UCN, University College of Northern Denmark
IT-Programme
AP Degree in Computer Science
dmai0919

Library management system - MiniProject 1

Martin Benda, Simeon Plamenov Kolev, Sebastián Labuda, Erik Petra
23-09-2019

Title page

UCN, University College of Northern Denmark

IT-programme

AP Degree in Computer Science

Class: dmai0919

Participants:

Martin Benda

Simeon Plamenov Kolev

Sebastián Labuda

Erik Petra

Supervisor: Mogens Holm Iversen

Abstract/Resume

This project is supposed to create a Library management system, which will help the librarians in the UCN library easily and effectively do their job. This report will explain the thought process behind the system design shown via various ways throughout this report.

Normal pages/characters: 10.1075 pages/24258 characters

Introduction

The name of the project - Library Management System, suggests a system with a complex back-end with many modules connected to each other and working together altogether with a beautiful, user-friendly and functional UI. The idea of the project is to further improve the UI and add new useful functionalities, so that employees can have an easier time doing their job.

Schedule

Time schedule:

Monday 23.09.

9:11 - 11:40: Figuring out interview questions, preparation, first draft of domain model/use cases and workflow

12:00 - 12:40: Further preparation

12:40: Interview

13:00 - 13:30: Summary of Interview, assignment of work to individual members

Tuesday 24.09.

8:30 - 13:00: Update of use cases, workflow and employee-task-goal table, creation of basic mock-ups and domain model

13:00 - 13:30: Think-aloud testing

13:30 - 14:40: Assignment of individual work

Wednesday 25.09.

8:30 - 13:30: Finishing mock-ups, working on report

Thursday 26.09.

8:30-14:00: Further working on report and finishing it

Speaking in general we met the schedule pretty well, but sometimes complications didn't allow it. Since we had a lot of time during our first day before even meeting the librarian and having a bit of understanding of their system and their desires, we could create some questions for the interview. We also created the first draft of workflow, use-cases and domain model based on what we thought that is needed. This worked very well and we were prepared for the interview. After the interview we realised that many of our thoughts were wrong, so we summarized the interview and assigned the individual work according our schedule. On Tuesday we updated together our use cases, workflow and employee-task-goal table, which we have done at home the day before. Since we had plenty of time before our think-aloud test with the librarians, we created then together the mock-ups and started creating the domain model. Before the testing we were done with workflow, mock-ups, workflow, use cases and even domain model, so we were even ahead of schedule. Even though the test went really good, we realised that a we understood some of the librarians desires a little bit differently as well as the system itself. That was the time when we were behind our schedule. We summed up what needs to be done and assigned individual work. During Wednesday we needed to change and add things in every thing we have created. But it didn't take a lot and thanks to our great group work we caught up to the schedule, managed to end every single thing needed for our report and we started writing the report. We decided to add some things during afternoon hours and assigned individual work. During our last day, we were working on our report according our schedule and finished it.

Preliminary study

Our first activity, following the schedule we made beforehand, was preparing questions for the interview with the librarian Maiken, which took place at Sofiendalsvej campus of UCN, which was scheduled for the first day of project work at 1 pm on 23 of September. Our questionnaire consisted of questions about how the current system works - every functionality it has, the steps needed to use them and every little detail about them. Next we asked about the problems of the current system, what needs to be changed, what we should improve or add and finally we asked some questions regarding some special cases. We came out with most of the questions during the interview, based on immediate answers of librarian. Since we had a lot of time before the interview, we summed up some ideas of how will the system look like, what do we think about the system. We also created our first visions of workflow, employee-task-goal table, use-cases and domain model, based on our prior knowledge and deduction.

Questions:

- How does the current system work?
    - Who uses the system?
    - Do you follow any steps?
    - How do you register a book?
    - How do you register a new customer?
    - What needs to be done to register a loan?
    - How does the reservation work?
    - Can you choose for how long will you borrow the book?
- Do you have any problems with the current librarian system?
- What do you need to change?
    - Why do you need to change it?
    - What should it do after the change?
- Do you use any pre-made documents?
- Do you have anything you have to use(documents/GDPR)?
- What happens if someone doesn't return the book?
- If you have 2 exactly same books- do you have a copy number?

After summarising all answers we have gained from the interview, we combined all our notes into one, well structured document. We listed all the problems with the current system, what we can improve, the attributes needed for the books and users, and some specific details regarding loan/reservation deadlines.

