2 Survey version 2

SIDA 1



SIDA 2: Ifall du svarade NEJ på fråga 1 (du har INTE lånat nån e-bok)

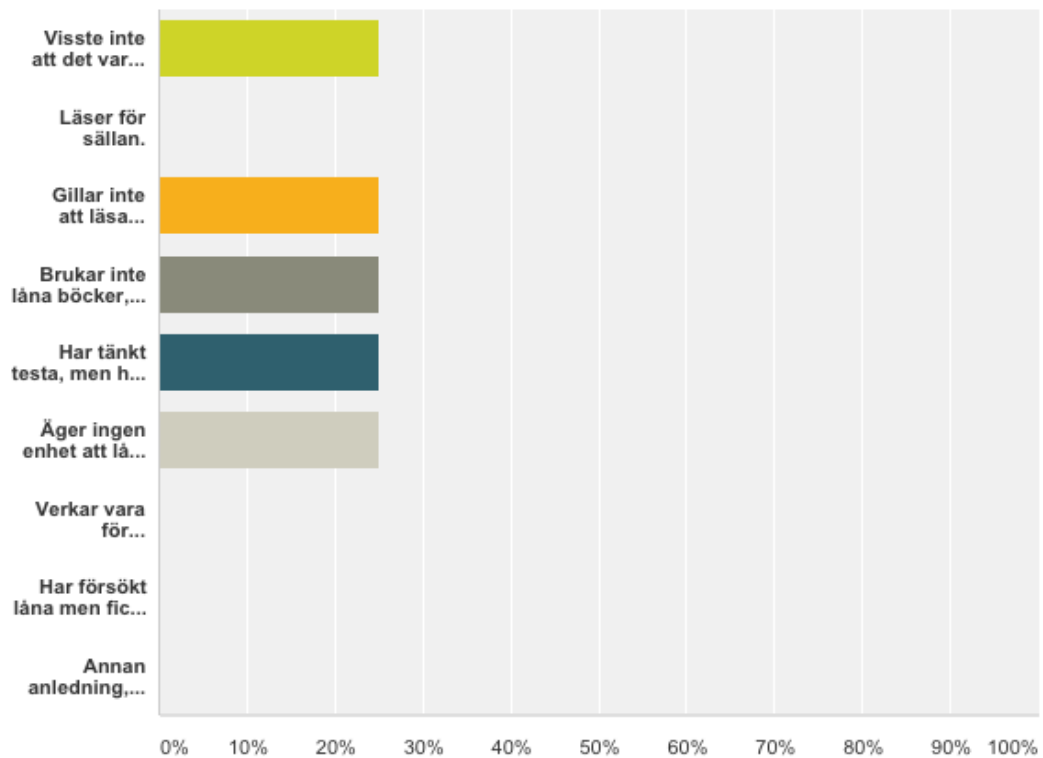
F2

Anpassa

Exportera ▾

## Vad är anledningen till att du inte har testat?

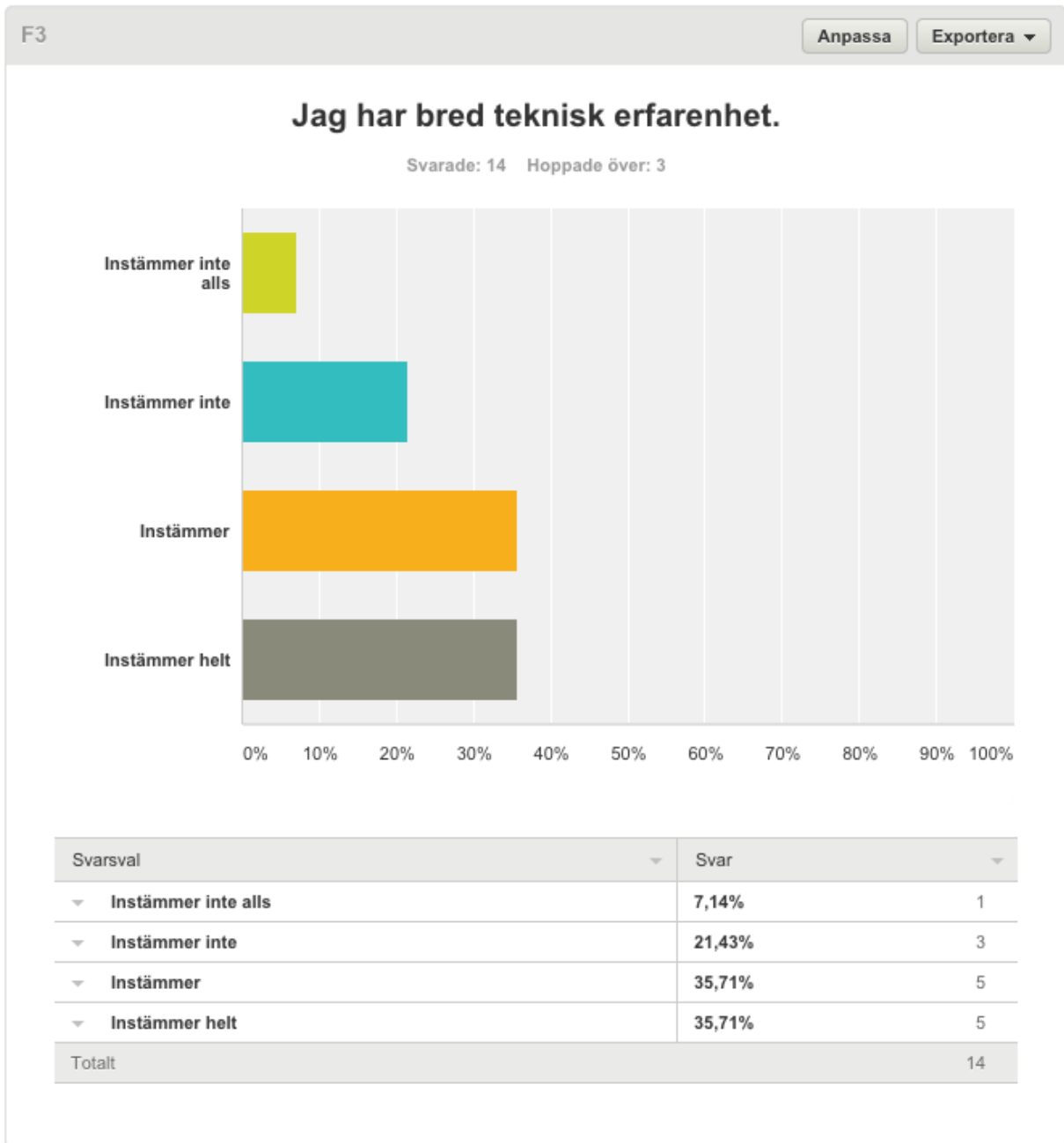
Svarade: 4 Hoppade över: 13



Svarsval	Svar
▾ Visste inte att det var möjligt att låna e-böcker.	25,00% 1
▾ Läser för sällan.	0,00% 0
▾ Gillar inte att läsa e-böcker (läser endast pappersböcker).	25,00% 1
▾ Brukar inte låna böcker, vill hellre köpa dem.	25,00% 1
▾ Har tänkt testa, men har inte blivit av än.	25,00% 1
▾ Äger ingen enhet att låna till /Fungerar inte för min enhet/läsplatta.	25,00% 1
