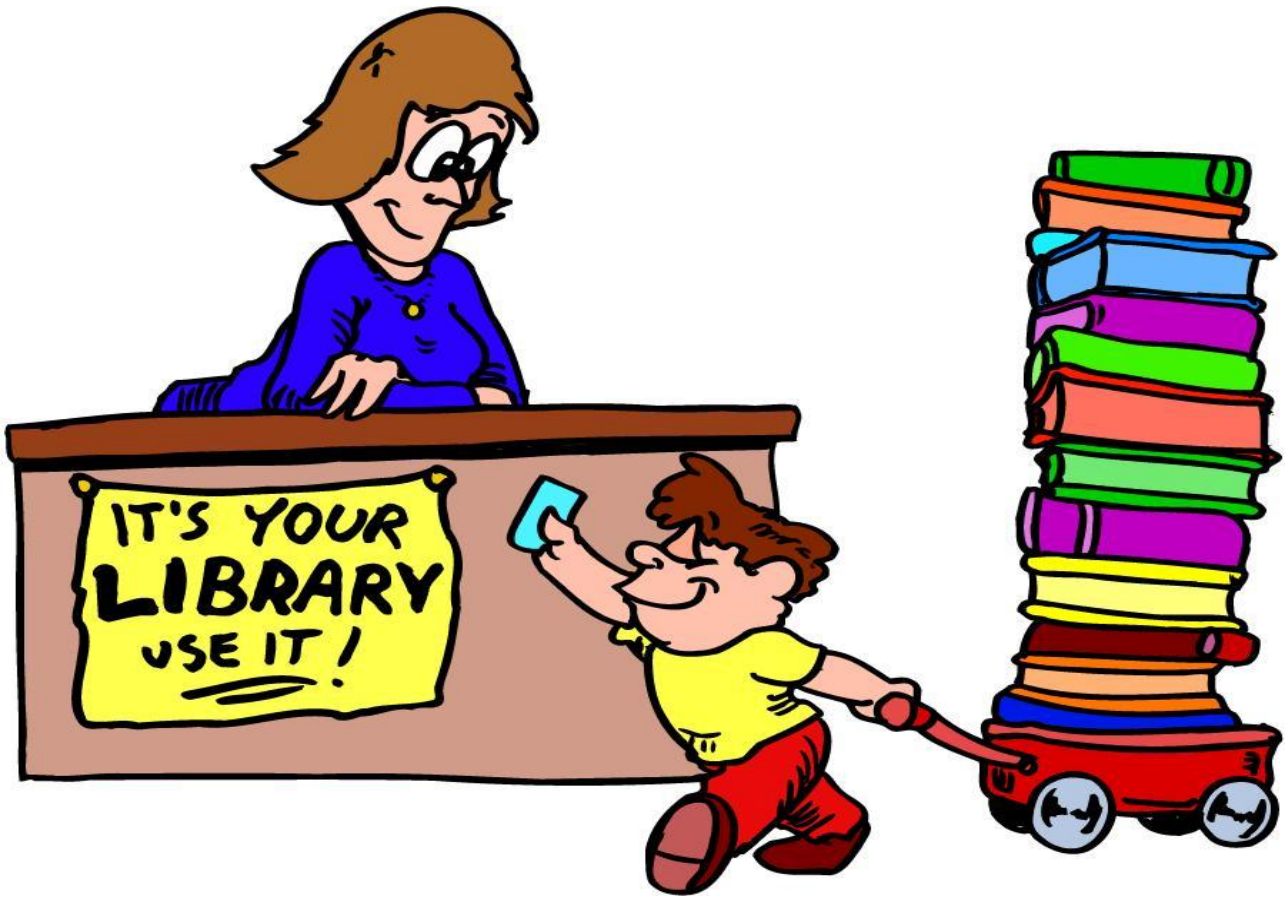
SPX5

FairytaleLib



GROUP MEMBERS:

JAVEED MAHMOUDI (139487)

CORTEL ALIN GEORGIE (164625)

RAFFAELE BRIVIO (185175)

Contents

1. Background description 4

1. Purpose 6

2. Problem formulation 6

4. Delimitation 7

5. Choice of method and model 7

6. Time schedule 10

7. References 11

1. Abstract 13

2. Introduction 14

3. Analysis 15

3.1. Requirements 15

3.2. Use case diagram 16

3.3. Activity diagram 18

4. Design 19

4.1. Graphical User Interface (GUI) 19

4.2. Model Class Diagram 23

4.3. Class Diagram 23

5. Implementation 24

6. Testing 33

6.1. Testing methods according to Use case 33

6.2. Test of GUI 35

7. Results 36

7.1. Known errors in the system 36

8. Conclusion 37

9. References 37

9.1. Books 37

9.2. Additional Resources 37

10. Appendix 38

10.1. Appendix 1 – Use Case Description 38

10.2. Appendix 2 – Activity Diagram 43

10.3. Appendix 3 – User Manual 47

10.4. Appendix 4 – Extended Class Diagram 54

11. Industries 55

12. Segmentation 57

13. Porter's 5 forces 58
14. Target group 63
15. 4 Ps 64
16. Project Planning 71
16.1. Purpose 71
16.2. Group Contract 72
17. Personal Considerations 75
17.1. Cortel Alin George 75
17.2. Javeed Mahmoudi 75
17.3. Raffaele Brivio 76

1. Background description

We represent a software development team, which was requested to assist in developing software that is capable of meeting the requirements established in the framework of a lending system in a library.

Fairytale Village University College has recently implemented study programs for students. These study programs require of the students and lecturers to research, and analyze information.

In order for the F.V.U.C. help their users, they have opened a FairytaleLib, a local library in Ping Pong Mesa. The current situation from an administrative point of view is that the newly opened library is managed by 2 librarians that keep track of items and logs of activities by using a spreadsheet, such as in excel for example.

The reason why they decided to hire us is because F.V.U.C is planning to expand the library, giving the possibility to the customers to have a wider range of items they can borrow. Due to expansion of range of items, information librarians need to register in the system is broader, therefore an excel sheet is not redundant for the future purposes.

Our purpose is to analyze the requirements of this project within the library and create a program that should be able to pin point all the functions such as stated within the interview that we had with the 2 librarians. After the framework of the software has been developed, our aim expands into looking in different industries, narrow them based on each market's requirements, select a specific market, narrow it down to one or more segments of the specific industry, and apply software framework with suitable adaptations in relation with the chosen segments of the market.

Current software framework is based on the interview between our development team and the librarians that require this program.

