Ukrainian Catholic University

Bachelor Thesis

ShelfShare - A Salesforce-based Online Library for Sharing Printed Books

Author:
Anna Korabliova

Supervisor:
Mykhailo Morozov

*A thesis submitted in fulfillment of the requirements
for the degree of Bachelor of Science*

in the

Department of Computer Sciences and Information Technologies
Faculty of Applied Sciences

Lviv 2023

Declaration of Authorship

I, Anna KORABLIOVA, declare that this thesis titled, "ShelfShare - A Salesforce-based Online Library for Sharing Printed Books" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Якщо не можна, але хочеться, то можна. А нам дуже хочеться".

Yuriy Shherbyna

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

ShelfShare - A Salesforce-based Online Library for Sharing Printed Books

by Anna KORABLIOVA

Abstract

In today's world, despite the existence of electronic books, people feel the need for printed books, but there are often problems with their availability, accessibility, and cost. Therefore, we set ourselves the goal of solving this problem with the help of personal libraries of people who are ready to lend books to others and give readers the opportunity to find the desired books locally. For this, we decided to create an online library, or more precisely – a service for a direct book exchange.

In this thesis, we will describe the concept of this Online Library, provide details about the implementation and tools we used for the MVP, describe the user testing results, and outline future plans for improvements of our program.

Acknowledgements

Firstly, I want to thank my supervisor Mykhailo Morozov for his help, advice and huge support.

Also, I am very grateful to all the Applied Sciences Faculty of Ukrainian Catholic University teachers for their hard work and great patience.

Finally, I am very grateful to my family and friends who have supported me all these years. And I am incredibly thankful to my mother, whose teachings, kindness, love and immense support I will remember for all my life.

Contents

List of Figures

List of Abbreviations

API	**A**pplication **P**rogramming **I**nterface
CRM	**C**ustomer **R**elationship **M**anagement
LWC	**L**ightning **W**eb **C**omponents
MVP	**M**inimum **V**iable **P**roduct
SaaS	**S**oftware **a**s **a** **S**ervice
SOQL	**S**alesforce **O**bject **Q**uery **L**anguage
SOSL	**S**alesforce **O**bject **S**earch **L**anguage
SQL	**S**tructured **Q**uery **L**anguage

Dedicated to my Mom

Chapter 1

Introduction

1.1 Formulation of the problem and motivation

Every year, the world is actively developing, new technologies are being created, and databases with various collections are expanding. Many libraries and other institutions worldwide are digitising their archives and communicating through a common system with each other.

Let's outline the main problems and tasks we set for ourselves in this thesis. First, we're targeting people who prefer printed books over digital ones. Not everyone can read online for one reason or another, but many enjoy the texture of a book, the turning of pages, and the smell of a book, as strange as it may sound. Therefore, we would like to reproduce the experience of visiting a real offline library with our service. Our second goal is to enable not only libraries to share their collections but also ordinary book lovers. Our service allows users to lend books directly from their personal home collections and borrow books from other such users. It is meant to be a very quick process, as it happens directly through communication between two users. And the third reason, which can often happen to students during studies or in other situations, is when the needed book is nowhere to be found. For example, if the book of interest is no longer published, or there are no spare copies in the libraries, our service will allow to find and borrow that book from people who may have it at home. Additionally, this applies to situations where printing or buying new materials is expensive and harmful to the environment. Again, this is for those who prefer printed books or if its digitised version has yet to be created.

Therefore, our task includes researching available solutions for this problem in the market, assessing their benefits and limitations, and developing our own solution.

1.2 Work carried out

During work on this thesis, firstly, we researched the main topics, knowledge of which is required for a proper start of the development of our program. These topics included: familiarisation with our primary tool – Salesforce (and its advantages), and researching the book market and the opinion of the target audience (mainly university students) on printed and electronic materials.

The next step was the development and implementation of the online library MVP. In this thesis, we will go through all the stages of the work in detail: the kind of data model we have, the internal architecture of the program, and the reasoning behind it. Next, we will review the user experience aspects of our application. It is divided into two parts: borrowing books from other users and lending them to others. Among the implemented user flows, the following main ones can be distinguished: the book exchange, adding and editing a book that the user wants to lend to others, the process of user communication, and the search system.

Lastly, we will discuss the resilience of Salesforce and the testing of our system in terms of our set goals. In the end, all flows seemed easy for users, and all their comments and suggestions were analysed and included in the list of planned improvements.

1.3 Thesis structure

Chapter 2. What is Salesforce. This chapter introduces Salesforce, starting with a brief overview of its creation and what it is. Then, we will analyse its architecture and the advantages of such architecture. We will also discuss Salesforce from the developer's perspective, including the necessary technologies and programming languages needed for development on the platform. Lastly, the chapter summarises the general advantages and disadvantages of Salesforce.

Chapter 3. Printed and Electronic Study Materials. Students perspectives. This chapter discusses a study conducted by Boros University in Sweden that analysed university students' perspectives on the advantages and disadvantages of printed and electronic study materials. Additionally, we will analyse this study in the context of our project.

Chapter 4. Related Works. In this chapter, we will look at solutions with similar concepts to ours, analyse their position in the global market, discuss the difference between them and our project, and summarise the information obtained. Afterwards, we will address the issue of copyright that emerged while researching existing solutions in the market.

Chapter 5. Proposed Solution. This chapter presents the main technical details of Library MVP. It is divided into three parts, the first of which contains the construction of the data model, the second one is - the review of the project architecture from different aspects (communication with the database and web components) and the third - the user experience section where we will start with the general logic of each page of the application and go through the main user flows.

Chapter 6. Salesforce Resilience. This chapter describes Salesforce Resilience. It explores various technologies and approaches that Salesforce uses to ensure the reliability and accessibility of its services. Some technologies which are considered in the chapter are cloud data processing centres, data replication, scalability, load balancing, and backup data copying.

Chapter 7. User Testing. In this chapter, we review various surveys conducted during the development of the MVP, as well as the final testing and analysis of the results.

Chapter 8. Summary. This chapter summarises all the work done during this study, analyses the problems that our program will solve, and discusses future plans for this project.

Chapter 2

What is Salesforce

Salesforce is a cloud computing and powerful social enterprise SaaS provider [26]. Not only it performs CRM functionality such as storing customer data, managing customer interactions and relationships and etc, but it also offers a wide range of various tools and applications available for businesses. Started in March 1999 in San Francisco, Salesforce initially focused on providing a CRM platform but has since expanded to provide a broad range of solutions for sales, service, marketing, and analytics. Therefore, it is wrong to consider Salesforce as a simple CRM system.

Before Salesforce, most businesses struggled with various challenges when they needed CRM, such as high costs, spending a lot of time, complexity, and limited access to data. These challenges often led to poor user experiences and other related problems. Salesforce disrupted the market by providing a cloud-based solution that was easy to use, accessible from anywhere, and cost-effective.

Salesforce quickly gained popularity and became one of the top CRM solutions. It has over 150,000 companies worldwide, including many Fortune 500 companies[1].

2.1 Salesforce Architecture

Salesforce Architecture is designed to be flexible, scalable, and customisable to meet the needs of businesses of all sizes and industries. Let's go through layers of the architecture defined by Salesforce itself and imagine them as a delicious cake (advice from Trailhead). [21]

Cloud company. Salesforce is a cloud-based CRM platform that runs entirely on the cloud. Salesforce prioritises trust and takes numerous measures to provide data security and privacy. The company also provides regular transparency reports on government requests for customer data and other security-related issues. A special site, trust site [22], shows performance data, information about data storage and planned Salesforce maintenance.

Multitenant Cloud. To describe this feature of Salesforce, it is worth imagining an apartment building where each family has its own space (apartment), but all residents use the same resources (elevator, water, gas, etc.). Similarly, Salesforce provides the same set of services to all its customers in the multitenant cloud, such as keeping their data separate and secure, computing power and core features, regardless of project size or resource sharing between customers. This allows Salesforce to offer its services at a lower cost than traditional on-premise CRM systems, as customers do not have to invest in expensive hardware or software licenses.

[1] Fortune 500 companies - Fortune Magazine compiles a list of the 500 largest corporations from the United States annually.

Metadata. In Salesforce, metadata refers to data that describes different page layouts, objects, fields, components and other project customisation. Thus, for example, the program determines by objects and fields metadata how data (number, geolocation) is output, visualised and stored. Or another example: indicating in the components' metadata that they can be inserted into specific program pages (App Page, Home Page, Record Page). If the corresponding mark is absent, the component will not be available for the editing page in App Builder [20]. Metadata allows for controlled and efficient customisation, deployment of changes and a fast Salesforce Platform.

API. Salesforce's architecture includes a variety of APIs that allow developers to build custom applications and integrations on top of the platform and to connect to other applications or software. These APIs allow developers to access and manipulate data stored in Salesforce from external applications and facilitate communication with the database. When we create objects and fields with a few clicks on Salesforce, the API name field is automatically filled, which we can change if necessary. Using this API name, we can access the objects and fields we need from the code or when using various Salesforce features (field with type Formula, Email Template, etc.). APIs give Salesforce flexibility, making creating more diverse and unique applications possible.

2.2 Salesforce from the developer's side

Let's start with a few benefits of Salesforce from the developer side:

- **Easy Customisation.** Salesforce provides a drag-and-drop interface for customisation that is user-friendly and efficient. Usually, many project stages do not require programmatic development (no-code or declarative development) but only a few clicks. At the same time, when we create an object or field, new app or page, the platform does a lot of work under the hood. In this way, even without knowing any programming language, users can create their programs, which is an excellent advantage for people who do not have programming experience.

- **Robust Security.** Salesforce has robust security features, including granular access control, secure APIs, and encryption. This ensures the safety and privacy of user data.

- **Scalability.** The platform is built to handle large amounts of data and traffic, making it scalable for businesses of all sizes and handling millions of users.

- **Community Support.** Salesforce has a large community of developers who share knowledge and best practices, making it easier for developers to get help and stay up-to-date with the latest developments. Salesforce Community includes various communities, such as the Salesforce Partner Community for partners, Trailblazer Community for customers and developers, and Industry-specific Communities for special industries. Each community has unique features and benefits but is part of the larger Salesforce ecosystem.

- **Trailhead.** Salesforce has its training platform - Trailhead. It is built as a gamified platform where you can immediately test the received material in practice, performing various tasks and earning points and badges.