▾ Verkar vara för komplicerat/svårt.	0,00% 0
▾ Har försökt låna men fick det inte att fungera.	0,00% 0
▾ Annan anledning, Svara gärna vilken: <span style="color: green;">Svar</span>	0,00% 0
Totalt antal svarande: 4	

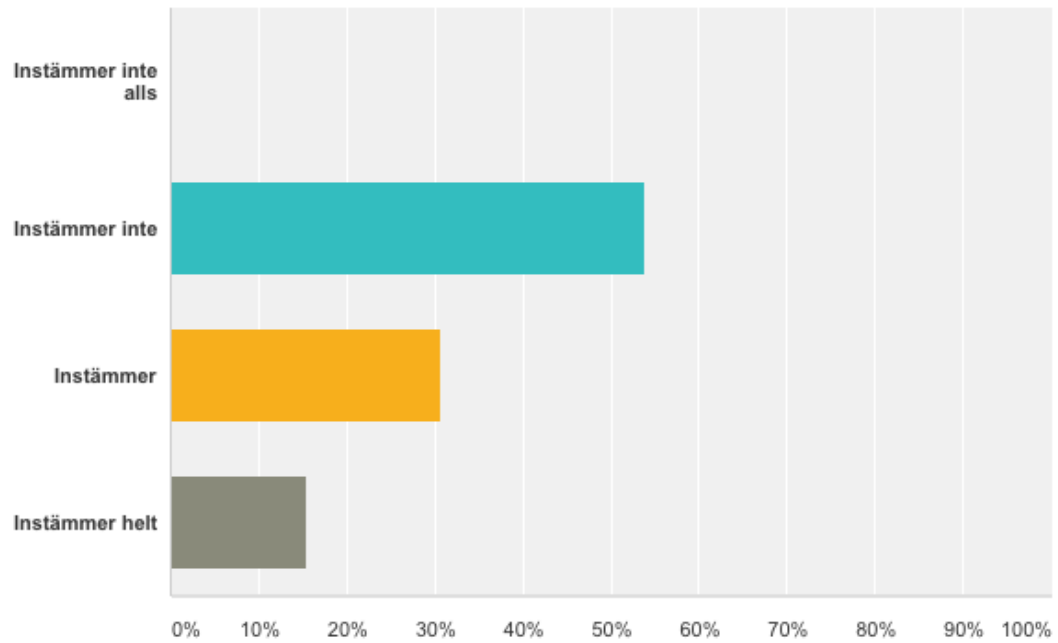


SIDA 3: Ifall du svarade JA på fråga 1 (du HAR lånat e-böcker)



## Jag tyckte att det var LÄTT att låna första gången jag testade.

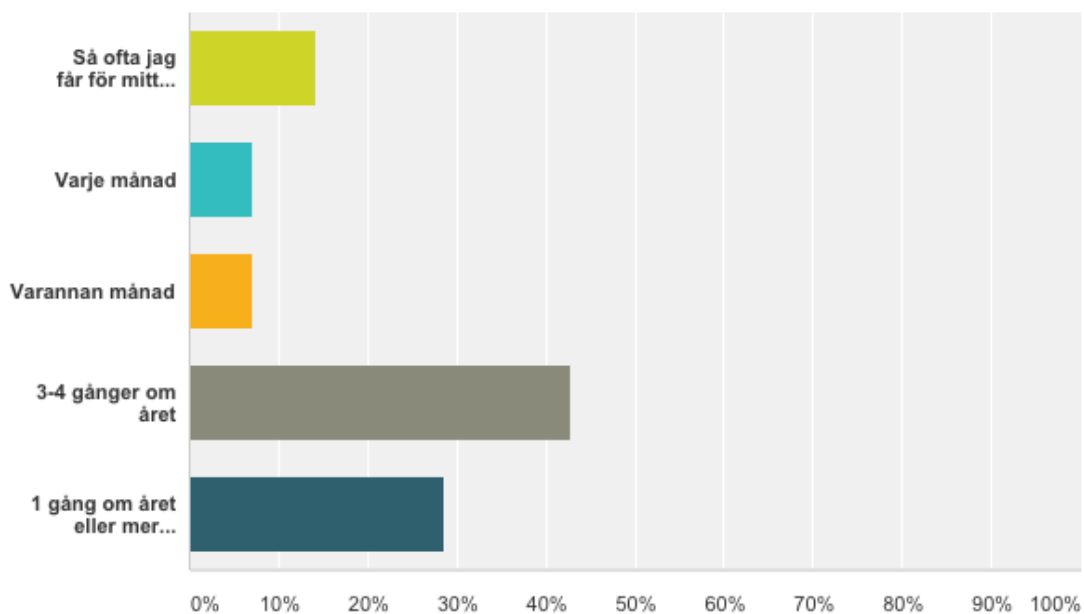
Svarade: 13 Hoppade över: 4



Svarsval	Svar
▼ Instämmer inte alls	0,00% 0
▼ Instämmer inte	53,85% 7
▼ Instämmer	30,77% 4
▼ Instämmer helt	15,38% 2
Totalt	13