The items that librarians will work with are represented initially by books, articles and afterwards an expansion of CD's and DVD's. These items need to be able to be lent out, or reserved to either students or lecturers. After an item has been lent out or reserved, the system must have a search function that marks the item as lent out or reserved.

Once an item has been borrowed, information regarding the person who borrowed the item and the returning date should be associated with the item.

When taking in consideration an item that is being reserved, it should be associated with the information of one, or more persons that have planned a reservation for the item.

There is a difference in the logistics of keeping track of books, articles, CD's and DVD's.

One point is whether the person that is borrowing is a student or a lecturer. For books, the returning date for a lecturer is 6 months while for a student is 1 month. Articles require a 14 days of lent period for both lecturers and students.

Another point is taking in consideration the age of the material, if it is old (minimum one year old) or new.

New CD's and DVD's have a lent period of 14 days for both students and lecturers, while old CD's and DVD's have a lent period of 6 months for lecturers and 1 month for students.

Because students or lecturers can forget about returning date, the librarians thought that an email should be sent 4 days before due date of returning the item. If the item is not returned, an email is sent 2 days after due date, with a fine of 5 dollars.

Librarians specified they do not wish an automatic email system, but they prefer to receive a message regarding information of the item that has not been returned, the due date, and email address of the borrower.

The information related to items is different depending on the type of the items. Books should contain information related to their title, author, or ISBN (barcode).

Articles should include their title, author and magazine from which they were excerpt, while CD's and DVD's should include just their names for search purposes.

The software should also give the possibility to add new items, and the information should be stored in files, and not in a database system. The system developed by our team must be capable of being fully functional while being independent towards internet.

Besides the technical aspects of the framework, we must take in consideration that the market is full of software developers and with the evolution and advancement of IT sectors, they will steadily increase. For this reason, after an established target group, within the selected segment of the chosen market, our team needs to focus on the analysis of the players of the market and the market itself.

This system can be broaden up, and reach different segments of market, since is basically a lending, reserving system. With a little modification, this system can be applied to lending or reserving cars, bicycles or any other vehicle. A hotel might use this software to rent out rooms which act on the same principles as the items borrowed or reserved. The software's framework is the same, but for each different segment chosen, it is required a different market strategy, therefore different adaptations of the framework based on each segment's requirements.

Besides the task of creating the software that fulfills all the requirements stated above, it is needed to identify different segments of the market, in which the software can be applied and sold for a better efficiency in the segment where the customer is developing

One of the main analyses we need to focus on, for the reasons mentioned before, is regarding composition of the market within its players. For this purpose, Porter's five forces analysis is most suitable, allowing us to gather information regarding different threats about entrants, substitutes and competitors.

Regarding our product's potential, we must also take in consideration the place, price, and promotion, and after the thorough analysis, SWOT can be used as a sum up to reach a conclusion.

This system can be broaden up, and reach different segments of market, since is basically a lending, reserving system. With a little modification, this system can be applied to lending or reserving cars, bicycles or any other vehicle. A hotel might use this software to rent out rooms which act on the same principles as the items borrowed or reserved. The software's framework is the same, but for each different segment chosen, it is required a different market strategy, therefore different adaptations of the framework based on each segment's requirements.

Besides the task of creating the software that fulfills all the requirements stated above, it is needed to identify different segments of the market, in which the software can be applied and sold for a better efficiency in the segment where the customer is developing.

1. Purpose

The central idea of the current project is to create software that is capable of easing the work of the two librarians in Fairytale Village University College, within all the requirements stated in the interview with the librarians, in relation to items functionality.

Besides technical aspects of this task, we will analyze segments of markets in which the system developed for the Fairytale Library might be adapted for, and if any, choose the most appealing market implementing a strategy to enter into it.

2. Problem formulation

1. **How to create a system meant for the usage of librarians, that keeps the track of available items?**
   1.1. Which users are involved within the system?
   1.2. What items are meant to be kept a track on with the system?
   1.3. What functionalities can be available to the users of the system?
   1.4. In which way can the design of the system be projected using diagrams?
   1.5. In what language should the program be implemented?
   1.6. How to create a user friendly graphical interface for administrators?
   1.7. How a user manual of the system does looks like?
   1.8. How can we test the software?
   1.9. Who can we test the software on?
2. **How to enter the market?**
   2.1. Which industries might be suitable for our product?
   2.2. How to define and choose one industry to focus on?
   2.3. Which are the existing segments in the chosen industry?
   2.4. What is the target group in relation with the segments of the chosen industry?
   2.5. Which are the characteristics of other players of the market? (Porter's 5 forces)
   2.6. Is there a possibility for our product to be competitive on the market?

2.7. What would be an efficient action plan to enter the market? (4 P's) (Adaptation of the product, attractive price, method of distribution, attractive promotion)

4. Delimitation

- The system should not produce an automatic e-mail concerning returning dates.
- We are aiming at a B2C market.
- Our software lending system will expand to also lend out CD's and DVD's
- The items like books should contain information related to their title, author, or ISBN (barcode).
- Articles should include their title, author and magazine from which they were excerpt, while CD's and DVD's should include just their names for search purposes.
- The software system will be a intranet system (No online reserving/lending).
- The adaptation to the new target markets will be described but not implemented.
- More than a general analysis of costs we won't take into consideration the financial aspects of the project.