If programmatic development is used for the project, the developer faces a new task - to study the various technologies and programming languages for developing on the platform. Here are a few of the most important ones. [26]

Apex. Apex is an object-oriented programming language developed by Salesforce itself. It's similar to Java in syntax and is used to write business logic and customisations on the platform. Apex runs directly on Salesforce servers and has security, scalability and reliability benefits. This programming language is easy to test and has a set of different layers you can create. Here are a few of them: triggers that are called before or after interaction with the database, controllers that are responsible for the interactions of external code with data (controller calls from js) or more complex actions, which include selectors, services etc., or, for example, selectors that are intended for direct interaction with the database. All of these layers intersect, call each other, and are responsible for managing flows and transactions on the Salesforce platform. We use Apex in our project, and in the section 5.2.1, we go into more detail about it.

Visualforce. Visualforce is a component-based markup language used to build reusable user interfaces on the Salesforce platform. It allows developers to build custom pages, forms, and views integrating with Salesforce data and functionality. Visualforce is built on HTML and can use JavaScript, CSS, and other web technologies to create responsive, dynamic UIs. We did not use Visualforce in this project.

Lightning. Lightning is a component-based framework for building responsive user interfaces on the Salesforce platform. It includes pre-built components and tools for designing and building custom components (LWC). Thanks to ready-made components, novice developers and even experienced developers spend less time developing their applications. We also use lightning in our program. But it is worth mentioning that Salesforce also has another version older than Lightning: Salesforce Classic. Classic interface is now being replaced by Lightning, a more modern version launched in 2014. Although both versions are currently supported, Classic will be replaced by Lightning completely over time.

SOQL and SOSL. SOQL and SOSL are query languages used to retrieve data from Salesforce databases. SOQL is similar to SQL and is used to query records from a single object, where developers can use various auxiliary clauses of filtering (WHERE), sorting (ORDER BY), or internal selectors to improve their queries. In our project, we use SOQL queries for retrieving related records, filtering and sorting the data as we need. SOSL, on the other hand, is used to perform text searches in Salesforce. SOSL constructs queries in such a way that it is possible to search in different objects or fields simultaneously. SOSL queries are written using the FIND statement and can include search terms, filters, and sorting options. We are not currently using SOSL in our project.

Such a variety of technologies, each with its advantages and intended for a specific part of the program, allows flexible program development. Suppose developers are not strong in learning new languages. In that case, they have an opportunity to try free courses for developers on Trailhead that will help them start with simple programs in practice.

2.3 Using Salesforce: Pros and Cons

And let's go through and summarise the more general pros and cons of Salesforce, some of which we've mentioned before. [26]

2.3.1 Benefits

Data Management. Salesforce offers a centralised platform that allows businesses to manage their customer data more efficiently, enabling easier access and analysis of this information. This can help businesses make better-informed decisions, save time, and improve performance.

Increased Efficiency. Salesforce helps businesses automate various processes, reducing the amount of manual work and increasing efficiency. Examples of this can be automated reports, a built-in calendar, and the use of many functions with a few clicks.

Scalability. Salesforce is a cloud-based solution which can scale with the business as it grows. This reduces the need for costly hardware upgrades.

Integration. Salesforce integration connects with other systems or applications to share data and automate processes. This is important because it allows businesses to streamline operations and improve efficiency. Some common examples of Salesforce integrations include integrating with marketing automation systems, accounting software, and customer service applications.

2.3.2 Drawbacks

High Price. Salesforce can be an expensive solution, especially for small businesses or startups. The cost of Salesforce is mainly attributed to its cloud-based infrastructure and the robust functionalities it provides. In addition, the licensing and subscription fees for Salesforce become more expensive with the project growing.

Dependence on the Internet. Salesforce is a cloud-based system which always requires a stable and reliable internet connection.

Limited Customisation. While Salesforce provides many features and capabilities out of the box, it may not meet every business's unique needs. Customisation can be expensive and difficult for some projects. Therefore, it is necessary to understand Salesforce and its capabilities well before choosing it.

2.4 Conclusions

In conclusion, Salesforce revolutionised the CRM market by providing a cloud-based solution that is easy to use, accessible from anywhere, and cost-effective. Despite its many benefits, businesses should consider the costs and potential challenges when deciding whether to use Salesforce. But considering the advantages, we still decided to try to create something new with the Salesforce platform. In this document, we will get acquainted with how we did it, why we made certain decisions and what we got as a result.

Chapter 3

Printed and Electronic Study Materials. Students perspectives

Technologies are developing; the book market is changing; electronic books and articles are increasingly replacing printed ones. Is it worth making a program designed for printed books? Will people need it? In this chapter, we will consider the study of Boros University in Sweden, "Advantages and disadvantages of printed and electronic study material: perspectives of university students" [13].

3.1 The Boros University study

Briefly about the survey itself. The survey was conducted online among students of the University of Iceland and the University of Akureyri. The survey used The Academic Reading Questionnaire of 2014, previously built by Mizrachi [8]. The survey included questions about the person completing the survey (age, sex, student status, and field of study), 17 general questions about print and electronic materials, and one open question about academic reading format preferences. The study, mentioned in the previous paragraph, is based on the analysis of the last open question.

Why was this research chosen? Although the results of an open question may not be so accurate and the sample of students is insufficient, such results also provide an opportunity to obtain a more concrete answer from students to general questions. We plan to investigate the importance of books for a part of the target audience as students, how interested they are in printed materials and how accessible they are. In addition, this study also analyses the results of dozens of other studies from different universities and compares them with the existing results. So let's get back to the research itself.

Águsta Palsdottir, the author of this article, divided the research into several main themes and sub-themes, namely:

- printed or electronic study material (flexibility; printing electronic material);
- learning approach (the ability to concentrate on the reading; the ability to remember what was being read; length of the text matters);
- convenience and expenses (organising, approachability and volume; expenses);
- technology: limitations and possibilities (searching and browsing; making notes, scribbling and highlighting; technological advancement; physical reasons);
- environmental issues.

3.2 Analysis of the results of Boros University study

Students did not unanimously choose either only printed or only electronic materials. Quite often, they emphasised that they want to have a choice between printed and electronic and decide which suits them better. The choice between these types of materials was influenced by many factors, and let's analyse them in more detail:

- Length of the text matters and purpose. Printed material is usually preferred if the material is large or the goal is to prepare for exams. It is easier to read and faster to remember, reading this way is physically less tiring, and handwritten notes are more convenient than digital ones. Interestingly, technology is developing, but people consider it easier to do everything by hand than to use various technologies for material marking. The reason for reading electronic materials, which students mentioned, can be more superficial reading when there is no need to delve into the material. Let's go through our program for this situation. Printed materials can be borrowed for a certain period, but readers cannot write on them. Our app is for printed books, which are associated with easier reading. Although the program was designed only for books, users can still share other materials in our program, such as articles.

- Concentration while reading. Most students find that one of the biggest problems with reading electronic materials is that it is very easy to get distracted and very tempting to do something else. In addition, due to rapid physical fatigue, this temptation becomes even more significant.

- The electronic format is more convenient from the point of view of various technologies (easy access, search, copying and transfer of information).

- Electronic material is cheaper. It is worth noting here that we plan for our program to provide printed material for free in the manner of public libraries. Therefore, students will not have to spend money on printed materials.

- Ecological problems. Some students are concerned about the state of the environment nowadays, which is why they prefer electronic options. But if it is possible to borrow already printed books or other materials, then it will, on the contrary, help the environment. For example, in the first years of university study, many students print materials themselves. Since these materials are already printed and are in good condition, they can be safely passed on to the next generation of students without waiting for their return. Subsequent students, if necessary, can already mark by hand if they wish if they wish to not transfer these materials further. The same applies to various books used in education. Students can find electronic copies or can find a senior student who bought a printed version and does not mind lending it for their studies. And it is also important to note that, as the current study also mentioned, ecology is not only about trees and paper but also about the electricity spent reading e-books. If this energy is not produced from environmentally friendly renewable energy sources, then there is a question of which is more harmful. Therefore, from an ecological point of view, borrowing books from someone is a more environmental option than e-books or newly printed books.

3.3 Conclusions

Although the number of students was not big enough to tell about the results obtained, during the study, each of the results was similar to the other studies mentioned in the article, which mainly used quantitative research methods and had different samples of students interviewed. Therefore, the results of this study are mostly accurate.

Students are guided by many factors when choosing study materials. And although technologies are actively developing and students are well versed in them, more students still prefer printed copies in most cases. From the researched topics, we determined that the main reasons electronic materials are used, even though it is more convenient to use printed materials in this situation, are availability, costs and environmental problems, which some students drew attention to. This is one of the main goals that we are trying to solve in our program - to find materials near users that are missing or in short supply in the nearest libraries, to borrow them as in a library and a new problem that we can solve - environmental, regarding the number of printed materials and the use of electricity when reading electronic ones.

Chapter 4

Related Works

Now, we will review the existing solutions on the market and analyse their main goals and concepts used in them. The work presented in this document is based on similar ideas, so it will be essential for us to generally outline the goals and ideas of our work and its importance with the help of the reviewed solutions.

4.1 WorldCat.org

The following information was collected and analysed from official general information pages - WorldCat [25] and OCLC: WorldCat [11] information pages.

WorldCat [24] is an archive for many collections and aims to make it easy to browse libraries worldwide in one place with a single search window. In addition, to motivate librarians to become community members, it also has such essential functions as cataloguing and research, sharing library metadata and bibliographic records, and the ability to copy materials, which saves about 10 minutes per document.

This archive is the largest public catalogue in the world, which gathers the collections of thousands of institutions worldwide. As of December 2022, the catalogue had over 3 billion assets, including over 550 million bibliographic records in 483 languages. In this catalogue, you can find many kinds of printed and digital content, research materials, articles with references to full texts, historically important documents to which direct access is prohibited, etc.

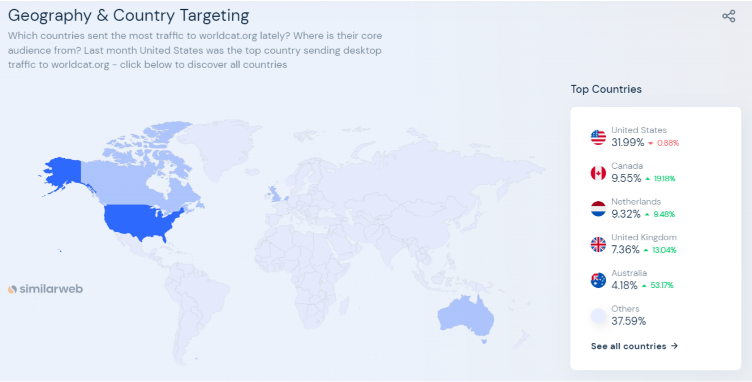
WorldCat is maintained by the OCLC [10] global cooperative and thousands of its member libraries. WorldCat itself is a free catalogue, but many OCLC services that use it are paid.