NOTES:

- **Better UX/UI**
    - Less scrolling and clicking as well as copy-paste
- Switching user status currently too bothersome
- **Search by:** title, author, publication record?, ISBN, subjects…
- CRUDs and listings of **Users, Books, Loans, Reservations** with filters
- Interface for manually registering reservation and loan.
- Same books copies have different barcodes.
- **User:** Name, Address, E-mail, Phone number, CPR, LN=ID, pin code(can be changed by user later), role
- **Book:** Title, Author, Publisher, Publi. Year, edition, ISBN (wanted - summary/content/subjects), aggregation with the bar code.
- User roles: Librarian, employee, student, external (same rights as student)
- **Reservation:**
    - Book reserved for 7 days until picked up
- **Loans:**
    - 30 days default / custom lease time
    - Option to prolong loan (max 10 times)
- **New Options:**
    - Option to lease an ebook from external DB
    - Better foreign student registration

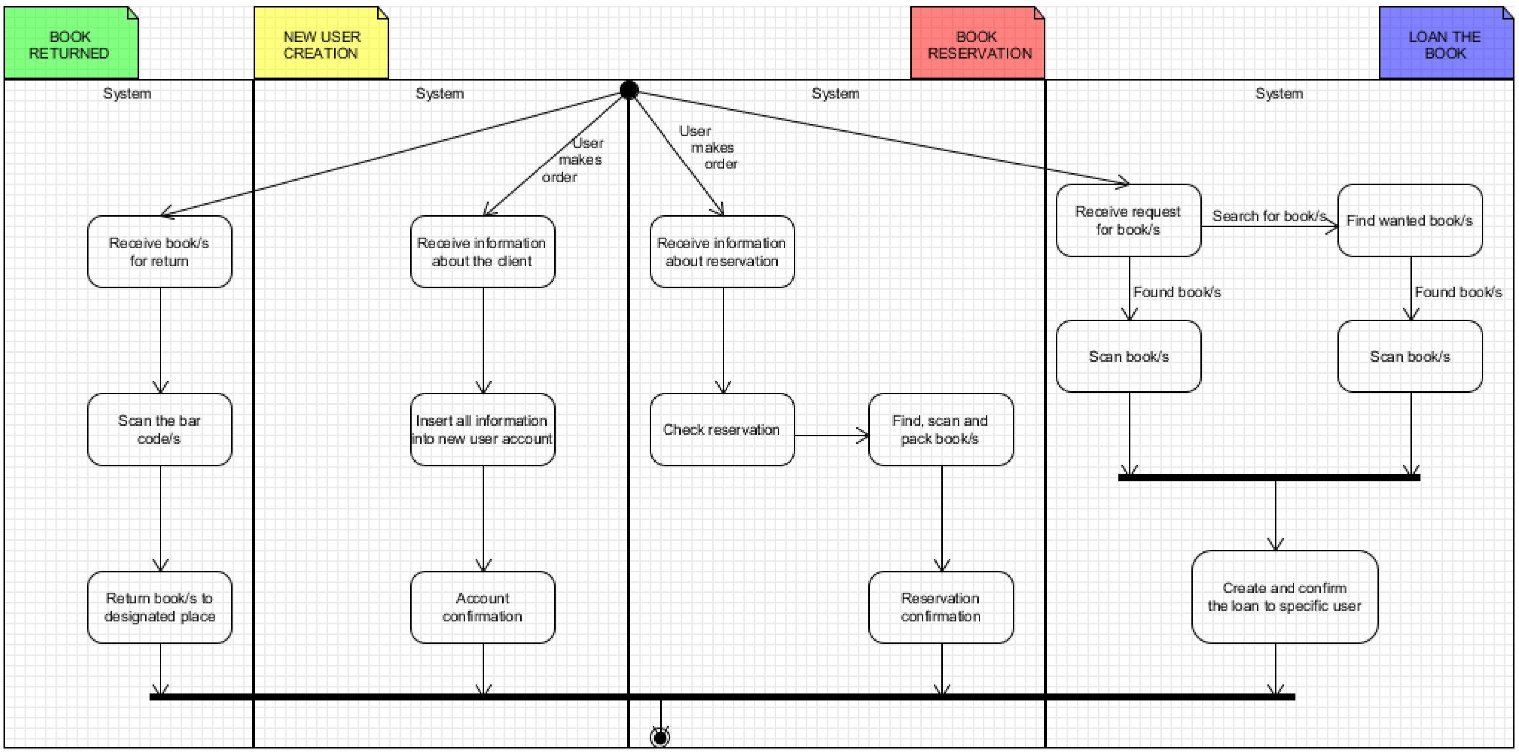
Employee-task-goal table

Then we decided to start with the Employee-task-goal table first, because it would serve us as a plan for making the Use Case Diagram and the Workflow Diagram. Employee-task-goal table is a simple sum of all people interacting with the system, their tasks in the system, things they have to achieve and how should it be done. The table contains the main actor, the librarian, the four tasks he does and the steps he takes to reach the goal. The librarian is the only actor, because the system is intended to be used by employees only and not customers.