## Hur ofta lånar du e-böcker från Biblioteket? (välj det du tycker passar bäst)

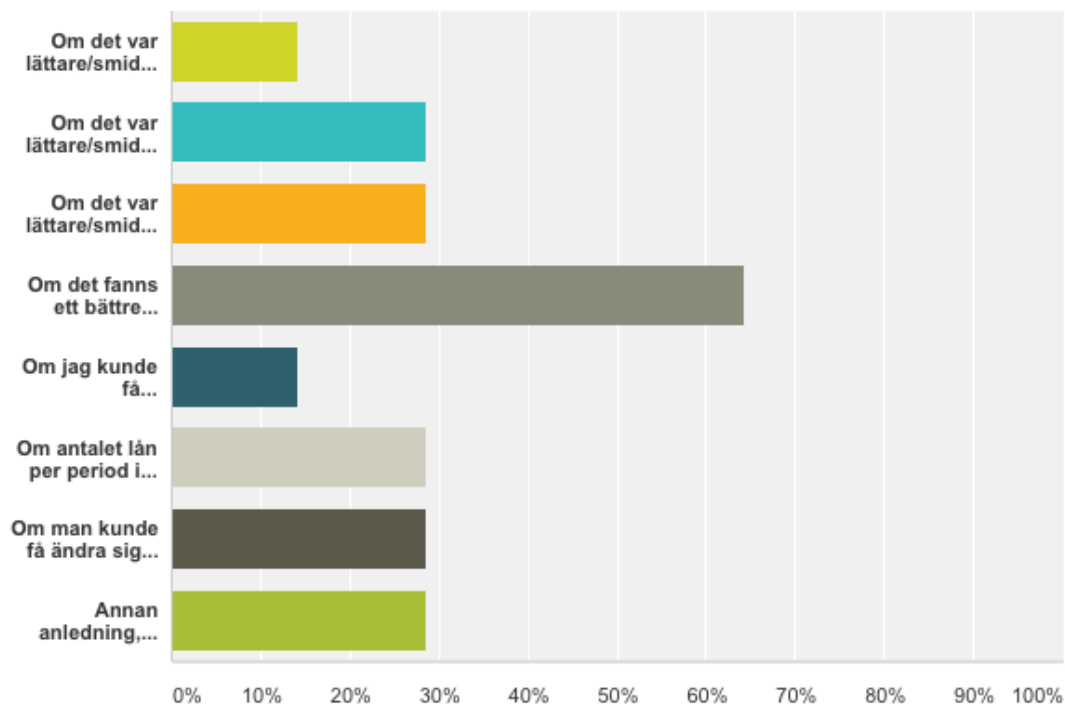
Svarade: 14 Hoppade över: 3



Svarsval	Svar
▼ Så ofta jag får för mitt bibliotek	14,29% 2
▼ Varje månad	7,14% 1
▼ Varannan månad	7,14% 1
▼ 3-4 gånger om året	42,86% 6
▼ 1 gång om året eller mer sällan	28,57% 4
Totalt	14

## Vad skulle få dig att låna oftare?

Svarade: 14 Hoppade över: 3

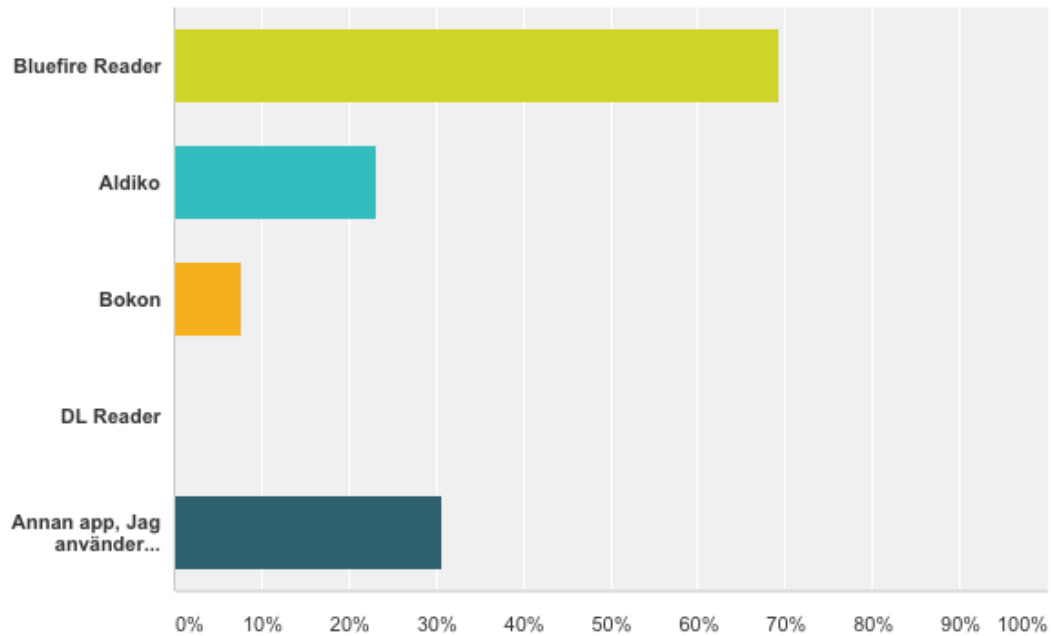


Svarsval	Svar
▼ Om det var lättare/smidigare att låna.	14,29% 2
▼ Om det var lättare/smidigare att läsa.	28,57% 4
▼ Om det var lättare/smidigare att hitta böcker.	28,57% 4
▼ Om det fanns ett bättre utbud böcker.	64,29% 9
▼ Om jag kunde få specialanpassade rekommendationer för kategorier jag valt.	14,29% 2
▼ Om antalet lån per period inte var lika begränsat.	28,57% 4
▼ Om man kunde få ändra sig och byta en bok man lånat.	28,57% 4
▼ Annan anledning, Svara gärna vilken: <b>Svar</b>	28,57% 4