Now let's turn to the catalogue's central concept, analyse it in the market, and identify common and distinctive features with the project described in this thesis. To analyse the catalogue for the global market, we will use SimilarWeb [16], which collects general statistics on the use of a certain site, such as geographic data, target audience, etc.

4.1.1 The placement on the global market

WorldCat.org [24] is mainly distributed (Figure 4.1) among English-speaking countries, such as the United States (31.99%), Canada (9.55%), the Netherlands (9.32%), the United Kingdom (7.36%), and Australia (4.18%). WorldCat is still not well-known in Ukraine and is mainly used only by the largest libraries.

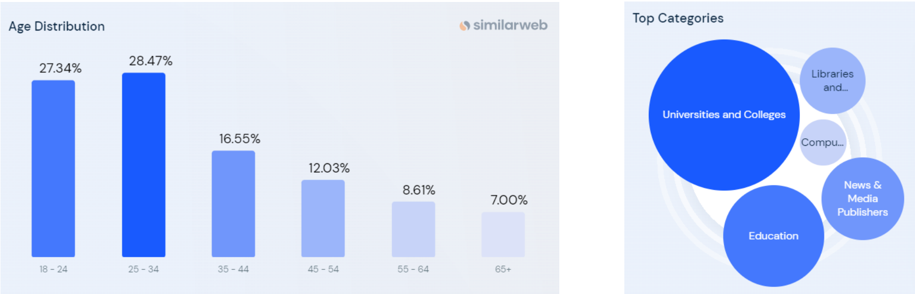
From the obtained statistics (Figures 4.2), it can be assumed that this catalogue is most used by students for educational purposes and by News & Media Publishers, which includes the distribution of books on various social networks.



Geography & Country Targeting

Which countries sent the most traffic to worldcat.org lately? Where is their core audience from? Last month United States was the top country sending desktop traffic to worldcat.org - click below to discover all countries

FIGURE 4.1: WorldCat Geography And Country Targeting [16]



(A) Age Distribution

(B) Top Categories

FIGURE 4.2: WorldCat Target Audience [16]

4.1.2 Generalisation and conclusions

WorldCat is one of the largest catalogues of libraries worldwide, but it does not allow you to distribute your books to ordinary users. It provides an opportunity to view digitised documents and find out where a physical copy of a particular document is located. Therefore, it gives all those who need the digital option and if the required book is available.

After analysing the concepts mentioned above, we can outline additional goals and problems, that our project can help to solve:

- the ability to distribute printed documents not only for libraries but also for ordinary users. This will allow to include a larger set of books in the exchange process, additionally benefiting for distribution of unique copies;
- a more localised book exchange would allow neighbours who have never met to exchange books. An additional advantage is socialisation through acquaintances, starting with an interest in similar books;
- popularisation the practice of reading and distributing books in Ukraine through the abovementioned socialisation;
- facilitating the finding of printed books unavailable in nearby libraries.

4.2 Open Library

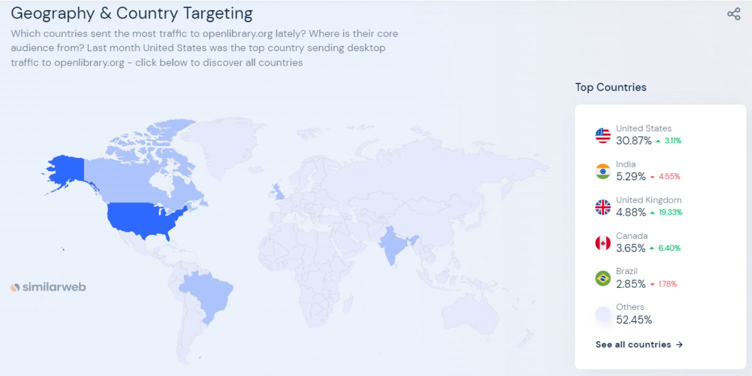
Open Library [5] is an online library that provides access to over 20 million digitised documents from 150 libraries and publishers [23]. This project was launched by the non-profit organisation Internet Archive [1], which houses collections of digitised books. Open Library aims "to create a web page for every book ever published." (Open Library [4]) Additionally, it cooperates with the previously analysed WorldCat, namely, references to relevant books in WorldCat, so that users can take the physical copies to read in the nearest library. [4]

Open Library has many advantages and disadvantages but is still actively developing and expanding. They also had legal problems with copyright, but we'll look at that in the next section 4.3, and now let's get back to the library itself.

Among the disadvantages of the project itself, which are worth noting, are: their audio reader uses a voice, which was generated automatically and can sound too robotic and several problems related to web development. The project is actively developing and has many advantages that can be taken into account. They provide users with statistics about the books they have read or are currently reading. They also allow keeping a list of books the user would like to read later, notes, and reviews of books.

In the next step, we'll review the placement of Open Library on the global market (SimilarWeb [15] was used to analyse the market).

4.2.1 The placement on the global market



Geography & Country Targeting

Which countries sent the most traffic to openlibrary.org lately? Where is their core audience from? Last month United States was the top country sending desktop traffic to openlibrary.org - click below to discover all countries

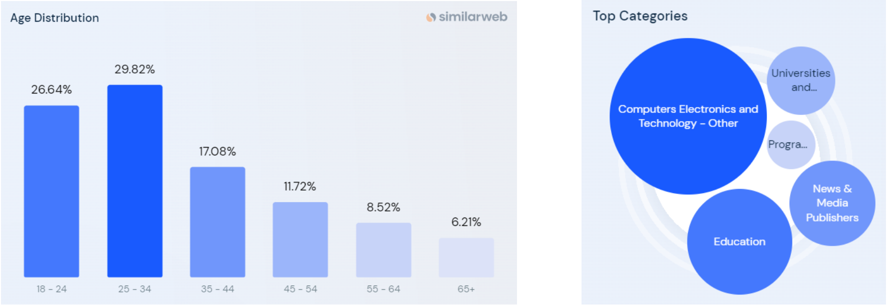
Top Countries

United States	30.87%	▲ 3.11%
India	5.29%	▼ 4.55%
United Kingdom	4.88%	▲ 19.33%
Canada	3.65%	▲ 6.40%
Brazil	2.85%	▼ 1.78%
Others	52.45%	

See all countries →

FIGURE 4.3: Open Library Geography And Country Targeting [15]

The situation differs from WorldCat among geographic indicators (Figure 4.3). The United States settled first place with 30.87%, followed by India with 5.29%, followed again by Great Britain and Canada, and Brazil closes the top five countries with the most extensive distribution. In Ukraine, like WorldCat, this library is not well known. Young people under 34 form the general active audience (Figure 4.4).



(A) Age Distribution

(B) Top Categories

FIGURE 4.4: Open Library Target Audience [15]

4.2.2 Generalisation and conclusions

Open Library is quite a convenient service for reading books, but again, they do not allow ordinary users to lend their books. Instead, they offer those to sponsor books and digitise them.

The organisation of a convenient service for users and data statistics is an essential point for our work. It can be noted that since our project will be focused on ordinary people and the exchange of physical documents, it is necessary to provide a way to analyse the reader or owner of the book by reviews and statistics about previously read books.

4.3 Copyright

In the previous section 4.2, we analysed the online library OpenLibrary [5] and mentioned that it had legal problems with copyright. To avoid legal complications and ensure compliance with copyright laws, we started researching the legal framework and rights associated with copyright at this stage.

Copyright is a set of laws that grants creators of original works, such as authors, artists, and musicians, with exclusive rights. Unauthorised use of these works is considered copyright infringement, which includes actions like copying, distributing, and publicly displaying the work without permission from the owner. [12]

Our platform, as well as OpenLibrary, which includes sharing relevant copyrighted content on the Internet, will, one way or another, face copyright issues. But here, it is also worth considering that after buying a book, the right to dispose of it belongs to the owner of the physical copy of the book. He can safely lend his copy to anyone for a certain period.

On the other hand, the book lending system is not about lending your book to a friend or neighbour. In this situation, our program acts as an online library analogue - lending a book to one person for a certain period. In many countries, copyright laws contain exceptions for public libraries.

Therefore, we can summarise several essential points that we should pay attention to in future research on copyright:

- our system acts as a public library. It is worth checking what laws must follow in this case and whether we can fall into such a category;
- taking into account the requirements of authors and publishers regarding restrictions on the use of their works. For example, OpenLibrary has a limit on books for you in the process of reading and a limit on users who can read a digitised book. In our case, there may be a limit on the number of copies of one book in one city because, at the expense of one document, we already follow the rule of reading a book by only one person during a particular time;
- how will the copyright be affected by the fact that the book is the property of the person who bought it and can lend it to anyone?

Since this topic has many ramifications and exceptions, it is worth contacting a specialist under any circumstances in the future.

Chapter 5

Proposed Solution

Before any project is developed, we should firstly create plans (i.e., a project work plan, a development plan, outline first important decisions, etc). We already have a rough idea of what we want to do, got acquainted with Salesforce, and researched the market for existing solutions and the target audience's attitude to printed and electronic books; but it is still too early to move on to the actual implementation.

In this chapter, we will go through the main stages of development, sorting them out to make it easier to familiarise ourselves with the project. The first stage is the Data Model, where we will outline the structure of our database (section 5.1). Next is the architecture of the program. This section includes communication with the database (section 5.2.1) and the division of the web part into components (section 5.2.2). After familiarising ourselves with the internal part of our application, we will proceed to the User Experience section (section 5.3), where we will start with the general logic of each page of the application and go through the main user flows.

5.1 Data Model

Salesforce has its specific data structure, so first, let's take a quick walk through the basic principles of the Salesforce Database.

5.1.1 Salesforce Database Overview

In Salesforce, database tables are considered separate objects. Each object (specific table) contains fields (columns) and records (rows).

Objects have different types. In this project, we will use two of them - Standard Objects (included in Salesforce) and Custom Objects marked with '__c' (that we created).