Employee	Task	Goal	Steps in task
Librarian	Receive new reservation	Prepare the book for loan	Check the reservation Find the book Scan the book Reservation confirmation and prepare for taking
	Register new client	Allow new client to borrow books	Input new user info into the system
	Loan a book	Pass the book to client	Scan the book Confirm order
	Return a book	Change system status of the book	Receive the book Scan the book Return the book to the designated place

Workflow Diagram and Scenarios

Next on our TODO list was the Workflow Diagram, which is a more visual representation of the Employee-task-goal table, showing more about the connections between actor and tasks.



For some people, workflow can be easier to understand then Employee-task-goal table. However for many people workflow can be very confusing and complicated. For this reason, we're going to describe our workflow more into the depth and more precisely. Described will be all the scenarios as well, which is one particular story of using system, or one path through the [system][...]. [1]

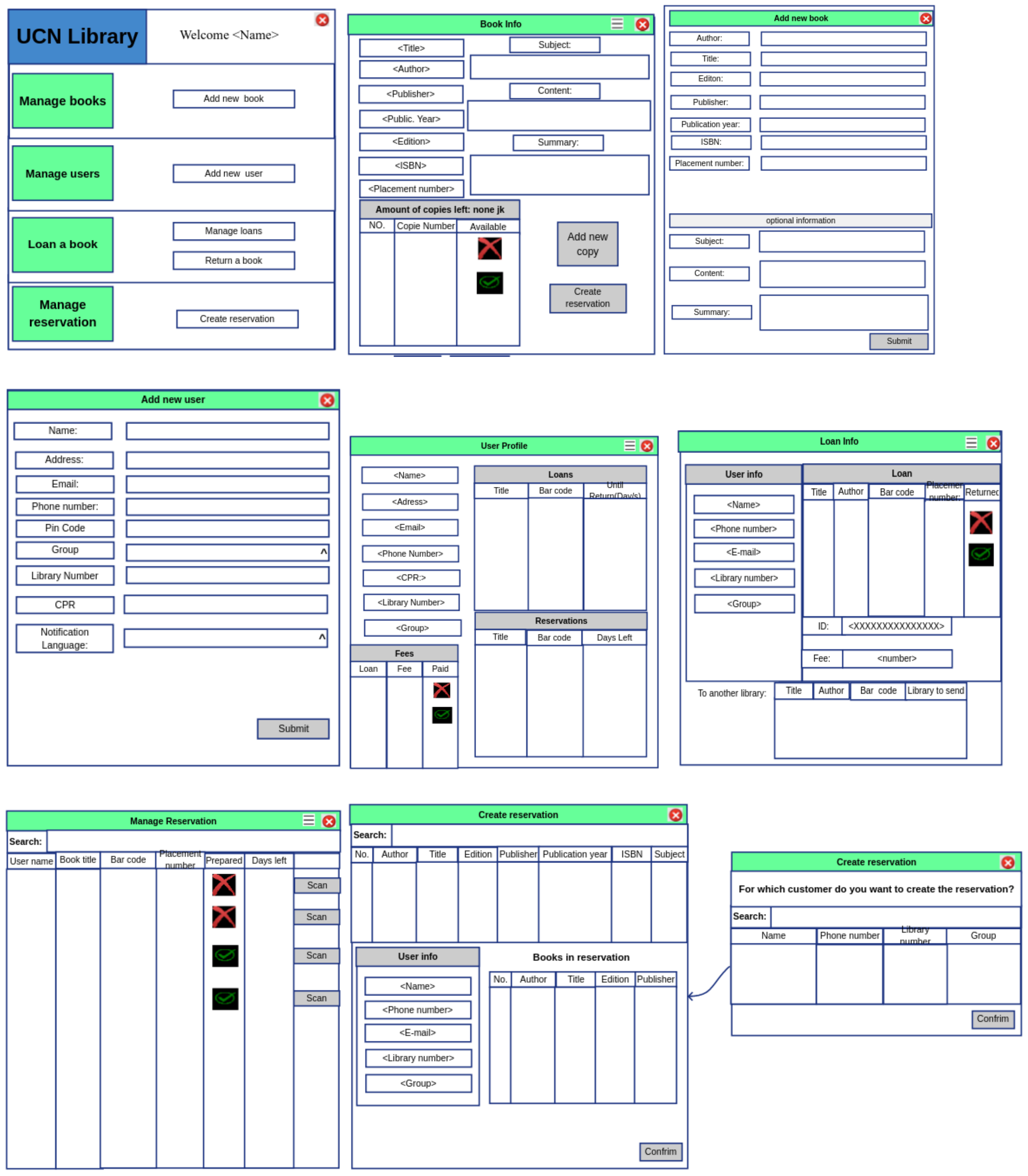
The workflow itself begins with big black dot as it represents the beginning of the whole workflow. Our workflow can be divided into four separate parts. To maintain order of whole workflow, we would like to start with describing the most right part, as it will be the most frequently used one. The right workflow "LOAN THE BOOK" obviously begins with task/activity "receive request for book/s. From this point, there are two different workflows. Customer may have already taken to book/s he wants, then the next task for librarian is to scan the book/s and then to create and confirm the loan to the specific users, just as it is described in the workflow. On the other hand, there is an option when the "customer" knows, which books he needs, but does not know where to find them. Responsibility of the librarian is to find the books thanks to the correctly described request from the customer. The course of actions is then same as in the first option. The workflow also includes long black line as it represents "the collection of flows". In other words, all of the tasks need to be finished before proceeding.