5. Choice of method and model

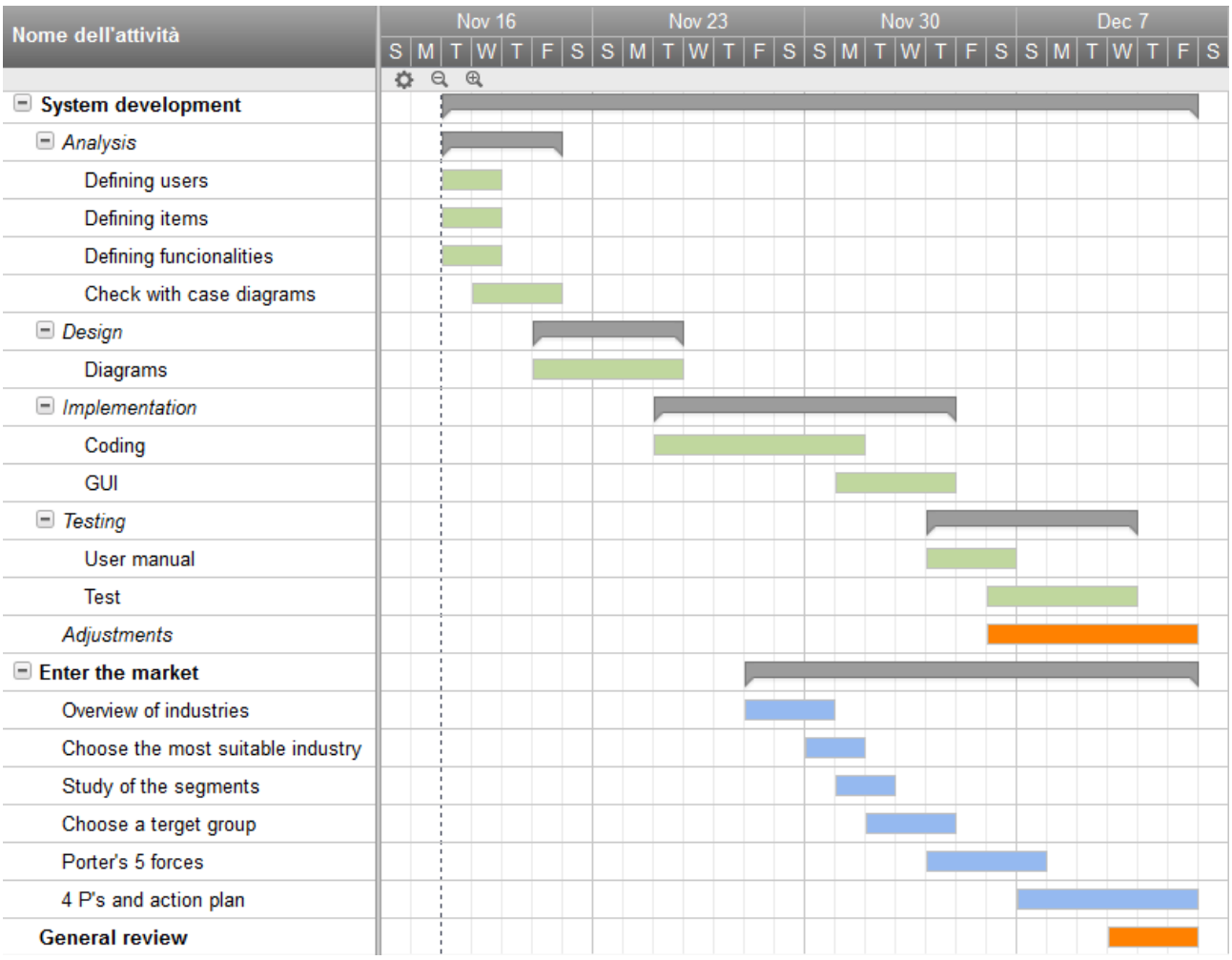
What is our problem?	**Why** do we wish to study it?	**How** will we solve it?(research, models ,theories)	**Which** depth should the problem have?	**Who** is responsible?	**When** is the Deadline?
What users are involved within the system?	To understand to whom we address the software	Interview and other resources	3	Javeed	19th November
What items are meant to be kept a track on with the system?	In order to understand what to implement in the software	interview	2	Cortel	19th November
What are the requirements of functionality available to the users of	In order to understand the difference between items and their usability	Interview And other sources	2	Cortel	19th November

the system?					
What are the non-functionalities of the system?	In order to get a more detailed vision of the software	Interview And other sources	3	Javeed	19th November
In what way can the functionalities of the system be projected?	In order to have in more detailed state the relation Actor-Functionality	Astah use case diagrams		Raffaele	19th November
In what way we can describe the steps involved within functionality?	In order to get a better grasp of a base and an exception sequence within functionality	Base and and exceptions analysis	1	Raffaele	19th November
How does each step interact within functionality?	To understand in a more detailed manor how the base and exception sequences work together	Astah activity diagrams	1		21st November
How will we implement the overall design of the system?	To get a better view of the design of the system	Astah class diagrams	1		25th November
How will we implement the activity within the whole system?	To understand how the classes between communicate	Astah sequence diagrams	1		25th November
How to create a user friendly GUI?	In order to make it more easier for the users to work with the software	Java programming	1		4th December
How to create a user manual of the system?	In order for the users to understand the functionalities within the software	Research and other resources	3		6th December

Who can we test the software on?	To understand if it is appealing to our target group	survey	3		10th December
How can we test the software?	To see if the software has any errors	Java Programming	2		10th December
Which industries might be suitable for our product?	in order to select the industry most appealing for our software's capabilities	Analysis Of appealing industries, brainstorming	2		30th November
How to define and choose one industry to focus on?	It is important because each industry requires different strategies, and adaptations	Decision based on analysis of industries	2		1st December
Which are the existing segments in the chosen industry?	Each different segment requires a different strategy, so we must know what is our focus	Segmentation	1		2nd December
What are the characteristics of other payers of the market?	This question is relevant for our understanding of our competition	Porter's 5 Forces	1		4th December
What is the target group in relation with the segments of the chosen industry?	This question is important because our strategy is directly related to our target group	Based on chosen segment, we compare which target group is most appealing	1		7th December
How to make our product competitive on the market?	This question is relevant to understand how efficient we can be on the market	SWOT	1		8th December
What would be an efficient	This question is related to	4P's and adaptation of	1		12th December

action plan to enter the market?	how can we increase our current level of efficiency on the market	the product			

6. Time schedule



7. References

Developing the software we need to meet the requirements of our customers. We are already in posses of the interview with the customers, stating their will; that is the most important document we have and we should base the structure of the software on that. Our task is to interpret the information we have in the best way possible.

Regarding the actual realization of the software we will need to look for information in academic books and online sources in case we don't find what we need in the academic material.

In order to find a suitable target group for our product and be able to enter the market we need to know the environment around us, collect data and be conscious of the area where we would like to work. To do this we have to extend our knowledge of the market and our purpose is to find all the information that we need through market research. In general all that we need, we can find in report already existing or academic books and also in some sources on internet.

Executive Summary

Fairytale Village University College has started developing new study programs for students. Besides creating this new environment for students, they had to take in consideration the fact that both lecturers and students require a place where they can research and analyze information. This university has implemented a FairytaleLib library system which should serve its purpose as an information center, a library, capable of lending items.

The problem that has arisen is related to how the information related to borrowed items and borrowers is stored in an efficient, safe and fast way.

Currently they are using an excel spredsheet to store their data, which is not as efficient as a system should be. Based on this case, the problem that arises is, how a more efficient system can be developed, meant to manage items, and include functionalities of which librarians wished for, stated in the interview.

Besides technical aspects, our team was needed to be able to pinpoint different industries in which our software framework can be implemented, select the most favorable ones, segment them and afterwards analyze segments in relation to the selected target group, in order to create an efficient marketing plan for the targeted group.