Besides, the fields of both objects can also be either standard, which is automatically added when the object is created (when and by whom it was created or modified, who is the owner, etc.), or custom, again with the mark '__c'.

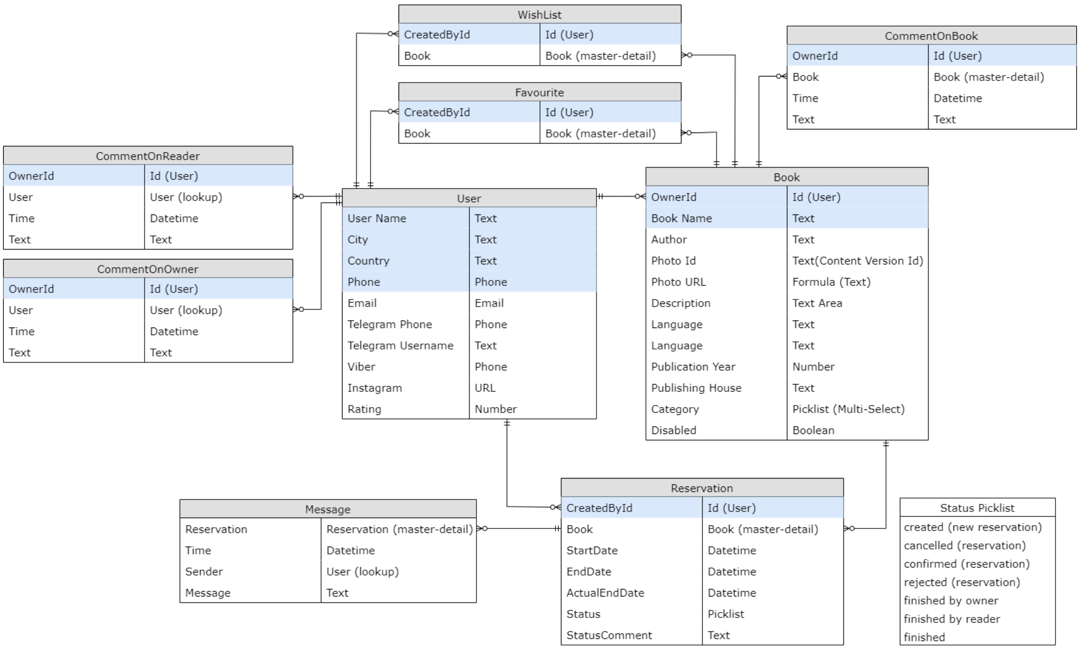
The last thing essential to know about the Salesforce database is Object Relationships, a field that connects two objects. The first field type we use is a lookup relationship. It is responsible for the connection between entities of the "one-to-one" or "one-to-many" type, and the importance of objects is equal. The second type is the "master-detail" relationship. Unlike the previous one, in the "master-detail", one object will be the main one. When a master is deleted, all dependent objects will be automatically deleted. Detail here does not work as a separate object and always has its own master. We do not encounter the third type of relationship in this work - Hierarchical relationships, which are available only for the User object and are often used to create control chains between different users.

You can find more information about Salesforce Objects and Fields in the Trailhead: Data Modelling. [19]

After reviewing the basics of a Salesforce database, we can move on to the data model itself.

5.1.2 Date Model Objects and Relationships

Important: The data model below contains object and field labels (without marks '__c'). The following description will use object and field names intended for programming with marks.



WishList	
CreatedById	Id (User)
Book	Book (master-detail)

Favourite	
CreatedById	Id (User)
Book	Book (master-detail)

CommentOnBook	
OwnerId	Id (User)
Book	Book (master-detail)
Time	Datetime
Text	Text

CommentOnReader	
OwnerId	Id (User)
User	User (lookup)
Time	Datetime
Text	Text

CommentOnOwner	
OwnerId	Id (User)
User	User (lookup)
Time	Datetime
Text	Text

User	
User Name	Text
City	Text
Country	Text
Phone	Phone
Email	Email
Telegram Phone	Phone
Telegram Username	Text
Viber	Phone
Instagram	URL
Rating	Number

Book	
OwnerId	Id (User)
Book Name	Text
Author	Text
Photo Id	Text(Content Version Id)
Photo URL	Formula (Text)
Description	Text Area
Language	Text
Language	Text
Publication Year	Number
Publishing House	Text
Category	Picklist (Multi-Select)
Disabled	Boolean

Message	
Reservation	Reservation (master-detail)
Time	Datetime
Sender	User (lookup)
Message	Text

Reservation	
CreatedById	Id (User)
Book	Book (master-detail)
StartDate	Datetime
EndDate	Datetime
ActualEndDate	Datetime
Status	Picklist
StatusComment	Text

Status Picklist
created (new reservation)
cancelled (reservation)
confirmed (reservation)
rejected (reservation)
finished by owner
finished by reader
finished

FIGURE 5.1: Library Data Model

5.1.2.1 User and Books Relationship

User (standard) - represents all users of our library. Every Salesforce user has this record. Therefore, we chose this object to define our users. Moreover, it contains personal information, various personal settings, and inbuilt Salesforce registration and login.

- Name (Text) - required user name.
- City and Country (Text) - required for further localisation of books in the project.

A set of contacts where the users can fill in any field, depending on which method of communication is more convenient for them:

- Phone (Phone).
- Email__c (Email).
- Telegram_Phone__c (Phone).
- Telegram_Username__c (Text).

- Viber__c (Phone).
- Instagram__c (URL).

The app will have the option to have a messenger for a specific book exchange right within the app. We want to allow users to share their contacts with another specific user, if needed, right in Messenger. This will protect them from spreading their contacts to everyone.

- Rating (Number) - calculated user rating according to his actions during each exchange.

Book__c (custom) - the minimum necessary bibliographic description of the book to familiarise the reader with it.

- OwnerId (User Id) - the users create the book, and only they can change it. This field is automatically generated.
- Name (Text) - required book name.
- Author__c (Text) - required author name.
- Photo_Id__c (Text) and Photo_URL__c (Formula Text) - when adding a photo, it is automatically recorded in the Content Document and other relevant fields (file storage in Salesforce). We read the photo location as Content Version Id and place it in the Photo_Id__c field to keep access to the photo. After that, the Photo_URL__c field is automatically calculated according to the formula we wrote. This way, we get the complete reference to the images that we can use to display them in the app.
- Description__c (Text Area) - required description.
- Language__c (Text), Publication_Year__c (Number), Publishing_House__c (Text) - additional fields for clarifying information which is not required. Language is an optional specification if the language of the book differs from the language of the title, etc. In the future, it is planned to implement a field with languages as a Picklist (Multi-Select), but for now, a simple text field is enough for MVP.
- Category__c (Multi-Select Picklist) - a field with categories (genres of books) that enable multiple selections from several options. Required - at least one item must be chosen.
- Disabled__c (Boolean) - indicates if the book is still available. If the user deleted the account or his book, and the Book__c record already has its history (list of who read it), we do not delete such records but block them with this field. For readers, disabled books will remain on the read list.

Wish_List__c and **Favorite__c** (custom) - objects that allow users to mark books. The Wish List is responsible for books the user wants to read, and the Favorite - is for books that have been liked. Users can use these lists for their own needs, if they wish.

- CreatedById (User Id) - a standard field that stores the Id of the user who made the record. The OwnerId field is missing from objects that have a master-detail relationship type.
- Book__c (Book master-detail) - link to the book the user marks. If the book is deleted, we can remove records from these Objects.

5.1.2.2 User and User Relationship

Reservation__c (custom) - the object responsible for the exchange of books.

- Book__c (Book master-detail) - binding to the book.
- CreatedById (User Id) - binding to the reader. A standard field that stores the Id of the user who made the record. The OwnerId field is missing from objects that have a master-detail relationship type.
- Start_Date__c (Datetime), End_Date__c (Datetime) - the specified date and time limits for which the book is taken.
- Actual_End_Date__c (Datetime) - the day when the book was returned.
- Status__c (Picklist) - the stage at which the exchange is now.
- StatusComment (Text) - a comment about the returned book. If the book was not returned, returned in terrible condition, or any other comment the book owner would like to add after the exchange is complete.

Message__c (custom) - an object connected to a specific Reservation record containing conversation between the owner and the reader regarding a particular exchange.

- Reservation__c (Reservation master-detail) - binding to a specific Reservation.
- Time__c (Datetime), Sender__c (User lookup) and Message__c (Text) - basic information about one message from correspondence.

Comments Objects (custom) - objects designed to store all kinds of comments. Comment_On_Book__c - comments for the book from the reader, Comment_On_Owner__c and Comment_On_Reader__c - comments about the owner and the reader, respectively, from the reader and the owner. We save the comment and the person who left it.

This is how our data model looks, necessary for the minimal functioning of our project and the implementation of its main ideas.

5.2 Architecture

During the development of web applications, various standard architecture layers are distinguished, such as the Presentation Layer (responsible for interaction with users through the User/Web Interface), Domain Layer (the main business logic of the app), Service Layer (coordinates the objects of the domain model) and Infrastructure (saving information in storage). This architecture has such benefits as:

- single responsibility principle (when the code is aimed at performing only a particular action);
- clear architecture;
- simple testing (to find any minor errors and debug the code);
- dynamic change of data storage logic.

This project has a similar structure. We will explore it more deeply.

In this section, we will look at the structure of our project from various aspects, such as the division into components, communication with the database and interaction between layers to interact with the database. Since we dealt with the data model in the previous subsection 5.1.1, we suggest starting with interaction with the database.

5.2.1 Interaction with the Database

Our interaction with the database consists of three levels (Selectors, Services and Controllers) that work together, each responsible for a specific set of functions. The main idea of such a division is to preserve the code's flexibility, comprehensibility and simplicity when the scale of the project increases. In other words, to ensure better code reliability.

5.2.1.1 Selectors

Select pattern is based on the Mapper pattern: "A layer of Mappers that moves data between objects and a database while keeping them independent of each other and the mapper itself." (Fowler, Martin)

The selector level is responsible for communication with the database through queries. In other words, we pass it the arguments the selector uses in the query, and the results are returned to us. In our project, we create separate selectors for each object and a set of particular methods in each selector containing a different query (Trailhead: Selector Layer Principles [17]).

5.2.1.2 Services

The service layer aims to encapsulate the code. As such, it acts as a set of fundamental object interactions that other layers already use — for example, adding, updating or deleting a new record (Trailhead: Service Layer Principles [18]).