While first workflow describes the process of loaning the book, the second part will describe the process, when the customer wants to "book/reserve" the book. Under word "reserve" we understand the process, during which "customer" reserves book/s through the internet. As we can see in the workflow, tasks for the librarian are to find, scan and pack books. This can be done by checking the reservation, Through checking the reservation, librarian will be well aware about which books he/she needs to find and where to find them. This is the reason why task "check reservation" is before task find, scan and pack book/s. Of course we cannot forget about task "reservation confirmation" as it is very end of whole workflow.

The next workflow that will be described is the most left one, as it features the process of returning book/s. The workflow is very simple as it consists of tasks such as: receiving book/s, scanning bar code/s of book/s and put them into their desired place. Yet, there is the question in the air: "Why is there no option for paying taxes if you are returning book/s lately?". The answer is very basic. After the interview and think-aloud testing, we have learned that people prefer to pay taxes lately rather than during the day of returning of book/s.

The very last workflow is the less used, yet very important one and that is "NEW USER CREATION". Students and teachers are automatically registered into the system. Nevertheless, there are still other people who use or would like to use our system. Therefore this function is more than just necessary. As you can see in the workflow, the librarian first receives information of new user. He/she will have to manually insert all information about customer into newly created user account and confirm the creation of an account. The whole workflow as it is then ends with already used before "the collection of flows" as all activities should be finished.

Mock-ups



Mock-up is a prototype of the system. Although it will not do useful work beyond what the user sees, it will give an image of how the final software will look like and what can a user do with the software. Here you can see the most important mock-ups and the rest in the appendix 1. The first one shows librarians main window and access to all functions of the system. Clicking on buttons will pop-up a new screen, which can be always closed by pressing Esc or red button in the top right corner. In the section manage book, they can search through all the books and find the desired one according to many categories, as seen on the second mock-up. Double clicking on the book will show pop-up screen with the books information. In this page librarians can add new copy of the book, see if there is any copy available or create a reservation for the book. If the librarian wants to add a new book, which is not defined in the system, they simply go to "Add a book" section where they only provide needed information and confirm it. In the similar way you can add new user. There you can also choose the language, in which the user will receive notifications as well as his group, if he is a teacher, student, other library etc. If it is needed, librarians can also browse through all users and search for some according all the categories written in the mock-up. As with the books, double clicking on the user the librarian can see users information. There you can also see all the users loans and reservations, if the books are returned or when they need to be returned, if the user needs to pay any fees and how much does they need to pay. Double clicking on the loan librarians can see information about the loan. In the loan information there are some pieces of user information, as well as everything about the loan. If there are multiple books, you can see if there are any returned and their placement number, so it is easier for librarians to return the book on the shelf. Librarians also have there a fee info, if something needs to be paid. After clicking "Return a book", a pop-up screen will come in front of the loan information and asks for scanning the barcode. After scanning it, the same pop-up will come in case there are multiple books. After scanning all the books, the librarians cancels the pop-up screen and will be able to see, that the status of the scanned books in the loan information is now "returned" (there will be a check mark next to the books name). In case a book needs to be sent then to another library, the system will automatically add it to the section below the loan info, as well as it will add the name of the place where it should be sent. Librarians can see all the reservations in "Manage reservations" section. At the top they can see all unprepared reservations, so it is easy to see the work they need to do. after finding the books, librarians click the button "scan" and after scanning the barcode, the confirmation paper will be printed. If the librarian wants to create a reservation, they can either do it in the "book info" screen or do it from the home page. There they will be asked to search for user for which will be the reservation created. After that librarians will search in the system for books they want to add. Right clicking on the book they can either add it or delete it from the final list. Function "Loan a book" works the same as the "create reservation", but in the end it will show the whole loan and information about it. Most of the functions also have menu button in the top right corner, which can be seen in the last picture of appendix 1. There librarians can always undo the last action, if there was something done by mistake, edit information or delete the user/loan/reservation. In the "user info" they can also change pin code of the user and in the "loan info" they can prolong the loan, if the need arises.