Totalt antal svarande: 14

## Vilken app använder du för att läsa böckerna du lånat?

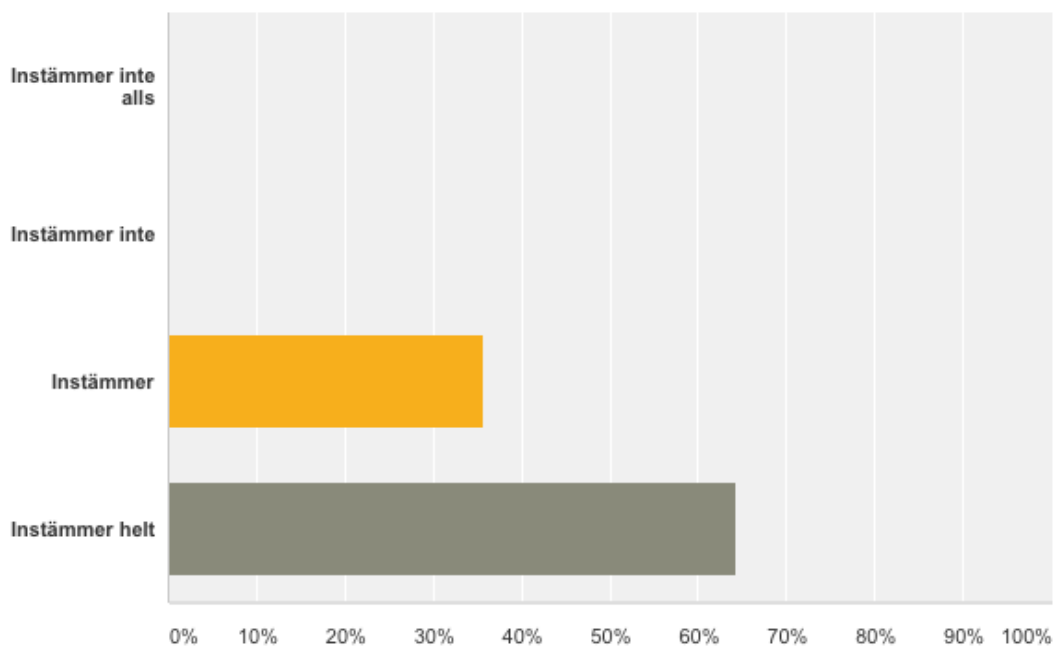
Svarade: 13 Hoppade över: 4



Svarsval	Svar
▾ Bluefire Reader	69,23% 9
▾ Aldiko	23,08% 3
▾ Bokon	7,69% 1
▾ DL Reader	0,00% 0
▾ Annan app, Jag använder... <b>Svar</b>	30,77% 4
Totalt antal svarande: 13	

## Jag hade gärna haft en och samma app för att både kunna låna och läsa.

Svarade: 14 Hoppade över: 3



Svarsval	Svar
▼ Instämmer inte alls	0,00% 0
▼ Instämmer inte	0,00% 0
▼ Instämmer	35,71% 5
▼ Instämmer helt	64,29% 9
Totalt	14

## A.2 Feature list

The following table presents the features that should be included in a finished product. Due to the limited time of the project, some had to be set aside. The features were prioritized after the degrees below.

- 1: *Must have*
- 2: *Would love to have*
- 3: *Would like to have*
- 4: *If there is time*

<i>Name</i>	<i>Description</i>	<i>Priority</i>
<b>Find/Borrow view</b>		
Start page		
See events	The user should be able to see current events that takes place in the chosen library	3: Would like to have
Adapt categories	The user should be able to choose if the start page only shows books from the user's chosen categories	4: If there is time
See alerts	The user should be able to see urgent messages from the library e.g. if loans are unavailable at the moment	1: Must have
See Help page	The user should be able to get information about how to borrow books and other FAQ	1: Must have
See library name	The user should be able to see which library their in (Malmös or Stockholms etc)	1: Must have
Change library	The user should be able to change which library their in (Malmös or Stockholms etc)	2: Would love to have
Top lists		
See general top list	The user should be able to see a general top list for all the books in the library	1: Must have
See category top list	The user should be able to see a top list for all the books in a certain category	4: If there is time
See new books top list	The user should be able to see a top list for all the newly released books	1: Must have
Recommendations		
See library recommendations	The user should be able to see a list of recommended books made by the chosen library's librarians	4: If there is time
See friend recommendations	The user should be able to see a list of recommended books made by their friends	4: If there is time
See category recommendations	The user should be able to see a list of recommended books for a certain category, made by the chosen library's librarians	4: If there is time
See custom recommendations	The user should be able to see a list of recommended books made according to the users already read books	4: If there is time
Search for Books		
Search book title	The user should be able to search a book by its title	1: Must have
Search author	The user should be able to search a book by the author's name	1: Must have

Search key word	The user should be able to search a book by a keyword	3: Would like to have
Search publisher	The user should be able to search a book by the book's publisher name	3: Would like to have
Sort books	<b>Sort in BOTH directions...</b>	
Sort titles	The user should be able to sort the books by their title	1: Must have
Sort authors	The user should be able to sort the books by their authors name	1: Must have
Sort publishers	The user should be able to sort the books by their publisher	3: Would like to have
Sort dates	The user should be able to sort the books by the book's publication date	2: Would love to have
Sort popularity	The user should be able to sort the books by their popularity	3: Would like to have
Sort rating	The user should be able to sort the books by their rating	3: Would like to have
Filter books		
Filter language	The user should be able to filter books by language, so that only books by a certain language(s) is shown	3: Would like to have
Filter publisher	The user should be able to filter books by publisher, so that only books by a certain publisher(s) is shown	3: Would like to have
Filter categories	The user should be able to filter books by their category	2: Would love to have
Lending		
Borrow book	The user should be able to choose a book to borrow	1: Must have
Enter pin code	The user should be able to enter their library cards pin code, to be able to borrow a book	1: Must have
Remember pin code	The app should be able to save and remember a user's pin code	2: Would love to have
See availability	The user should be able to see if a book is available for borrowing or not	1: Must have
See unavailability reason	The user should be able to see a reason why a certain book is not available	3: Would like to have
Queue books	The user should be able to queue a book that is unavailable at the moment	2: Would love to have
Return book	The user should be able to return a book that he/she do not want to read/borrow anymore	3: Would like to have
Book Information		
See title	The user should be able to the book's title	1: Must have
See author(s)	The user should be able to see who wrote the book	1: Must have
See cover	The user should be able to see a picture of the cover of the book	3: Would like to have
Read description	The user should be able to read a description of the book's story/plot	1: Must have
See nbr of pages	The user should be able to see how many pages a book has	2: Would love to have
See ISBN	The user should be able to see the book's ISBN number	1: Must have
See publication date	The user should be able to see the date the book was published	2: Would love to have
See translator	The user should be able to see who translated the book, IF the book was translated	2: Would love to have
See original title	The user should be able to see book's original title, IF the book was translated	3: Would like to have
See publisher	The user should be able to see the publisher of the book	1: Must have