5.2.1.3 Controllers

Controllers are the basis of user interaction with all system parts where data is processed. As soon as some event occurs that requires access to the database, the method of the corresponding controller is called, which in turn interacts with selectors or services.

5.2.1.4 Examples of interaction between layers

Let's highlight two main types of interactions available in our project (Figure 5.2).

The first is reading data. When we need some data, we turn to the controller, which passes arguments to the selector and returns the obtained result. Sometimes

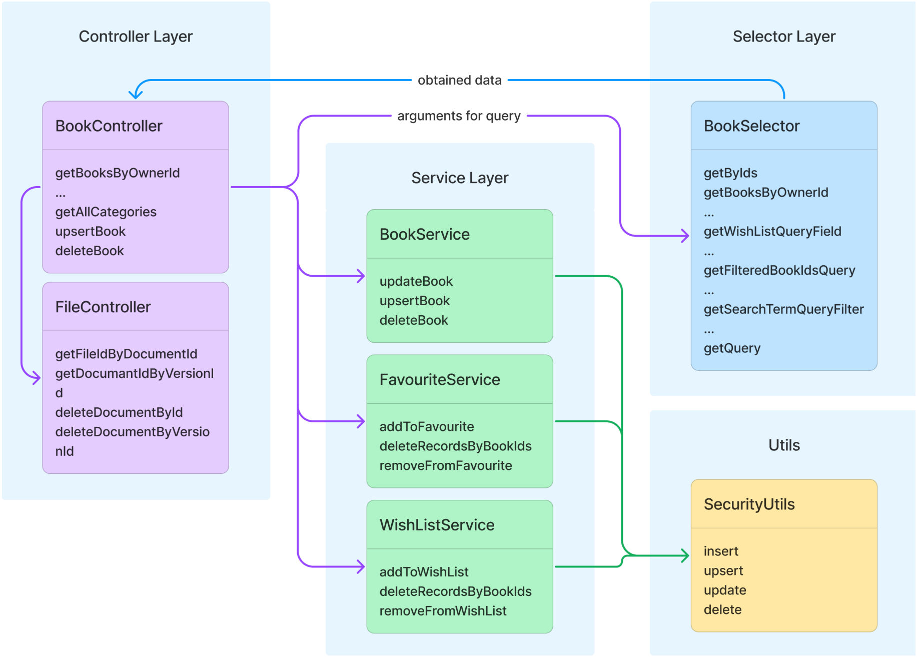


FIGURE 5.2: Interaction between layers from Book Controller

we don't even need to refer to the selector if we don't need to build SOQL queries. One such case is when we get a Multipicklist of categories.

The second type is actions like insert/update/upsert/delete the record. Here we again refer to the controller that calls the corresponding service. By the way, for greater security, we have created an additional Security Utils, which is responsible for any changes to the database. It checks if the data exists and can be added or removed from the database. If everything goes well, the service performs the appropriate operation.

5.2.2 Components (Presentation Layer)

At this stage, we will review the program's structure at the level of dividing the web part with which the user interacts into components. this part of the architecture is no less important than all the others.

If the code is split into the wrong components, we may face several bottlenecks or pitfalls in the future. For example, the following may occur:

- duplicate code in many places, leading to its accumulation and complexity. And when the project is expanding, it is vital to keep the code as optimised as possible in terms of the amount of code and interactions so that later one of the variables or a specific method does not have to be changed in the entire project in each place;
- code inconsistency (especially when developed by several people). The case when there is a method in one of the components that other developers do not know about, and everyone develops their approach. The code then becomes dirty and overgrows in the worst sense;
- unclear distribution. New developers or even the author will have to sweat after a particular time to understand what is happening and where the program crashes.

The component structure of our project follows two ideas: fewer duplicates and the most transparent and understandable code. Let's walk through it step by step (you can see the structure in Figure 5.3).

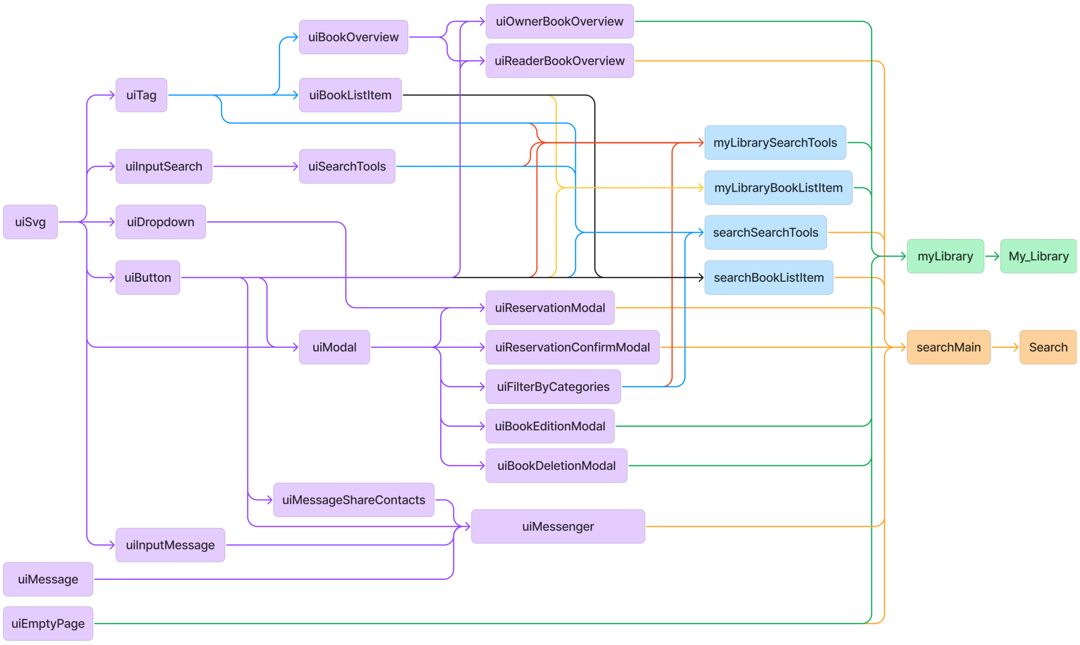


FIGURE 5.3: Components distribution for MyLibrary and Search pages

A page (green (myLibrary) or orange (searchMain) blocks) is a component responsible for collecting all the necessary components in one place. The HTML part is as simple as possible because it consists only of calling other components and putting them together. Let's look at the last components included in MyLibrary (green arrows). These are separate components with different structures and purposes that our main component places together on the page and is responsible for the interaction between them: uiOwnerBookOverview, myLibrarySearchTools, myLibraryBookListItem, uiBookEditionModal, uiBookDeletionModal and uiEmptyPage.

The page has three main components - a set of search tools, a book overview, and a set of components that lists all the books in order.

The search toolset (myLibrarySearchTools and searchSearchTools) consists of a special template (uiSearchTools) and slots inserted in the template. Since the pages have a similar structure and different actions, we must avoid code duplication. So each component refers to a common component where the main immutable content resides using slots.

The book list item (myLibraryBookListItem and searchBookListItem), similarly to the search tools, uses the uiBookListItem template, where the appropriate parts for a particular page are inserted using slots.

The book overview has similar logic to the previous components, but no separate component exists for the corresponding page. We only have a separation between the book view by the reader (uiReaderBookOverview) and the book view by the owner (uiOwnerBookOverview). Therefore we created two components which are similar templates created from the general template uiBookOverview.

While looking at the three main components, you'll notice that common components that act as templates have 'ui' in their name. These components are in a separate directory and are used in different places simultaneously. These can be basic components such as uiSvg, uiButton, uiModal (template for all modals that open over the page and have the appropriate structure) and more complex ones, such as various modals that use uiModal, uiMessenger and the previously mentioned uiOwnerBookOverview or uiReaderBookOverview. In the figure, such ui components are marked in purple.

My_Library and Search (the last blocks on the right) are custom pages that were created using the Lightning App Builder [20]. The Lightning pages we use are custom layouts that are composed of regions that, in turn, contain components. Such pages adapt to the device and change the positions of the regions accordingly. In our application, we use single-region pages where we insert our main components (myLibrary and searchMain). In the future, if necessary, we can easily change the set of regions and, with a few clicks, change the components.

5.3 User Experience

This section will give an overview of the primary user interactions with the site. Let's start with a general overview of page construction.

5.3.1 Application construction

The set of pages and the method of communication with the site are the basic things of user experience. Therefore, developing a structure where the user can go through the primary user flows with a minimum number of clicks is essential. Let's move on to our proposed solution for the distribution of the main logic elements of the site.

The library consists of five pages:

- My Library - users can view all the books they added, which they lend to others.
- Search - a page where users can view books available for reading.
- My Lists - two subpages with Favorites and books added to the Wish List.
- Borrowed Books - books that the user borrowed for reading. This page acts as a user-read book history.
- Lent Books - all exchange requests, past, present and future lending of books to someone. This is the history page of all books which users lend. This is where communication with readers is located and where a set of functionality for BookCrossing flow is provided (from the book owner side)

Each page has a similar structure: a search bar with filters, a list of books, and a book overview with more detailed information. The overview also contains reviews of the book, which previous readers left.

Reader pages (Search, My Lists, Borrowed Books) contain such additional functionality:

- reservation of books;
- reschedule a reservation;
- cancel a reservation;

- finish reservation;
- add/remove from favourites and wish list.

Book owner pages (My Library, Lent Books) include:

- book editing;
- deleting a book;
- transition to the history of the book (list of users who have read it).

My Library page also contains a button for adding new books. On all other pages, books with active or finished exchanges have a messenger for communication.

Now that we know the pages' basic structure, let's go through some of the main flows of user interactions with the site.

5.3.2 Book Exchange Flow

Book Exchange Flow (Figure 5.4) is one of our program's most important User Experience flows. It consists of several stages, so let's go through each of them step by step.