Our software developer's team's main purpose from a technical aspect is to focus on what are the requirements of librarians. Our main findings regarding the requirements are the possibility to add, lend, return, reserve, search books, articles, and afterwards an expansion of CD's to DVD's , in relation to the clientele of the library which consists of students and lecturers.
Since for each different item and for each different borrower the system should have an unique functionality, we had to create an overall user case diagram, and later on, create case descriptions, case diagrams and activity diagrams for lending, reserving, returning and searching.

After this has been implemented, the class diagram has been developed and the code was written for the whole class diagram, including java documentation. Since librarians need an easy way to harness the code, a Graphical User Interface also had to be implemented, and the functionalities of the code needed to be adapted to the Interface, with an extra File class, and 5 Adapter classes.

After finishing segmenting the chosen industries, the car rental and accommodation industry, we worked by analyzing Porter's 5 Forces, and developed more the problem within the 4 P's.

Our recommendation is to remain in Denmark Area, and to create a long term relation based on loyalty with our clients, to deliver direct proportional quality/price ratio software and to be active in the business to business network, with expectations of possible collaborations with different partners.

1. Abstract

FairytaleLib is a library recently open at Fairytale Village University College in Ping Pong Mesa. It is run by two librarians, Tasja and Uniqua, which are currently employed in the library. Their boss, Don Austin, required a rental software system for the library. The system, based on Java, must meet the requirements of the customer.

Program must be able to create reservation, lend already created reservations, list all the items (book and articles), list all available items, list specified books or articles, add to the system, list information about items and calculate the due returning date.

The software, which we developed specially for this customer, matches all the requirements and it can be easily adapted to different businesses, like car rental, accommodation rental or sport equipments rental. The program stores the data in files and gives possibility for future extensions.

2. Introduction

Lending books and articles seems to be a simple task both for customers and employee at a library. A library in a university lends items such as books and articles from magazines for different period of times (from 14 days for articles to a maximum of 6 months for books), to both students and lecturers; sometimes giving the possibility to borrow and reserve also online.

FairytaleLib is a library situated in Ping Pong Mesa, managed by Don Austin who wants us to create a rental software system for his library because the old method, using excel sheets is not efficient anymore.

FairytaleLib wants a single user system, to handle the operations for their customers. The librarians want to have an easy way to reserve items, see the list of all the books and articles present in the library, see which of these are available at a specific time and to store information like new items and information regarding customer data, all stored in files; furthermore, the librarians required for the system the possibility to warn them when the due returning date is approaching and when is past.

The manager does not want to have an online renting system.