See language	The user should be able to see the language the book is written in	1: Must have
See categories	The user should be able to see which categories the book belongs to	1: Must have
See series order	The user should be able to see what order in a series a book has, if it is part of a series	4: If there is time
See series	The user should be able to see the other books in the series that it belongs to, IF it belongs to any	4: If there is time
Read sample	The user should be able to read a small part of the book before deciding to borrow it	4: If there is time
Add to TBR list	The user should be able to add a book to the TBR-list that the user wants to read in the future	3: Would like to have
See borrowing period	The user should be able to see how long a book can be borrowed	3: Would like to have
See similar books	The user should be able to see a list with similar books	4: If there is time
<b>Rating</b>		
Rate books	The user should be able to give a book a rating	4: If there is time
Write comment	The user should be able to give a short comment/review about a book	4: If there is time
See blog reviews	The user should be able to see links to blog posts with reviews about a certain book	4: If there is time
<b>Read view</b>		
<b>Overviews</b>		
See chapters	The user should be able to see an overview of all of the book's chapters and page numbers for each chapter	1: Must have
See position	The user should be able to see which chapter they are currently at	2: Would love to have
Change chapter	The user should be able to change which chapter they want to read	1: Must have
See percentage	The user should be able to see how much percentage of the book the user has already read	3: Would like to have
See reading rate	The user should be able to see how fast he/she is reading	4: If there is time
See Help page	The user should be able to get information about how to borrow books and other FAQ	1: Must have
<b>Auto synchronization</b>		
Sync books	The app should be able to synchronize the user's borrowed books in the library on all their connected devices	3: Would like to have
Sync bookmarks	The app should be able to synchronize the user's bookmarks on all their connected devices	2: Would love to have
Sync settings	The app should be able to synchronize the user's settings on all their connected devices	3: Would like to have
<b>Usage/Utilization</b>		
Read book	The user should be able to read a book in their own book case	1: Must have
Translate text	The user should be able to translate a marked word/sentence with e.g. Google Translate	4: If there is time
Save quote	The user should be able to save a marked quote	4: If there is time
Share quotes	The user should be able to share a saved quote to a friend (with facebook, twitter etc)	4: If there is time
Open link	The user should be able to open links found on book pages	4: If there is time
Write comment	The user should be able to write and save a comment about a certain part of the book they are reading	4: If there is time

Share comment	The user should be able to share a saved comment to a friend (with facebook, twitter etc)	4: If there is time
See reader activity	The user should be able to see how many people are reading the same book, and their comments about certain parts of the books	4: If there is time
<b>Settings</b>		
Change brightness	The user should be able to adjust the brightness of the book pages	1: Must have
Change font size	The user should be able to change the size of the text in the book	1: Must have
Change font	The user should be able to change the font for the text in the book	4: If there is time
Change font color	The user should be able to change the color for the text in the book	4: If there is time
Change background color	The user should be able to change the background color for the view in the book	4: If there is time
Navigate pages	The user should be able to choose to navigate between pages vertical or horizontal with scroll	2: Would love to have
Highlight text	The user should be able to highlight text in different colors	4: If there is time
Change margins	The user should be able to change to different paddings/margins for the text in the book	3: Would like to have
Change line spacing	The user should be able to change the line spacing for the text in the book	3: Would like to have
Change mode	The user should be able to change between night and day mode	2: Would love to have
<b>My pages view</b>		
<b>My bookcase</b>		
See borrowed books	The user should be able to see which books the user are currently borrowing	1: Must have
See days left	The user should be able to see how many days are left until the loan of a certain book expires.	1: Must have
See books in grid	The user should be able to view the books in the bookcase in a grid view	2: Would love to have
See books in list	The user should be able to view the books in the bookcase in a list view	2: Would love to have
See Help page	The user should be able to get information about how to borrow books and other FAQ	1: Must have
<b>To Be Read-list</b>		
See TBR books	The user should be able to see a list of books marked as TBR	3: Would like to have
Remove TBR book	The user should be able to remove a book from the list if the user don't want to read that book any more	3: Would like to have
<b>Queued books</b>		
See queued books	The user should be able to see a list of books they are queueing for	2: Would love to have
<b>My friends</b>		
See friends	The user should be able to see a list with their friends	4: If there is time
See friends' pages	The user should be able to see parts of their friends' personal pages	4: If there is time
Recommend book	The user should be able to recommend a book to a friend	4: If there is time
Sync with other social accounts	The user should be able to share to Goodreads, Twitter, Facebook, Instagram	4: If there is time
Discuss books	The user should be able to discuss currently reading/already read books in a forum	4: If there is time

See challenges	The user should be able to see challenges the user is currently taking	4: If there is time
Add challenge	The user should be able to create a challenge (e.g. read x books, read book of a specific genre)	4: If there is time
Change challenge	The user should be able to change a created challenge	4: If there is time
Remove challenge	The user should be able to remove a challenge	4: If there is time
Challenge friends	The user should be able to ask friends to take part of a challenge	4: If there is time
Create Book Club		4: If there is time
Join Book Club		4: If there is time
Remove joined Book Club		4: If there is time