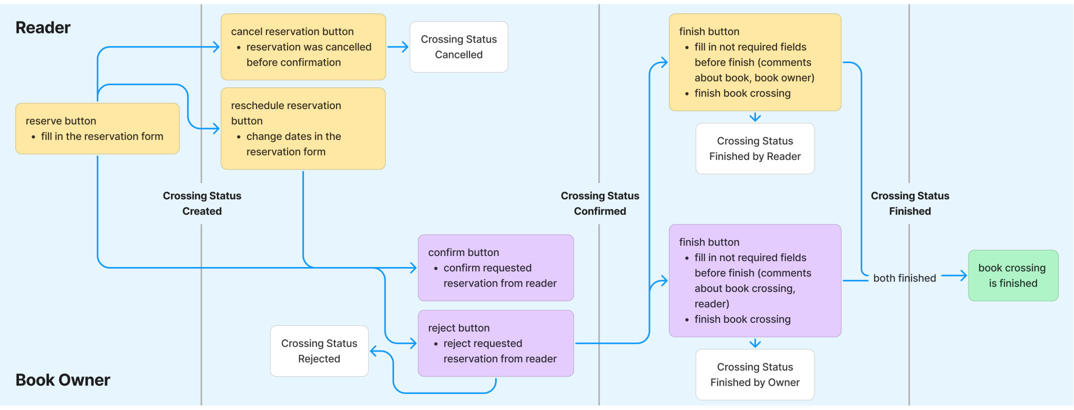
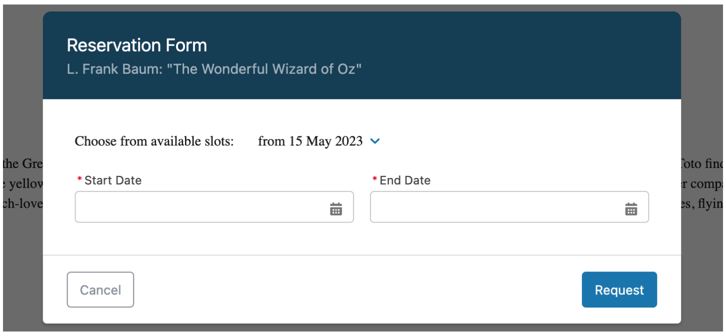

Reader
reserve button
• fill in the reservation form
cancel reservation button
• reservation was cancelled before confirmation
Crossing Status Cancelled
reschedule reservation button
• change dates in the reservation form
Crossing Status Created
confirm button
• confirm requested reservation from reader
reject button
• reject requested reservation from reader
Crossing Status Rejected
Crossing Status Confirmed
finish button
• fill in not required fields before finish (comments about book, book owner)
• finish book crossing
Crossing Status Finished by Reader
finish button
• fill in not required fields before finish (comments about book crossing, reader)
• finish book crossing
Crossing Status Finished by Owner
Crossing Status Finished
both finished
book crossing is finished
Book Owner

FIGURE 5.4: Book Exchange Flow

The first stage is the reservation. When the reader has chosen a book to exchange, he must fill out a small reservation form (Figure 5.5).


Reservation Form
L. Frank Baum: "The Wonderful Wizard of Oz"

Choose from available slots: from 15 May 2023

*Start Date

*End Date

Cancel

Request

FIGURE 5.5: Book Reservation Form

The user has to select from the available dates the beginning and end of his future exchange. Dates are filtered by already existing exchanges of this book that are confirmed and not yet finished. A Reservation record is created at the code level with the status ’created’. Next, we’ll use status instead of the Status field of the Reservation record.

When the readers make the reservation, they can contact the book owner through the built-in messenger. In addition, before the book owner confirms or rejects the reservation, the reader still has the option to reschedule and cancel the reservation. During cancelling, the status becomes cancelled, and the reservation is no longer valid.

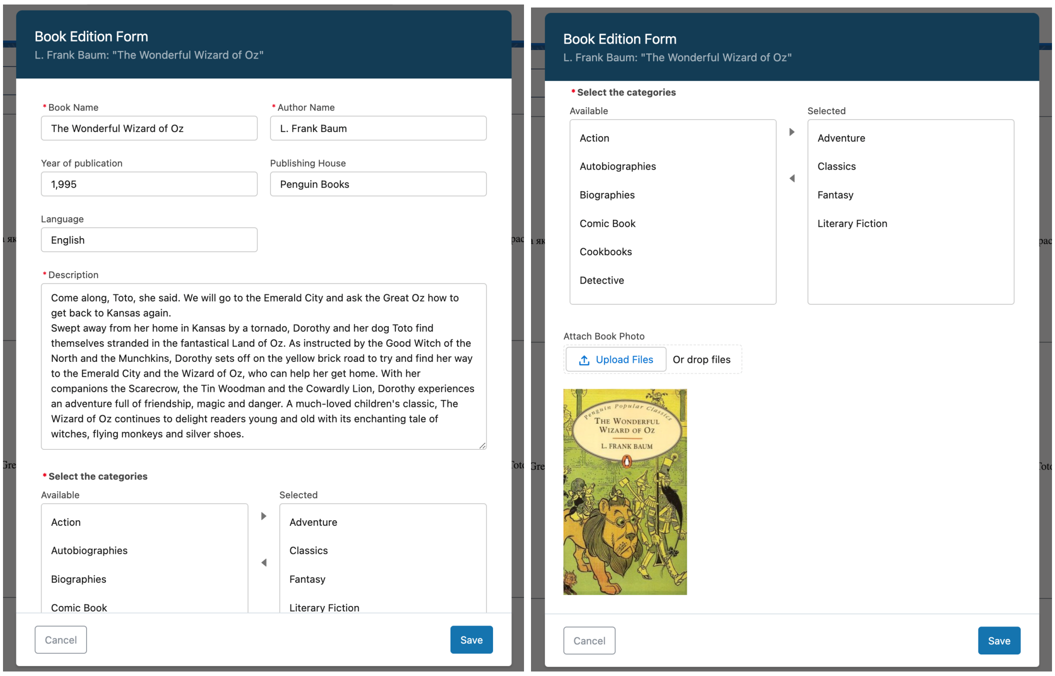
Let’s go to the book owner. Once a reservation has been created, it will wait for confirmation from the book owner. There is already a problem here because several people can request the book simultaneously. Then the book owner has to choose for whom to confirm and for whom to reject the reservation. To make this choice, the owner can communicate with readers in the messenger, agree on rescheduling the date by readers (reschedule reservation), or choose who will read the book without conversation.

It is worth noting that if the owner confirms one reservation, all other requests overlapping with this reservation will become unavailable until the dates are changed. As a result, the book owner can also choose the most convenient option for himself (how the exchange will take place, who is the reader of the book, and what dates are required). The status at this stage becomes confirmed or rejected (the exchange will not occur).

Suppose the reservation has been confirmed, then the owner of the book and the reader agree in a convenient way about the book crossing and everything else.

After the exchange is complete, both sides must confirm the finish. The reader and the book owner have to finish the exchange by pressing the finish button and, if desired, can fill in the fields for various comments in the form before finishing. As a result, after completion on both sides, the status becomes finished.

5.3.3 Creation and editing of the book



Book Edition Form
L. Frank Baum: "The Wonderful Wizard of Oz"

* Book Name: The Wonderful Wizard of Oz
* Author Name: L. Frank Baum
Year of publication: 1,995
Publishing House: Penguin Books
Language: English
* Description:
Come along, Toto, she said. We will go to the Emerald City and ask the Great Oz how to get back to Kansas again.
Swept away from her home in Kansas by a tornado, Dorothy and her dog Toto find themselves stranded in the fantastical Land of Oz. As instructed by the Good Witch of the North and the Munchkins, Dorothy sets off on the yellow brick road to try and find her way to the Emerald City and the Wizard of Oz, who can help her get home. With her companions the Scarecrow, the Tin Woodman and the Cowardly Lion, Dorothy experiences an adventure full of friendship, magic and danger. A much-loved children's classic, The Wizard of Oz continues to delight readers young and old with its enchanting tale of witches, flying monkeys and silver shoes.

* Select the categories

Available	Selected
Action	Adventure
Autobiographies	Classics
Biographies	Fantasy
Comic Book	Literary Fiction

Cancel Save

Book Edition Form
L. Frank Baum: "The Wonderful Wizard of Oz"

* Select the categories

Available	Selected
Action	Adventure
Autobiographies	Classics
Biographies	Fantasy
Comic Book	Literary Fiction
Cookbooks	
Detective	

Attach Book Photo
Upload Files Or drop files

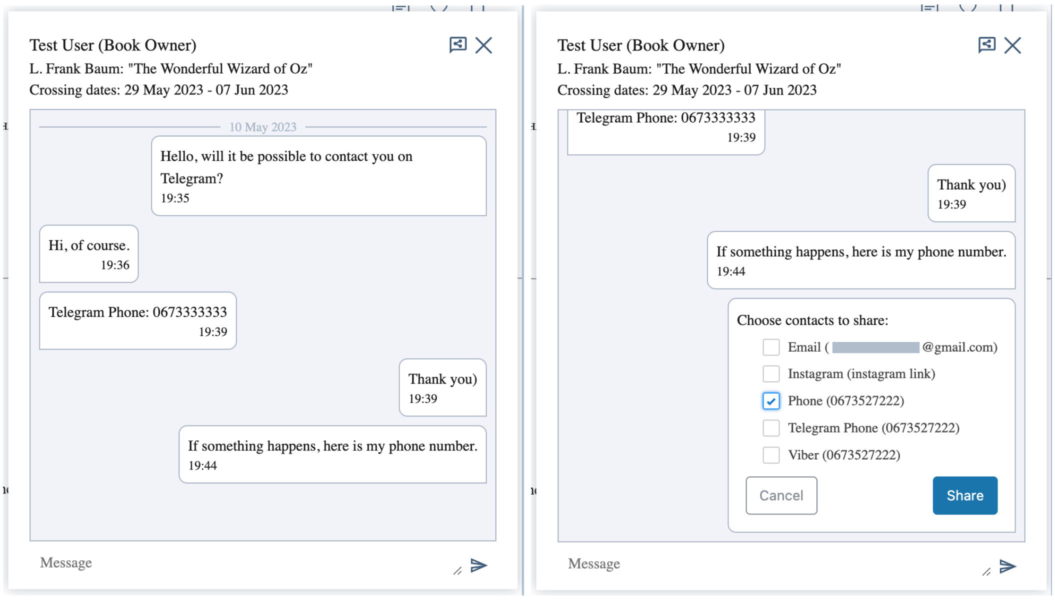
Cancel Save

FIGURE 5.6: Book Form

We use the Book Form (Figure 5.6) to create and edit a book. All fields are validated before submitting the form and only then sent. The image at the code level is required, and a separate logic is prescribed for it because it is stored separately from all other data. That is, when changing an image, the last version of the uploaded files is saved. If changes are cancelled, the uploaded new images are deleted.