Test Plan

The appointment with the librarian for the Think-aloud test was scheduled to be on 24 September 1 pm in the UCN library at Sofiendalsvej Campus. We decided that everyone together should present the mock-ups, help the librarian go through them (test them) and take notes of different problems that occurred during testing. The mock-ups we prepared beforehand served us as our test cases. We tried to add into the mock-up every possibly needed function according to our understanding of the desired system, so we could test every single detail. Everything went well regarding the time schedule.

Evaluation of the test

Since we prepared the mock-up depending on use-case diagram, as well as on domain model, we had to approach another step forward to creating a working and effective program, the think aloud testing. Test itself consisted of taking a closer look to our freshly developed mock-up. The test was meant to find different types of mistakes that could have been made during the preparation of mock up. What is more, the test itself showed new ideas that then could have been implemented into the system created by us.

Thanks to the result, we were capable to find solution to newly found mistakes, increase the potential of developing system and find new approaches how to solve various sorts of problems more effectively. During the test we also realised, that some functions in our system were different from what the librarian imagined. The Think-aloud test helped us to find those differences and agree with the librarian about the solutions, so we had a much better understanding of the system desired by librarians. We needed to change positions of some buttons or even delete them, exclude things in case of adding new user and book, we added the notification language section. We didn’t even know that the books could be transferred to other libraries right after confirming their return, so we needed to implement that to our mock-ups. We added fee section to every user as well as more sections, according to which you can look for a book and we changed the name “role” to “group” as desired, changed it in the domain model and added fee to the domain model.

System vision

The purpose of the project is to create better and more user-friendly system for the librarians, so they can work more effectively and more efficiently. We wanted to create a system, which will improve existing features and add new ones that will be useful or were requested by the librarians. After the systems implementations, the system should be easier to use, spare a lot of time and be richer in terms of functions. The scope of our project is essentially remastering the whole current Library System in UCN. The current system is very useful, but is lacking some key features and a lot of the work now is redundant and repetitive. A lot of the tasks that could be done in one step require two or even more, which takes a lot of librarians valuable time. We strive to correct that!

Stakeholders

Stakeholders in this project

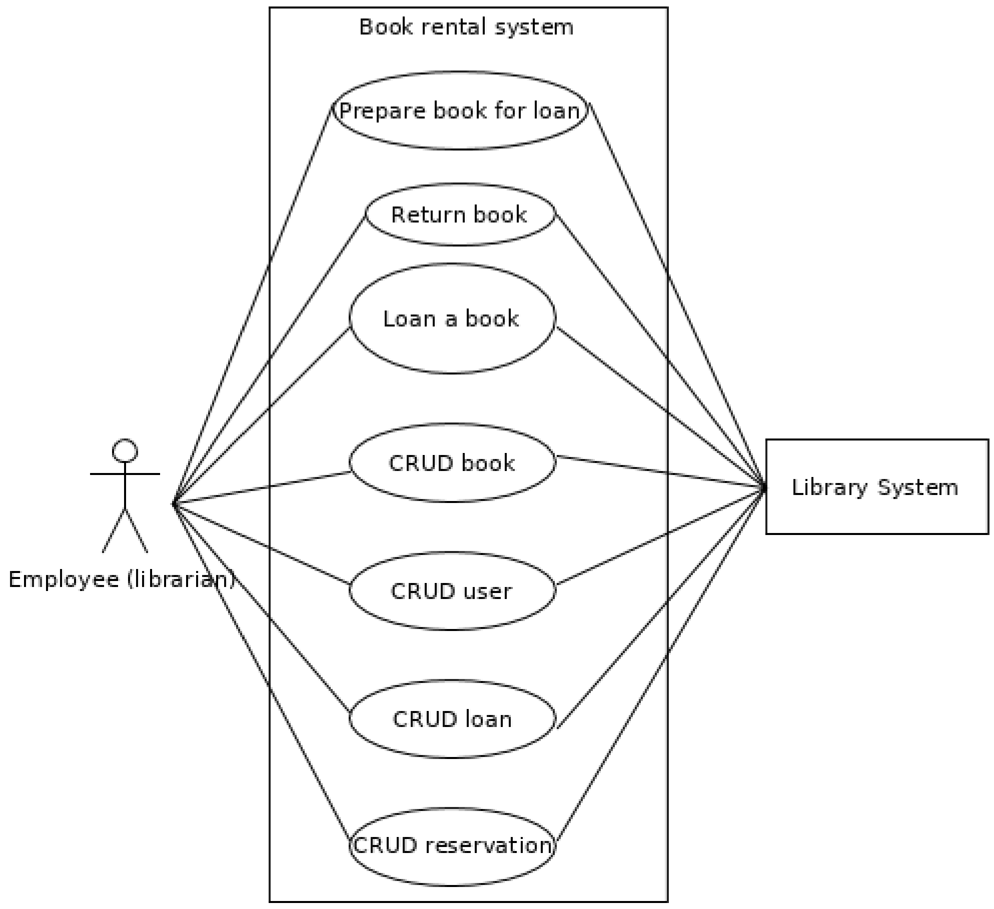
Suppliers	Clients
Project team: ● Martin Benda ● Simeon Plamenov Kolev ● Sebastián Labuda ● Erik Petra	UCN Organisation, UCN Library

Technology

Application will be written as Java Desktop application with Database located in cloud or school server (This may change). Application will also cooperate with school system for things like automatically registering new students.