### A.3 The Mid-Fi design made in Balsamiq



Figure A.3.1 Start screen

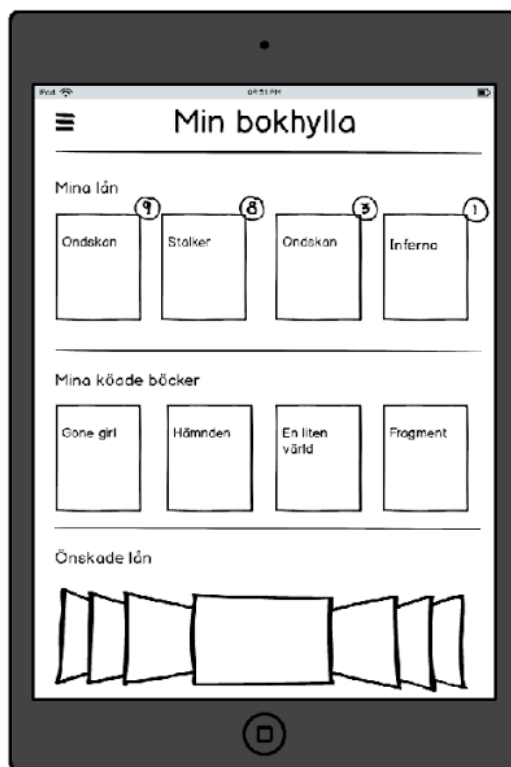


Figure A.3.2 My Bookshelf - loans, queue and to be read

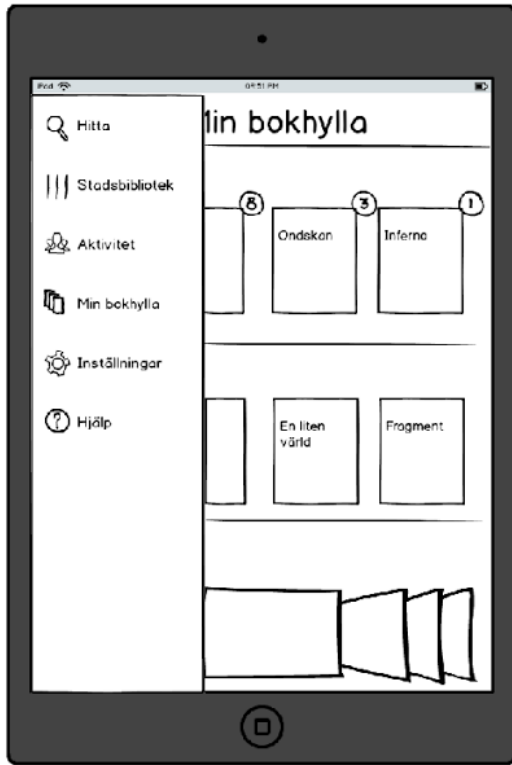


Figure A.3.3 Side menu

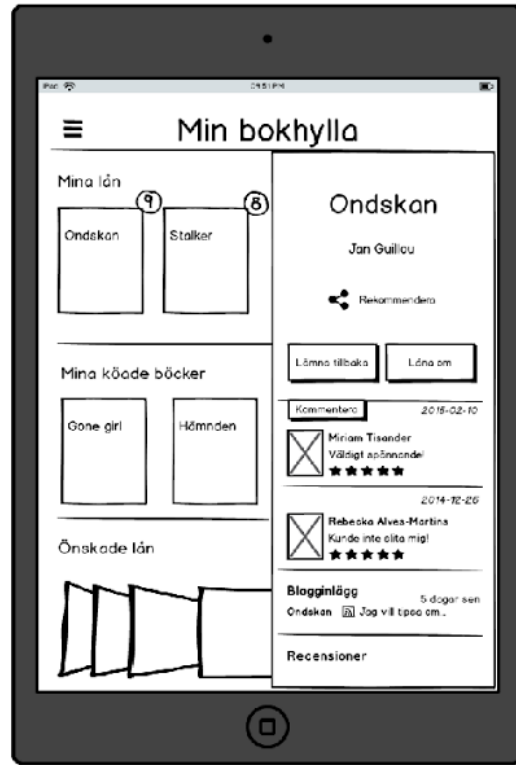


Figure A.3.4 My Bookshelf - book info

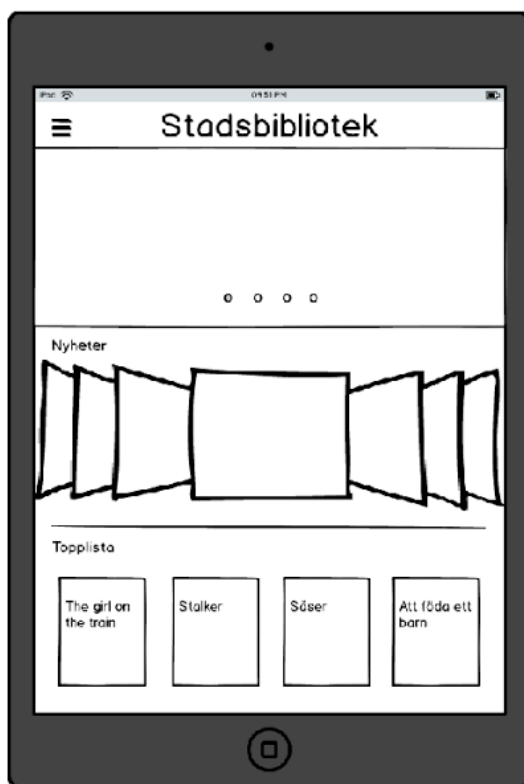


Figure A.3.5 Public Library – events, news, toplist

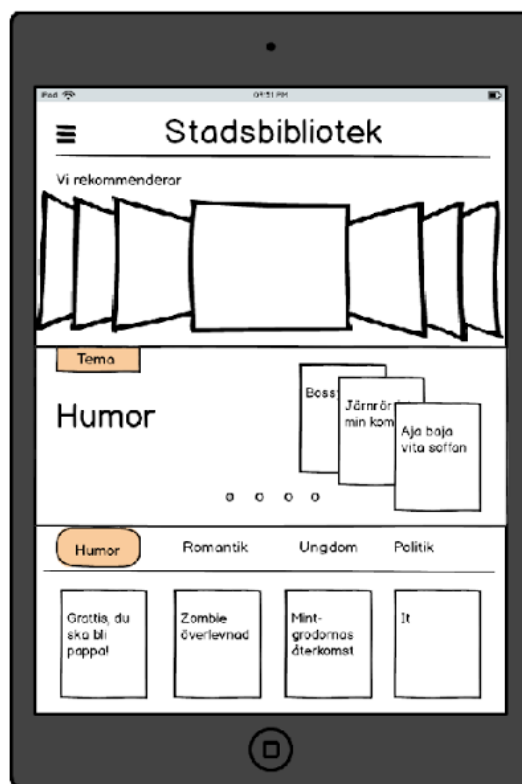


Figure A.3.6 Recommendations, themes

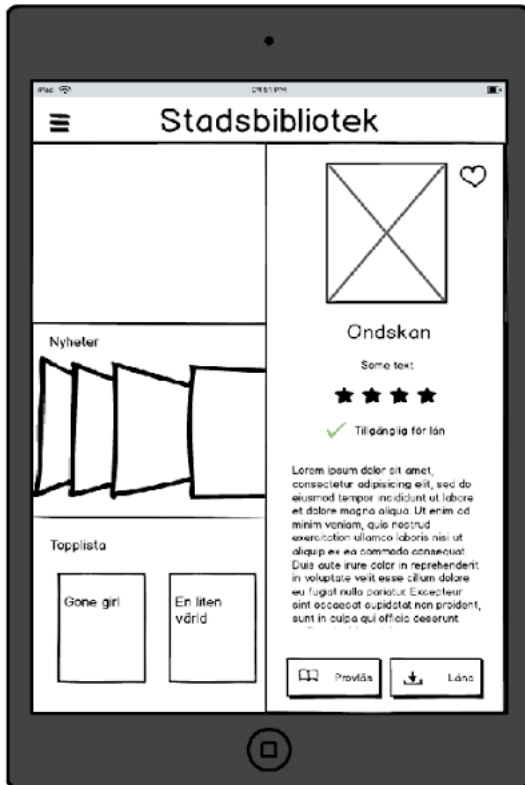


Figure A.3.7 Book info - available

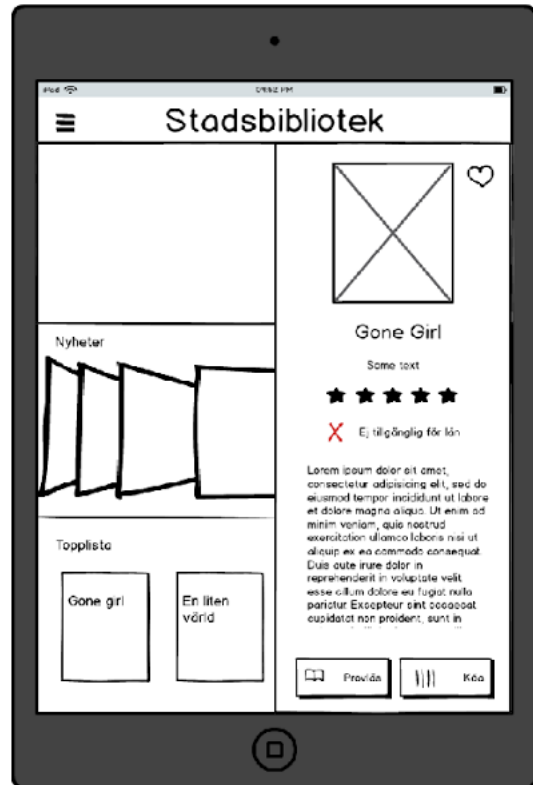


Figure A.3.8 Book info - not available

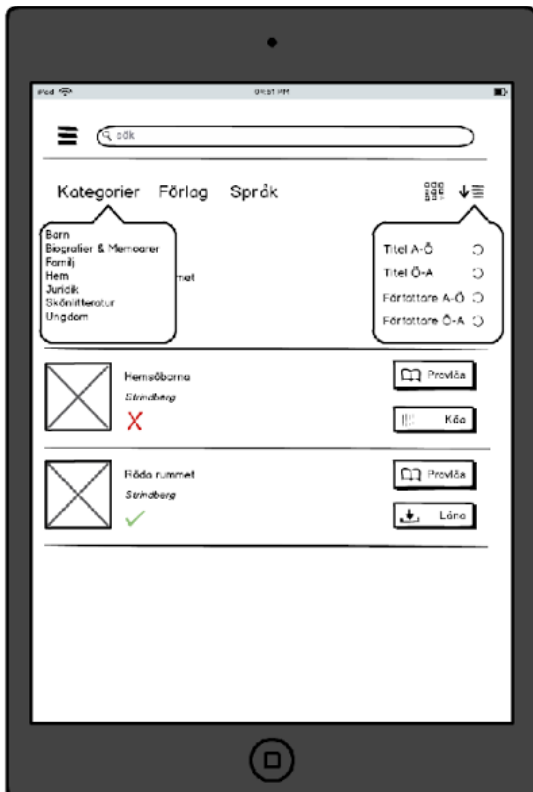


Figure A.3.9 Search view

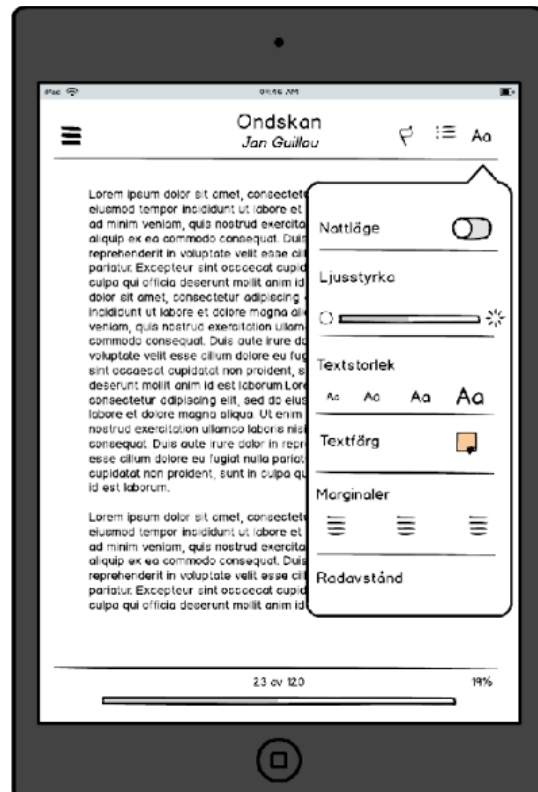


Figure A.3.10 Reading view

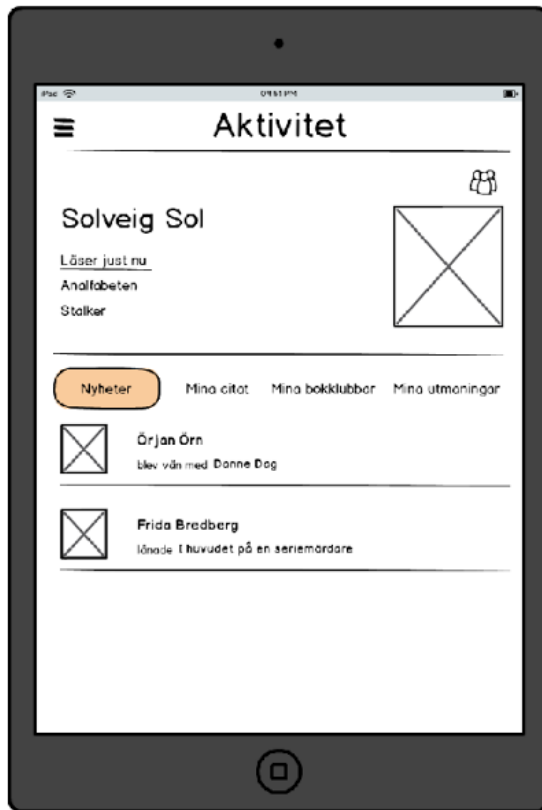


Figure A.3.11 Social activities

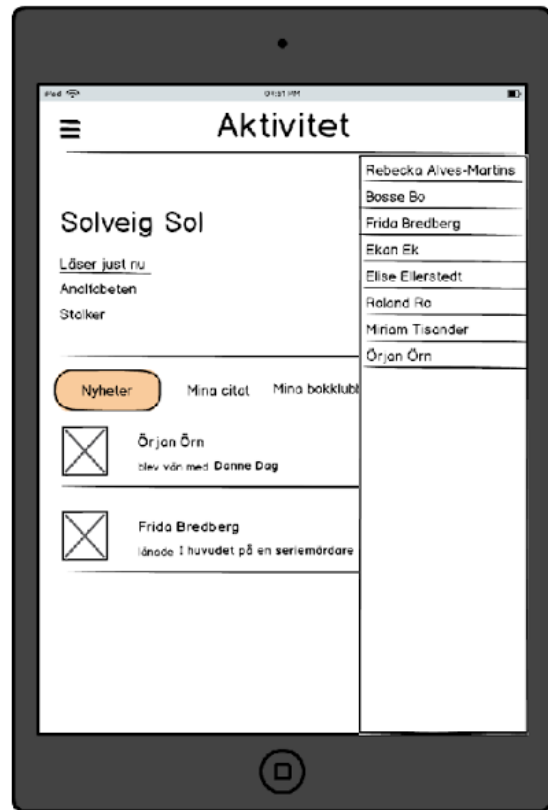


Figure A.3.12 Social activities – list of friends

## A.4 Final user study

### A.4.1 Part 1 - Analysis of the current system

Show current solution, so that the participants have something to compare with.

- Get a library card
- Get user account at Adobe
- Download a reader application
- Search for e-books on the webpage (through the web browser of the device)
- Download and open a book in the application

	YES	NO
Does the current solution seem complex?		