5.3.4 Messenger

The messenger (Figure 5.7) has minimal functionality for communication. Users can share their contacts with the conversation partner if they want to continue communicating in a more convenient environment. To do this, they need to press the button to distribute contacts, and then a form with available contacts will appear in front of them (Figure 5.7b). The contacts selected by the user will be sent to the chat. To not show everyone user's personal data, it was decided to let the user choose who to show contacts to in this way. This made it possible to solve two problems. The first is using data for actions prohibited by law, and the second is possible legal problems for distributing personal information.



Test User (Book Owner)
L. Frank Baum: "The Wonderful Wizard of Oz"
Crossing dates: 29 May 2023 - 07 Jun 2023

10 May 2023

Hello, will it be possible to contact you on Telegram?
19:35

Hi, of course.
19:36

Telegram Phone: 0673333333
19:39

Thank you)
19:39

If something happens, here is my phone number.
19:44

(A) Messages and shared contacts

Test User (Book Owner)
L. Frank Baum: "The Wonderful Wizard of Oz"
Crossing dates: 29 May 2023 - 07 Jun 2023

Telegram Phone: 0673333333
19:39

Thank you)
19:39

If something happens, here is my phone number.
19:44

Choose contacts to share:
- □ Email ([redacted]@gmail.com)
- □ Instagram (instagram link)
- ☑ Phone (0673527222)
- □ Telegram Phone (0673527222)
- □ Viber (0673527222)

(B) Share Contacts Form

FIGURE 5.7: Messenger and Contacts Sharing

Since our site is specifically for book sharing, messages are attached to a particular reservation.

5.3.5 Search tools

Search tools consist of a search field and a set of filters. For MVP, we decided to search by book title or author (search all books in which the author or name fields contain input value). Filtering is based on book genres and exchange status. Search tools are available on each page with the corresponding functionality there.

Chapter 6

Salesforce Resilience

Earlier in the Chapter 2 "What is Salesforce", we learned what Salesforce is to have a more general idea about it. After familiarising ourselves with the proposed solution's structure, we suggest diving into the study of Salesforce resilience.

Salesforce ensures the reliability and accessibility of its services through various technologies and approaches. Let's take a closer look at how Salesforce ensures the resilience of its system.

Cloud data processing centres. Salesforce uses cloud data processing centres located in different regions worldwide to store its clients' data. This provides high data availability and processing speed.

The distributed network of global data processing centres. It is also worth noting that Salesforce has a distributed network of global data processing centres, which ensures high availability and speed of system operation regardless of user location. If one data processing centre goes down, other centres can continue to operate without disruptions.

Data replication. Data replication is one of the key technologies that Salesforce uses to ensure the resilience of its system. This technology involves creating copies of data and storing them in different places. Multiple identical and simultaneous Salesforce instances can improve data protection, reliability and improve network performance. This also enables hosting identical data in different locations, reducing network lag and improving user experience, system performance, and data security. So, if one of Salesforce's servers goes down and data is lost, the system automatically switches to another server. But it is very important to correctly define the data to be replicated, choose the right tool for replication and think through all the essential points like setting a disaster recovery strategy or testing. Data loss will lead to serious consequences such as refunds, destroyed reputation or, worst case, closing the business. [14]

Scalability. Scalability allows automatically adding more servers or instances to handle increased user traffic. This can be accomplished through horizontal scaling, which involves adding more servers to distribute the workload, which supports by load balancing and clusters of application servers. For example, Salesforce can automatically launch new application instances to handle additional traffic during peak times and then reduce their number during periods of decreased load. This ensures stable system operation even with changing workloads. In addition, Salesforce can easily handle the growing volume of data.

Load balancing. Load balancing means that the Salesforce system automatically distributes the workload across multiple servers or instances, ensuring optimal use of resources, avoiding overloading individual servers and increasing system productivity. This technology helps to improve application availability by distributing requests across multiple servers. Thus, the system will not be affected by the failure of one of the servers or maintenance, as well as the large load that will be distributed between the servers.

Backup data copying. Salesforce uses backup data copying to ensure security and reliability. This means that the system automatically creates copies of data and stores them in backup locations. If there is a problem with the primary copy of the data, the system can quickly switch to the backup copy, preventing data loss. This helps during user error, system failure, or malicious activity. Backups can be performed manually or automatically using various tools and can be stored on-premises or in the cloud. Backups are essential for data recovery and business continuity in case of unexpected events. [7]

Therefore, it can be concluded that Salesforce ensures the resilience of its system using various technologies, such as data replication, scaling, load balancing, and data backup. Additionally, Salesforce has a distributed network of global data processing centres. All these approaches and technologies allow Salesforce to provide high availability, reliability, and performance of its system, which is crucial for businesses and users.

Furthermore, when discussing the system's reliability, we cannot fail to mention the security part.

Security Measures. Salesforce implements various measures to protect user data and ensure system reliability. These measures include data encryption, access controls, intrusion detection and prevention, and regular security audits. Let's look at data encryption and access controls in more detail.

Security Measures. Data encryption. Data encryption is a security measure that is used to protect sensitive data by encoding it in a way that makes it unreadable to anyone without the proper decryption key. In Salesforce, encryption can be used to protect sensitive information, such as customer details, financial data, and other confidential information. Encryption is available for data at rest and in transit. By encrypting sensitive data, organisations can minimise the risk of data breaches, unauthorised access to confidential information, and other security threats. [3]

Security Measures. Access Controls. For projects just entering the market and not ready to use various paid features for security, Salesforce provides a standard set for logging authentication events. This set includes: [2]

- IP restrictions - restrict access to the system based on IP addresses. This helps to protect sensitive data from external threats.
- Login hours - automatically log out users from the system after a specified period of inactivity. This helps when a user forgets to log out. The administrator can change the session time limit if necessary.
- Two-factor authentication - providing two forms of identification to access the system.
- Device Activation - requires users to activate their device before accessing the system. This involves receiving an activation email or code and following the instructions to verify their device.
- Password policy - allows administrators to set rules and requirements for user passwords. This helps with requiring users to create strong and secure passwords.

Each login attempt is recorded in the history, which can be used to prevent unauthorised access to the system.

Besides own data centres and infrastructure, some Salesforce services are hosted on AWS (Salesforce data centre locations [6]).

Salesforce uses Amazon Web Services (AWS) to provide the infrastructure that is a key factor in the success of its resilient system. AWS is a leading cloud service provider that enables infrastructure scalability and reliability. [9]

Thanks to AWS, Salesforce ensures the flexibility and scalability of its infrastructure, allowing them to work with millions of users worldwide. Additionally, using AWS enables Salesforce to provide high availability and reliability of its services.

Overall, it can be said that Salesforce has quite impressive technological capabilities and uses them to ensure high availability and reliability of its system. Additionally, Salesforce is constantly working on improving and expanding its capabilities, which allows it to provide quality service and meet the needs of its users.

Chapter 7

User Testing

One of the most important stages of development is user testing. Although we did not focus heavily on the program's design, testing with the audience is a crucial part of creating it. Even for MVP, we needed to conduct small surveys among people. In this way, we adjusted the direction of development, which features are needed for the user, and which should be removed.

7.1 Surveys during program development

During one of the surveys, we discovered that users need the ability to manage multiple book lists. In other popular applications, users can mark items as favourites, save them for later, or add them to the cart. As a result of this experiment, we created two lists (Favorites and Wish Lists) where users can add books, and on the "My Lists" page, they can view these lists. For the minimal MVP, we chose to start with two default lists that users most commonly requested. The option for users to create their own lists was added to the list of planned improvements.

Another survey we conducted was related to the set of pages in the program. We wanted to know how interested users are in different variations of the program pages with books that are currently available or in the exchange process. The options were as follows:

- We have two main pages, "Borrowed Books" and "Lent Books", where active books can be obtained using filters.
- Create a separate page that only displays active book exchanges and has two subpages for borrowed and lent books. The separate page for read books by users history.

Results: most people are interested in fewer pages, so they chose the first option with two main pages.

Also, during the development of our MVP, we needed to choose a minimum list of questions to finish the book exchange. Here, the answers differed more. Some users wanted the option of not leaving comments at all, while others considered one comment on the book sufficient. Some preferred to rate the process with some stars, while others suggested a huge list of questions that should be asked. After carefully weighing all the answers, for the first variance of the Finish Form, we chose two optional comments from the reader and owner of the book. The reader can leave a comment on the book they have read and comment on the book owner if they wish. Similarly, the book owner can leave a comment about the reader and comment on how the book borrowing process went overall.

Now let's move on to testing our MVP.

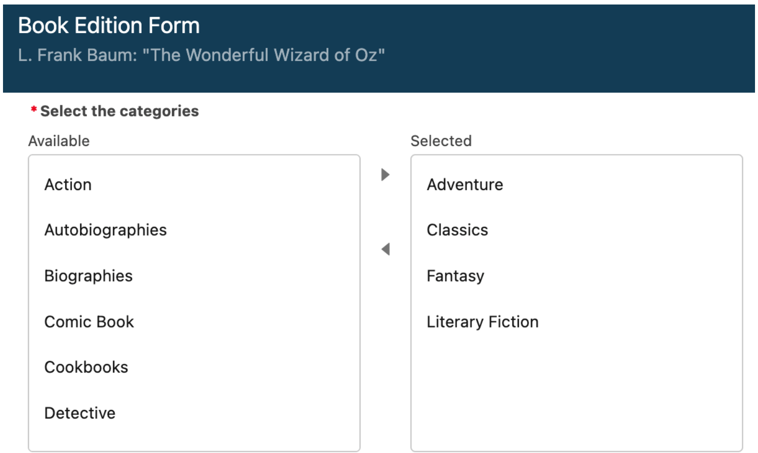
7.2 Testing MVP

Let's divide it into several parts.

Checking the primary user flows. To test, users were given a ready-made account with a few books already added. Users were given the following tasks:

- Add a new book to their library (a book that the user plans to lend to others).
- Edit a book.
- Delete one of the books from the library.
- View the history of one of the available books in the library.
- Choose a book to read and reserve it.
- Write a message to the book owner, and try to send their contact information for communication.
- Finish the book you are already reading.
- Approve or reject a request for your book (another user wants to read this user's book).
- Mark a book as a favourite or wishlist, and find where to view the marked books.
- Use filters and a search query to find a book of a particular category, exchange status, and with a specific word (either author or part of the book title).