Functional requirements

Use case diagram



Prioritization of use cases

Use case	Business value	Architectural importance	Resulting priority *
Prepare book for loan	10	4	40
Return book	4	1	4
Loan a book	9	3	27
CRUD book	2	9	18
CRUD user	4	7	28
CRUD loan	4	8	32
CRUD reservation	9	4	36

* (Resulting priority = Business value * Architectural importance)

A use case diagram is a graphic depiction of the interactions among the elements of a system. It can have many actors and unlimited number of verbs (use cases) as long as they interact with each other and not crash the system.

In our project - Library Management System, there is only one actor, the librarian, because the system is specifically made for the employees to work with. As for customers, they have to either ask the librarian when they have to borrow a book or use the Self-service machine to do it themselves. The latter option has only a limited amount of functions, like borrow a book and return it.

Every actor has a set of tasks to do. In our case the librarian is preparing books for loan, returning books, lending books, CRUD (Create, Read, Update, Delete) books, users, loans and reservations. While doing these tasks, the Library system is working “behind the scene” by storing the information inputted by the employee and displaying the result of the actions.

Every use case has a different priority. As shown on the table above, preparing the book for loan has the highest priority, because it’s the longest, most tedious task, which can’t be fully automated because it involves physical activity. That’s why we described it fully-dressed, explaining the main success scenario with all its steps needed to be done and the alternative scenarios, where there are problems with the system’s work process and displaying the proper error messages. We also described all other use cases in so called brief description, which means only the successful scenario is defined.

Books, users, loans and reservations are associated with CRUD (Create, Read, Update, Delete), because they involve all the basic operations. The librarian has to be able to do them freely, while on the other hand the customers asking for books or using the Self-service machine have no control over them.

Use case descriptions:

Prepare book for loan (brief):

Librarian checks the reservation made by a user in the Library System. The system then displays details about the book and the librarian finds it, scans it and confirms the reservation. After that they receive a paper with information about the reservation, which is put together with the book on the agreed place.

Return a book (brief):

A customer comes to the library to return the book. The librarian than scans the barcode of the book/s. The system displays the loan and shows the book as returned. After that, the librarian will put it on the designated place.

Loan a book (brief):

A customer comes to the library and asks for borrowing a book. If the customer has already found the book, the librarian scans it and adds the loan to the customers account. If they don't have it, the librarian helps them find it, scans it and add the loan to the customers account.

Create user (CRUD) (brief):

Librarian receives personal data from a customer and sends it to the Library System, which then creates customer's account. The librarian gives credentials to the customer.

Prepare book for loan (fully-dressed):

Actors: Librarian
Pre-conditions: Reservation exists in the system
Post-conditions: Reservation confirmed and customer has picked up the book
Frequency: 5 times a day
Main Success Scenario:

Actor(Action)	System(Response)
1.Librarian checks the reservation made by a user in the system.	2.System displays details about the reservation (name, book…).
3.Librarian finds the book.	
4.Librarian scans the book.	5.System receives information from the scan and sends notification to user.
6.Confirms reservation and prepares the book for the pick up.	