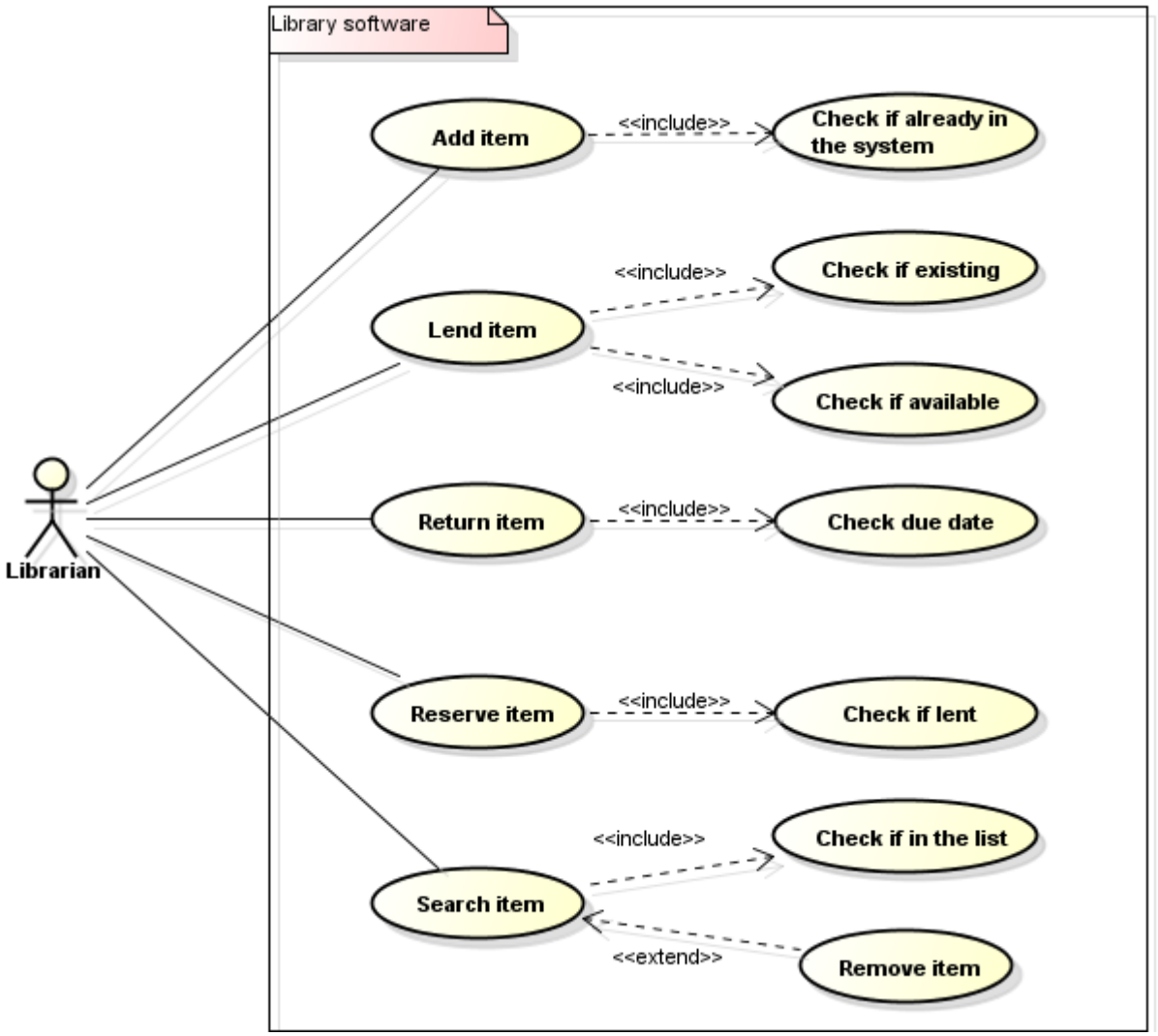
3. Analysis

3.1.Requirements

- User must be able to add books with Title, Author, ISBN and date of publishing
- User must be able to add articles with Title, Author and Magazine.
- Users must be able to lend books to lecturers and students.
- Users must be able to lend articles to lecturers and students.
- Users must be able to mark a book as lent writing Name, Surname, Birthday, Email of a borrower and the returning date.
- Users must be able to mark an article as lent writing Name, Surname, Birthday, Email of a borrower and the returning date.
- Books need to have a lent period of 6 months for lecturers.
- Books need to have a lent period of 1 month for students.
- Articles need to have a lent period of 14 days for students.
- Articles need to have a lent period of 14 days for lecturers.
- Users must be able to return lent books.
- Users must be able to return lent articles.
- Users must be able to reserve books for one or more students or lecturers if items are not available.
- Users must be able to reserve articles for one or more students or lecturers if items are not available.
- Information related to reserved items must consist of borrower's First Name, Last Name and email address.
- Books and articles become available for those who reserve, after due date of previous lender has come.
- The software must be able to notify with a message 4 days before the due date of items that must be returned, with information related to item, due date, and email address of borrower.
- The software must be able to notify with a message 2 days after the due date of items that must be returned, with information related to item borrowed, due date, email address of borrower, and 5$ fine.
- Users must be able to search books by Title.
- Users must be able to search articles by Title.
- Users must be able to store in files borrowed books and articles
- Users must be able to store in files books and articles
- Users must be able to store in files borrower's personal information.
- Users must be able to store in files information of students or lecturers who reserved any items.
- Users must be able to search books by Author.
- Users must be able to search articles by Author.
- Users must be able to search books by Barcode.
- Users must be able to search articles by Barcode.

- Users must be able to search books by Date in which it has been published.
- Users must be able to search articles by Magazine from which article has been excerpt.
- Users must be able to search articles by Date in which it has been published.

3.2.Use case diagram



Add item: The librarians should be able to add items from the system.

> **Check if already in the system:** The software check if an item already exists in the files, to avoid adding it double.

Lend an item: The librarians should be able to lend an item, storing the information of the borrower in the system.

> **Check if existing:** The software checks if the item exists in the system for lending purposes.

Check if available: The software checks if the item is available to be lent or if already in possess of another customer.

Return an item: The librarians should be able to mark an item previously lent, as returned.

Check due date: The software checks if the item is returned on time or late.

Reserve an item: The librarians should be able to reserve items for customers that require the reservation.

Check if lent: The software checks if the item is lent or available, in case of availability, it cannot be reserved.

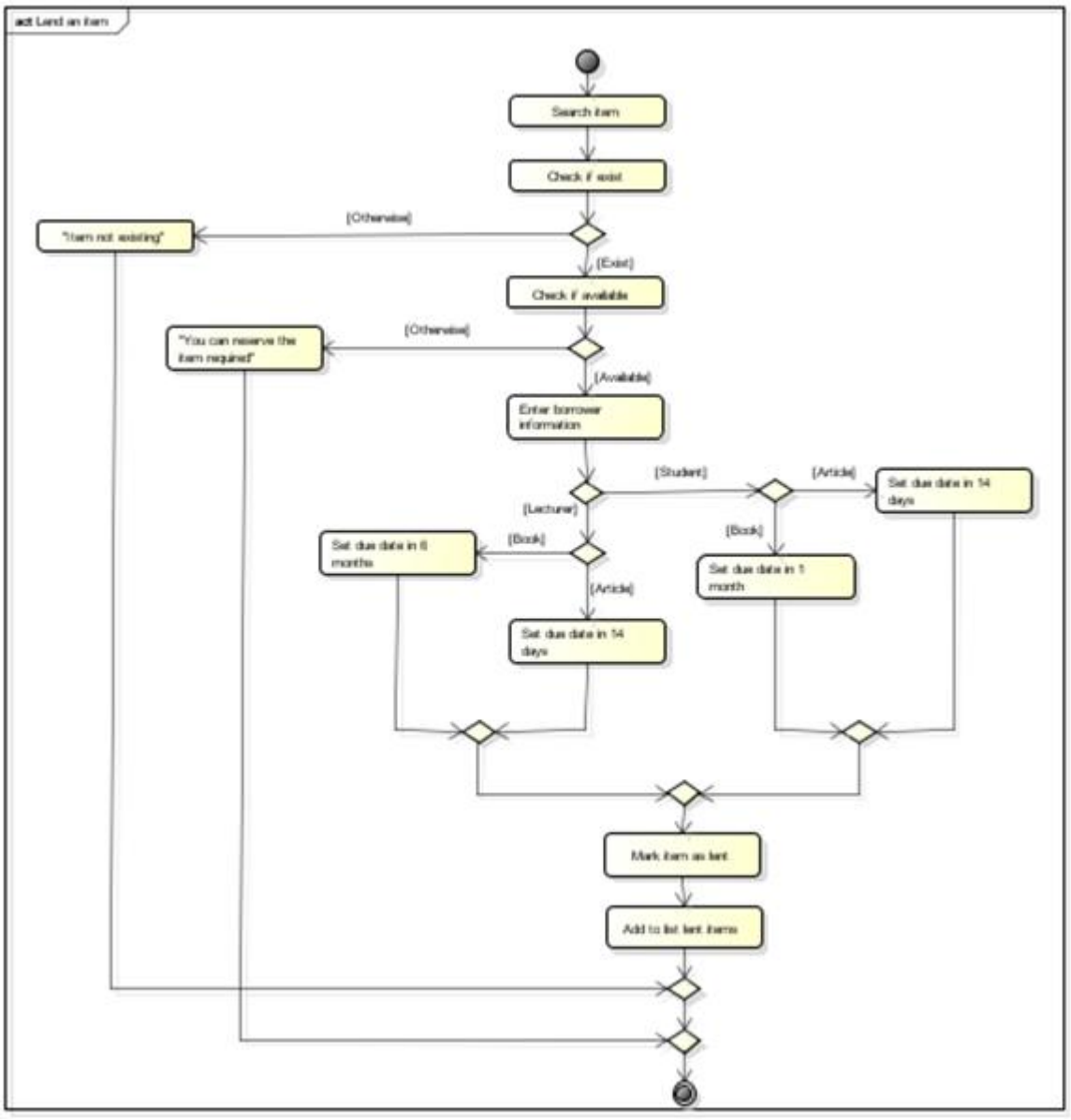
Search item: The librarian should be able to search in the software all the items stored in files, looking for title, author, magazine or ISBN.

Check if in the list: The software display a warning message in case there are not items in the list or in case there are none meeting the research criteria.

Remove item: The librarian once having the result of the research, should be able to delete items from the files in case they need to.

Appendix 1 – Use Case Description.

3.3. Activity diagram



The function of lending is the most essential features of the system because the main purpose of a library and accordingly, of the software is to allow people to borrow that the library offers, in this case with students and lecturers that can borrow books and articles. The lending activity diagram displays step-by-step the process of lending an item, distinguishing carefully basing on who is borrowing and what.