Results: All tests were passed. Users achieved the set goal quite quickly. The difficulty users encountered was with the multi-selector. It is present in filters and the book creation form (Figure 7.1).



Book Edition Form
L. Frank Baum: "The Wonderful Wizard of Oz"

* Select the categories

Available	Selected
Action	Adventure
Autobiographies	Classics
Biographies	Fantasy
Comic Book	Literary Fiction
Cookbooks	
Detective	

FIGURE 7.1: Multi-selector in Book Form

The difficulty was that users tried to move an element from one column to another. They said they didn't know why it wasn't moving, and when they mastered the functionality, they said that it was not convenient to click.

User comments. This is one of the main parts of this testing because it gives us new perspectives on our program, which can be discussed, undergo additional testing, and ultimately, if decided, be implemented in our program.

Let's go over the main comments from this testing (issues):

- Uncomfortable interface in the multi-selector.
- Getting a notification about the deadline for reading a book several days before.
- All other notifications from the app - who the message came from, a request for a book (our MVP does not include a notification system, but this functionality is on the list of planned improvements).
- Inconvenient names (more related to design) - some users got lost among tabs and suggested that the pages should be named in other way.
- One user suggested modifying the messenger button to make it larger and more visible.

Overall impression. On the last question about what users think of the app and whether they would use it in the future, we mostly received affirmative answers. Some people who either prefer e-books or are neutral about whether the book is printed answered that they are not sure or definitely would not use the application. Those who prefer printed books responded yes to this question, but in order to obtain more accurate results, testing needs to be conducted on a larger number of people in future. Regarding the entire program in general, they mentioned that they liked the easy functionality and user-friendliness.

Chapter 8

Summary

8.1 Work Results

During this project, we analysed the book market and existing solutions, people's attitudes towards printed and electronic books, familiarised ourselves with Salesforce and discussed its capabilities. As a result, within this project's scope, we could implement our ideas and create an MVP of the online Library with all the necessary functionality.

Let's go through the problems that we aim to solve when the program will be finalised:

- **Availability and accessibility of books.** Finding a required print book is not always easy in our time, especially if they are rare. Now, a user can find the necessary book among other users who live nearby or in other cities and then arrange to have the book sent by mail. By increasing the number of book options in different cities, books will become more accessible, and there will be fewer problems with book shortages.

- **Increasing the selection of books that can be read.** Ordinary users often have books that are not available in libraries or online stores. In addition, the number of copies, if it does not contradict copyright, will increase multiple times, allowing a larger group of readers to borrow the book.

- **Reduction of expenses.** As mentioned in the Chapter 3, students often refrain from printing articles or buying new ones because of their cost. In the future, we hope that users will not have to worry about the price, as they will be able to borrow the necessary materials from senior students and use the program to find available materials. But the number of students is large, and the required materials can quickly run out. The reduction of expenses also applies to all other potential readers who choose electronic copies to save money.

- **Ecology.** In the same Chapter 3, the ecological problem was mentioned, and it was noted that constantly printing materials or buying new ones is not desirable. Borrowing books solves several ecological problems: there is no need to consume electricity when reading electronic books, there is no need to buy new copies or print electronic copies, and now there is no need to dispose of old materials (throw away or give away for recycling). Users can now borrow all the books that are gathering dust at home.

- **Simplicity and clarity (received from testing).** In their recent comments, users mentioned that they appreciate the simplicity of the app, its understandable functionality, and even the design. However, there were issues with tab names, which will need to be solved in the design phase.

- **Ideas for using this program during the war (received from testing).** One of the users, while sharing their experience with the app, mentioned the initiative of gifting books to soldiers on the front lines. This led to a realisation that many people have lost everything they had due to war, and buying books is not convenient for them due to the need for minimal possessions. Of course, they can borrow books from libraries, but borrowing books through the app would be much more convenient for them, and at least they can return the book by mail.

Whether the app will be useful needs to be tested in practice, but according to many studies, including one that we are analysing in the Chapter 3 "Printed and Electronic Study Materials. Students perspectives", and based on the results of our testing, we can say that there is a demand for this among people.

8.2 Future Plans

In the previous section, we mentioned our final aims. In this section, we will discuss the details of our future plans.

During the development of our program, we discussed various solutions and potential future improvements and implementations. Each time a new idea came up that was worth considering, we added it to the list of planned improvements. In the following list, we have grouped the main plans for the future collected throughout the program's creation.

UI/UX design. One of the first steps in further developing our program will be to improve the design. To achieve this, we will use various design development methodologies. For example, two types of tests to improve the user experience that we plan to use are A/B testing (testing two similar pages that differ in a few elements) and low-fidelity prototype testing (testing the convenience of the program as only basic information is preserved while any images or colours are removed).

Messenger functionality improvements. This includes using standard Salesforce tools for notifications and updating real-time messaging. Additionally, during development, it was noted that we can implement auto-deletion of frequently shared contacts in the messenger after a certain period. However, the need and method of implementation for this feature still need to be discussed.

Improving system functionality. For our MVP, we selected the essential functions that we needed. Our next steps consist of adding new features such as a different way of viewing books (visualising books not in a list but in a grid with several books in a row), marking each book from which date it is available, expanding filtering options by city, language, authors, etc., and implementing sorting of the list alphabetically or by book exchange dates.

Implementation of new functionality. In order to improve user experience, we plan to implement some new functionality.

- Two types of calendars: the first, where for each book, the busy days will be visualised, and the second, where all the planned book exchanges made by the user will be visualised.

- Pagination. For a better user experience, we plan to add book list pagination to each of the pages.

- (obtained from surveys) The ability for users to create their own book lists and manage them instead of the two default lists.

- User profile page, where comments about the user and their reading statistics will be displayed. We also plan to develop a user ranking system. This improvement requires further analysis and research on what to include in the ranking and what other information about the user is desirable to visualise and can be viewed by other users.

External database. To ensure the success and expansion of the project, we need to address resource usage issues. The first and primary step will be connecting an external database to store photos. This step will help us increase system resilience and reduce workload. Additionally, we will be able to decrease our expenses as the use of the Salesforce database becomes increasingly expensive as the amount of data grows, and using external databases will be a much cheaper solution.

Mobile App. One of our future goals is to create a mobile version of the application so that users can use it on the go.

The project team and legal aspect. To further develop the program, we plan to involve a team of developers and a legal specialist who will help us investigate the copyright situation in more detail and assist with any other legal issues.

Bibliography

[1] Internet Archive. *Digital Library of Free & Borrowable Books, Movies, Music & Wayback Machine*. URL: https://archive.org/.

[2] NCC Group Research Blog. *Salesforce Security with Remote Working*. Apr. 2023. URL: https://research.nccgroup.com/2020/10/02/salesforce-security-with-remote-working/.

[3] CapStorm. *Salesforce Encryption: Levels of Encryption & How They Work*. May 2023. URL: https://www.capstorm.com/salesforce-tips/salesforce-encryption/.

[4] Open Library. *About open library*. URL: https://openlibrary.org/help/faq/about.

[5] Open Library. *Welcome to open library: Open library*. URL: https://openlibrary.org/.

[6] Data Center Locations. *Salesforce data center locations*. Sept. 2020. URL: https://datacenterlocations.com/salesforce/.

[7] Metallic.io. *Salesforce Backup*. URL: https://metallic.io/knowledge-center/glossary/salesforce-backup.

[8] Diane Mizrachi. "Online or Print: Which Do Students Prefer?" In: *Information Literacy. Lifelong Learning and Digital Citizenship in the 21st Century*. Ed. by Serap Kurbanoğlu et al. Cham: Springer International Publishing, 2014, pp. 733–742. ISBN: 978-3-319-14136-7.

[9] Zoë Morris. *Why use Salesforce and AWS together?* Sept. 2021. URL: https://www.salesforceben.com/why-use-salesforce-and-aws-together/.

[10] OCLC. URL: https://www.oclc.org/en/home.html.

[11] OCLC. *WorldCat: World's most comprehensive database of library collections*. URL: https://www.oclc.org/en/worldcat.html.

[12] U.S. Copyright Office. *What is copyright?* URL: https://www.copyright.gov/what-is-copyright/.

[13] Ágústa Pálsdóttir. *Advantages and disadvantages of printed and electronic study material: Perspectives of University students*. May 2019. URL: https://informationr.net/ir/24-2/paper828.html.

[14] Josh Rank. *10 Salesforce Data Replication Best Practices*. Mar. 2023. URL: https://www.autorabit.com/blog/10-best-practices-for-salesforce-data-replication/.

[15] SimilarWeb. *Openlibrary.org Traffic Analytics & Market Share*. URL: https://www.similarweb.com/website/openlibrary.org/.

[16] SimilarWeb. *Worldcat.org Traffic Analytics & Market Share*. URL: https://www.similarweb.com/website/worldcat.org/.

[17] Salesforce Trailhead. *Apex enterprise patterns: Domain & Selector Layers | Learn Selector Layer Principles*. URL: https://trailhead.salesforce.com/content/learn/modules/apex_patterns_dsl/apex_patterns_dsl_learn_selector_l_principles.

[18] Salesforce Trailhead. *Apex enterprise patterns: Service layer | Learn Service Layer Principles*. URL: https://trailhead.salesforce.com/content/learn/modules/apex_patterns_sl/apex_patterns_sl_learn_sl_principles.

[19] Salesforce Trailhead. *Data modeling*. URL: https://trailhead.salesforce.com/content/learn/modules/data_modeling.

[20] Salesforce Trailhead. *Lightning app builder*. URL: https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder.

[21] Salesforce Trailhead. *Salesforce Platform Basics*. URL: https://trailhead.salesforce.com/content/learn/modules/starting_force_com.

[22] Salesforce Trust. *Salesforce Trust Site*. URL: https://trust.salesforce.com/.

[23] Wikipedia. *Open library*. Apr. 2023. URL: https://en.wikipedia.org/wiki/Open_Library.

[24] OCLC WorldCat.org. URL: https://www.worldcat.org/.

[25] OCLC WorldCat.org. *About worldcat.org*. URL: https://www.worldcat.org/about.

[26] Kinza Yasar, Karolina Kiwak, and Delaney Rebernik. *What is Salesforce? A beginner's guide*. Sept. 2022. URL: https://www.techtarget.com/searchcustomerexperience/definition/Salesforcecom.