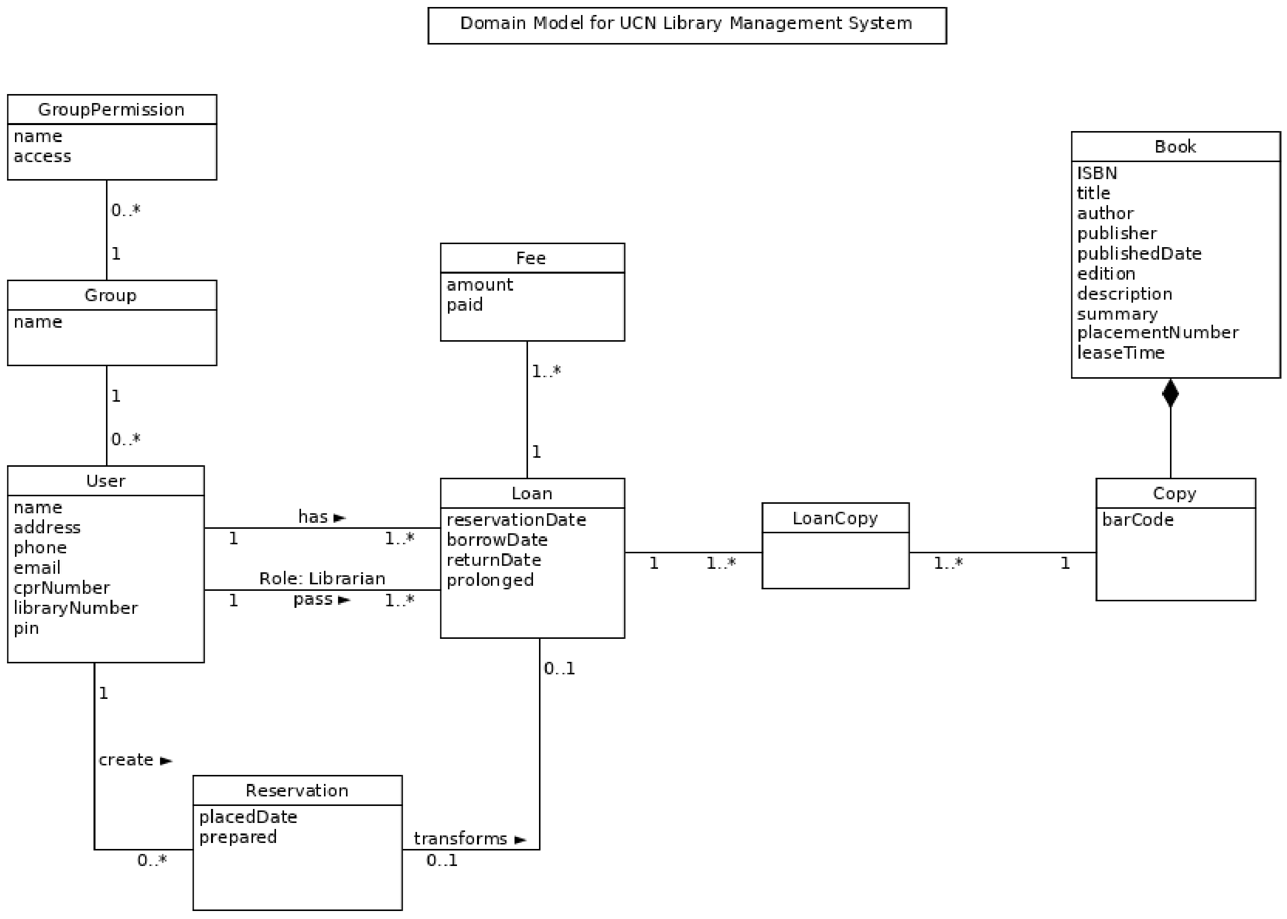
Information requirements

List of class candidates

Class	**Description**	**Use cases**
User	Represents every user that interacts or is registered in system	CRUD user
Group	Represents user group. Created to differentiate users.	
Group Permission	Represents permissions of certain group.	
Reservation	Act of reserving or keeping back specific book/s before their pick up	CRUD reservation, Prepare book for loan
Loan	Represents loan, contains all the dates needed.	CRUD loan, Loan a book, Loan a book

Fee	Represents fees bound to loan/user	
LoanCopy	Association table	
Copy	Aggregation of book, contains the bar code of each copy.	
Book	Information about a book.	CRUD book

Domain model



Domain model description

In library system user will be divided into groups (Class: Group) and each group will have certain rights/permissions (Class: GroupPermission) in the system. Relationship between User (Class: User) and Loan (Class: Loan) represents that Loan keeps track of User (Role other than librarian) who borrows books and User (Role of librarian) who passes books to the customer. Between User and Loan there is also Reservation (Class: Reservation). This class keeps track of whether the reservation for books has been made before borrowing. There is also the possibility that customer won't return the books on time, so we need to keep track of unpaid fees (Class: Fee). This class is bound to Loan, and thus to User. There is aggregation for books (Class: Book) called copy (Class: Copy). This is because each book title may have more copies, that are identified by unique bar code. Because the loan may contain more than one book, there is a need for association class (Class: LoanCopy).

Quality requirements

The Librarian pointed out one specific non-functional requirement. This requirement was about making user interface much clearer and easier to work with.

This requirement included:

- More relevant information shown in listings
- Connection of entities (Showing books in loan and clicking on book will cause book information to show), so that Librarian would have faster access to information.
- Faster UI (Smoother scrolling/less scrolling, faster and better filtering.

Conclusion

In summary we have created user-friendly functional system, which should help librarians work easier and reduce redundant work. We tried to include all the users requirements and wishes as well as added features which we considered useful and effective. The whole work went very smoothly without any delays, we easily agreed on most important parts and had no problem making compromise in smaller things.