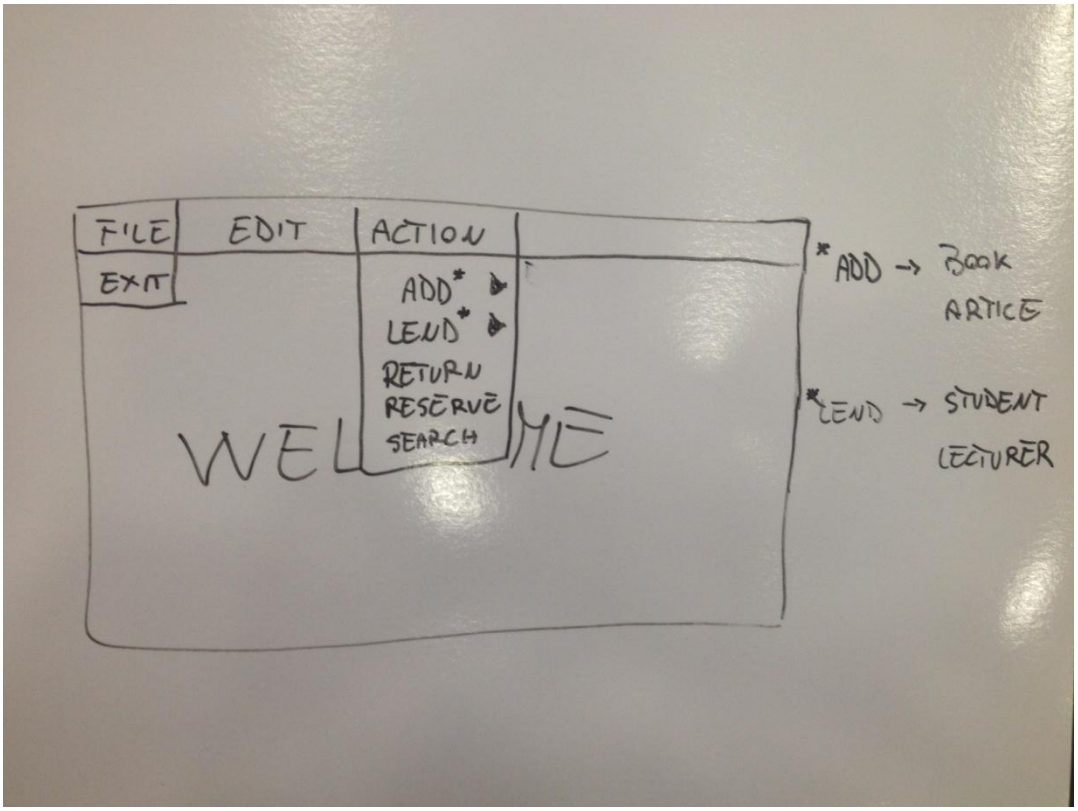
Appendix 2 – Activity Diagrams.

4. Design

4.1.Graphical User Interface (GUI)

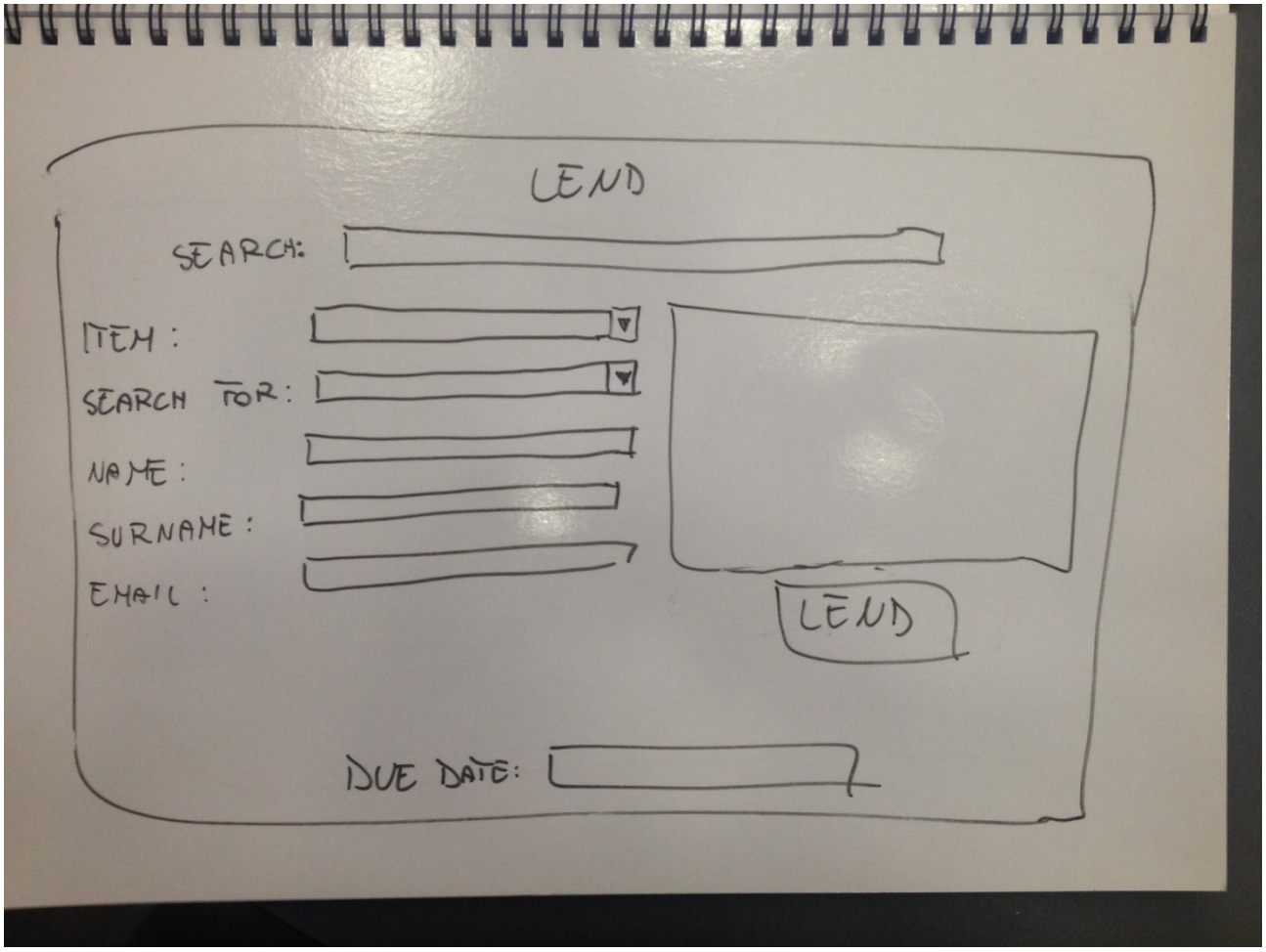
Main Menu: Before implementing the GUI, sketches of those were drawn; the main GUI, the one where is possible to reach and open all the others, consist in a main menu with items of the menu representing the main features of the system.