Does this intimidate you from trying the process of borrowing e-books?		



#### A.4.2 Part 2 - User scenarios

1. Borrow the book that is placed as number one in the toplist of the library.
2. You want to borrow a thriller, but you cannot remember the title, however you know that the author's last name starts with the letter 'Å'. How do you find it?
3. Sort your books after author A-Ö.
4. Reserve a book that is unavailable.

#### A.4.3 Part 3 - Post-test debriefing - Interview

- Was there anything that you felt was unclear? Was there anything missing?
- Do you believe that this solution can be as popular as Spotify or Netflix?
- In regards to how much it would cost to develop a complete solution, is it worth it compared to the current system?
- Would you consider borrowing e-books if this application was released?
- Is social media important to you?
- Does social features like book clubs, rating, reviews, sharing or challenges appeal to you?
- Would these features make you use the application more often?

#### Observation protocol:

<i>Buttons/Clickable items in the app</i>	YES	NO	Comments
Do they understand the <b>sort button</b> ?			
Do they understand the <b>grid/list button</b> ?			
Do they understand that they can click on the covers to get more info about the book?			
Do they understand the difference between the <b>"Borrow/reserve" buttons</b> ?			

<i>Search for books</i>	<b>YES</b>	<b>NO</b>	<b>Comments</b>
Do they understand that they can search for different things, not only title/author?			
<i>Malmö City Library</i>	<b>YES</b>	<b>NO</b>	<b>Comments</b>
Do they look for a search function in this view?			
Do they understand that a book can be borrowed?			
<i>My Bookshelf</i>	<b>YES</b>	<b>NO</b>	<b>Comments</b>
Do they understand that this is their own borrowed books and not the selection of the library?			

#### A.4.4 Part 4 - Survey

*How important are these features for your reading?*

	Very important	Important	Nice to have but not important	Not needed
Be able to change <b>brightness</b>				
Be able to choose <b>nightmode</b>				
Be able to change <b>margins/line spacing</b>				
Be able to change <b>colors</b> of text/background				
Be able to change <b>font</b>				
Be able to choose how I <b>navigate</b> (vertical or horizontal)				
Be able to see in <b>percent</b> how long I have read				
Be able to see how <b>fast</b> I am reading				

*How important are these social features to you?*

	Very important	Important	Nice to have but not important	Not needed
Be able to <b>rate</b> books				
Be able to read/write <b>reviews</b>				
Be able to <b>share</b> with friends				
Be able to join <b>book clubs</b>				
Be able to have <b>challenges</b>				