Appendices

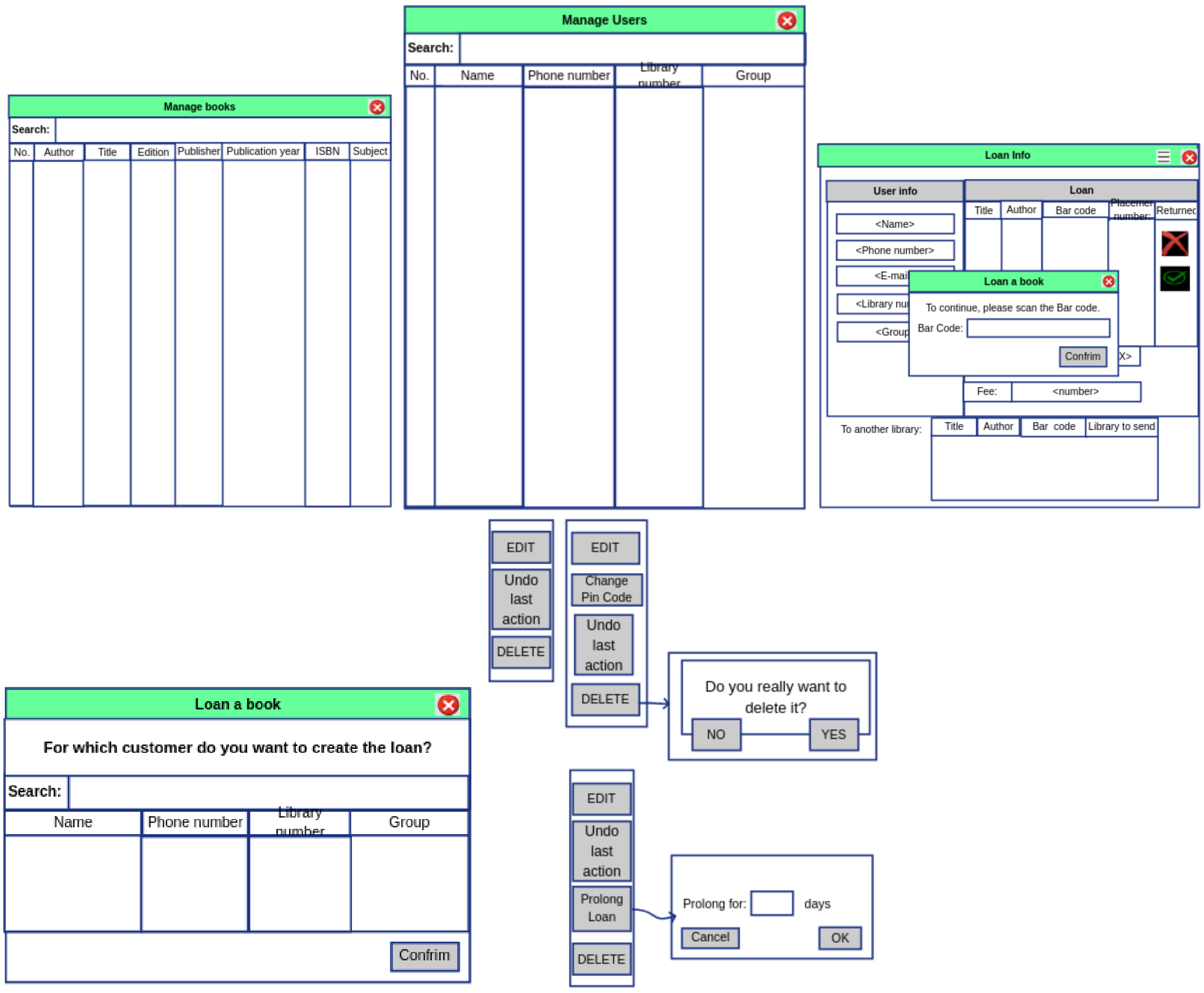
Appendix 1: Mock-ups

Literature

[1] C.Larman, “Applying UML and Patterns_ An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)“ p. 48, 2004.

Appendices

Appendix 1 (Mock-ups)


Manage Users
Search:
No.
Name
Phone number
Library number
Group
Manage books
Search:
No.
Author
Title
Edition
Publisher
Publication year
ISBN
Subject
Loan Info
User info
Loan
<Name>
<Phone number>
<E-mai
<Library nu
<Group
Title
Author
Bar code
Placement number
Returned
Loan a book
To continue, please scan the Bar code.
Bar Code:
Confrim
X>
Fee:
<number>
To another library:
Title
Author
Bar code
Library to send
EDIT
Undo last action
DELETE
EDIT
Change Pin Code
Undo last action
DELETE
Do you really want to delete it?
NO
YES
Loan a book
For which customer do you want to create the loan?
Search:
Name
Phone number
Library number
Group
Confrim
EDIT
Undo last action
Prolong Loan
DELETE
Prolong for:
days
Cancel
OK