A simple interface was chosen for the main menu, with a menu bar and its menus and items. Each item has a field which displays his name. In this way the user can interact with the system very easily.

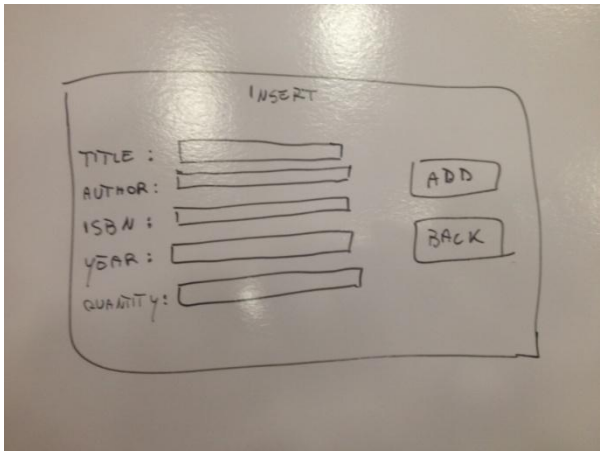


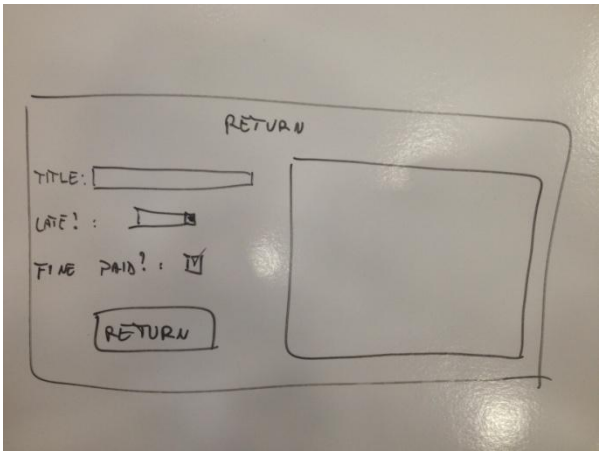
The most important function of our software is the one that allow the librarians to store information regarding the lending of the items offered by the library.

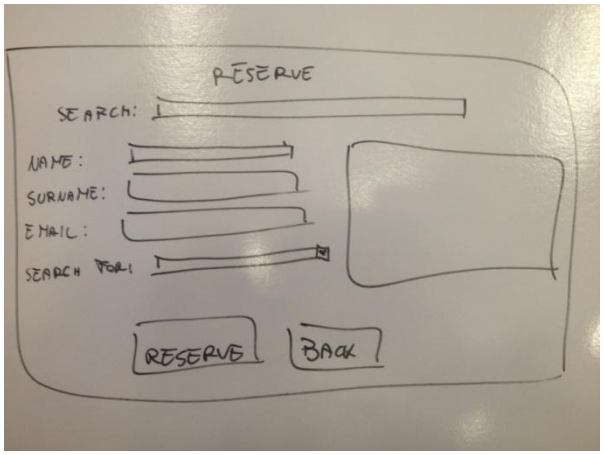
This method is designed with a search bar where is possible to look for the items stored in the system, choosing the criteria of the research; the user lend an item filling the information of the borrower and selecting the item chosen.

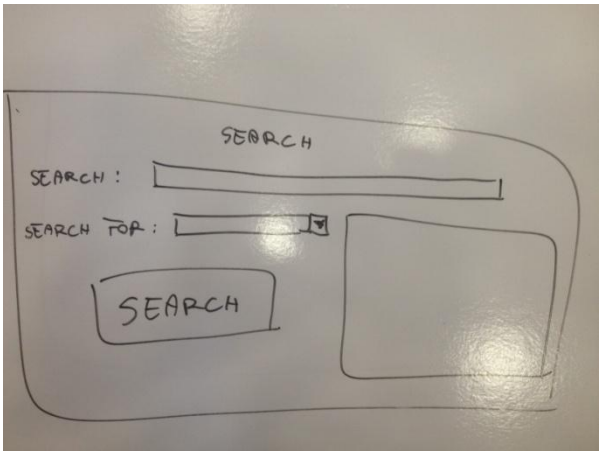


The system allows the customers to reserve an item in case in the moment that is required, if there is no availability in the library, storing the information in a list; there is also the possibility to add new items and to search among those already in the system. Furthermore it is possible to register that an item it has been returned after the allowed period of time, taking into consideration if it has been returned on time or late.









The result it's similar to the sketch because we tried to recreate that as much similar as we could; the reason is that it's simply to understand and use.

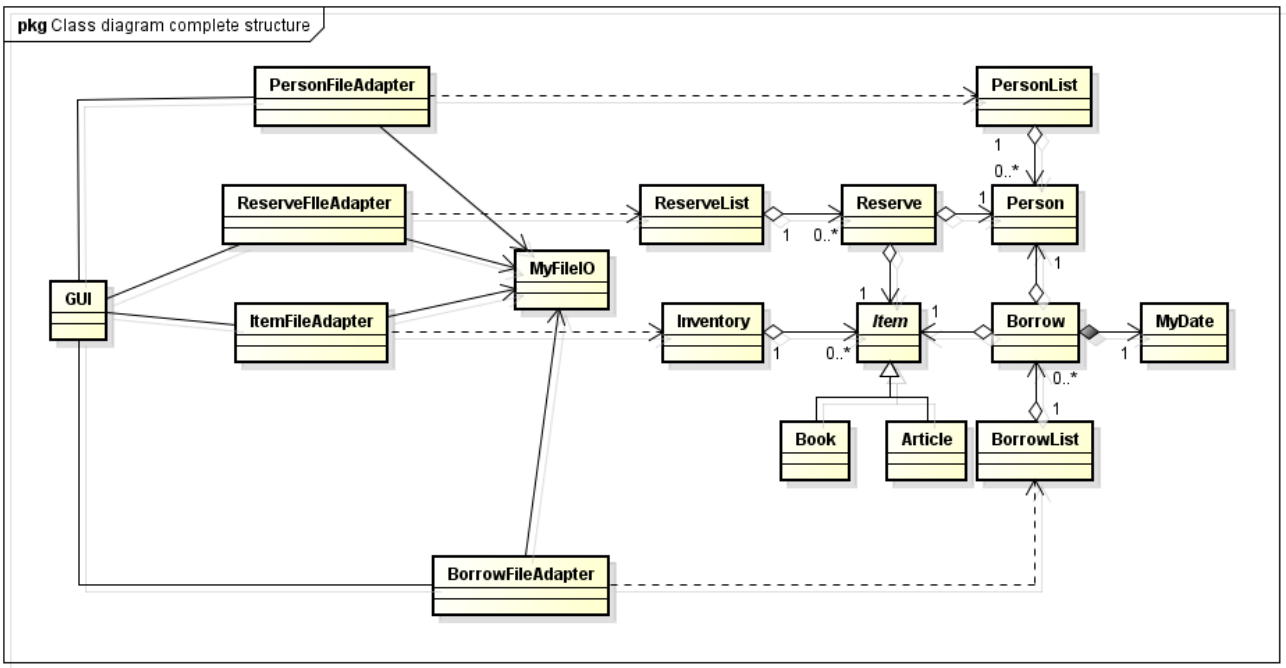
Appendix 3 – User Manual

4.2.Model Class Diagram

The basic model shown below illustrates the basic structure of the system. Hereafter we are going to briefly explain the classes present in our system. Book and Article classes are subclasses of the super class Item; they are used to create new items, respectively books and articles in the system, storing information about those. Inventory class stores all the items in the system and the relative information. The Person class "creates" new persons, customers of the library and PersonList class stores all the persons with the relative information. Person and Item classes are used to add objects of those two to the class Reserve; ReserveList stores all the Reserve objects with different information and methods to get the reservations. MyDate class creates new dates with different methods for the calculation to compare different dates. Person, Item and MyDate classes are used to add objects of those to the class Borrow; BorrowList stores all the Borrow objects with methods to store and return all of them, differentiating between books and articles.

In the system we have four adapter classes: ItemFileAdapter, PersonFileAdapter, BorrowFileAdapter and ReserveFileAdapter that are used to connect the classes previously mentioned, the lists in particular, to the Graphical User Interface: The adapter classes are also linked to MyFileIO, class that allow to write in binary code all the information we have in the classes, storing them in files .bin.

4.3.Class Diagram



Appendix 4 – Extended Class Diagram

5. Implementation

One of the most important methods of our program is to lend items to customers that require that.

In order to lend an item, using the GUI designed, the user has to search the item in the system that is looking for, selecting the criteria of the research; pressing the button “Search” all the item that satisfy the criteria will appear in a list. The user will then write the information of the borrower in the right text fields together with the date of the lending and select the item desired. Pressing “Lend” the system will store the information of the person in a list, in case he’s not already there and both info regarding customer and item will be written in a list with all the lendings.

```
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;

import javax.swing.*;

public class LendStudent extends JFrame
{
   private ItemFileAdapter adapter;
   private PersonFileAdapter adapter2;
   private BorrowFileAdapter adapter3;

   private static final long serialVersionUID = 1L;
   private final int WINDOW_WIDTH = 600;
   private final int WINDOW_HEIGHT = 330;

   private JLabel labelHead;
   private JLabel labelSearch;
   private JLabel labelName;
   private JLabel labelSurname;
   private JLabel labelEmail;
   private JLabel labelDueDate;
   private JLabel labelBox1;
   private JLabel labelBox2;
   private JLabel labelAvailable;
   private JLabel labelDate;
   private JTextField textSearch;
   private JTextField textName;
   private JTextField textSurname;
   private JTextField textEmail;
   private JTextField textDueDate;
```

```
private JTextField textAvailable;
private JTextField textDay;
private JTextField textMonth;
private JTextField textYear;
private JButton lendButton;
private JButton backButton;
private JButton searchButton;
private JList<Item> studentList;
private JComboBox itemBox;
private JComboBox searchBox;
private JPanel panel;
private JPanel panel2;
private JPanel panel3;
private JPanel panel4;
private JPanel panel5;
private JPanel panel6;
private JPanel panel7;
private JPanel panel8;
private JPanel panel9;

public LendStudent()
{
  super("Library software");

  adapter = new ItemFileAdapter("items.bin");
  adapter2 = new PersonFileAdapter("customers.bin");
  adapter3 = new BorrowFileAdapter("borrowed items list.bin");

  setSize(WINDOW_WIDTH, WINDOW_HEIGHT);
  setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
  setLayout(new BorderLayout());
  setResizable(false);

  buildNorthPanel();
  add(panel, BorderLayout.NORTH);
  buildWestPanel();
  add(panel4, BorderLayout.WEST);
  buildCenterPanel();
  add(panel5, BorderLayout.CENTER);
  buildEastPanel();
  add(panel6, BorderLayout.EAST);
  buildSouthPanel();
  add(panel9, BorderLayout.SOUTH);
}
```

```
private void buildNorthPanel()
{
  labelHead = new JLabel("Lend to a student", SwingConstants.CENTER);
  labelSearch = new JLabel("Search: ");
  textSearch = new JTextField(40);
  searchButton = new JButton("Search");
  searchButton.addActionListener(new MyButtonListener());

  panel = new JPanel(new BorderLayout());
  panel.setPreferredSize(new Dimension(400, 70));
  panel.setBorder(BorderFactory.createEmptyBorder(0, 0, 5, 0));
  panel2 = new JPanel();
  panel3 = new JPanel(new FlowLayout(FlowLayout.CENTER));
  panel3.setPreferredSize(new Dimension(100, 35));
  panel2.add(labelHead);
  panel3.add(labelSearch);
  panel3.add(textSearch);
  panel3.add(searchButton);
  panel.add(panel2, BorderLayout.NORTH);
  panel.add(panel3, BorderLayout.SOUTH);
  panel.add(panel3, BorderLayout.SOUTH);
}

private void buildWestPanel()
{
  labelName = new JLabel("Name: ");
  labelName.setBorder(BorderFactory.createEmptyBorder(0, 0, 5, 0));
  labelSurname = new JLabel("Surname: ");
  labelSurname.setBorder(BorderFactory.createEmptyBorder(0, 0, 5, 0));
  labelEmail = new JLabel("Email: ");
  labelEmail.setBorder(BorderFactory.createEmptyBorder(0, 0, 5, 0));
  labelBox1 = new JLabel("Item: ");
  labelBox1.setBorder(BorderFactory.createEmptyBorder(0, 0, 5, 0));
  labelBox2 = new JLabel("Search for: ");
  labelBox2.setBorder(BorderFactory.createEmptyBorder(0, 0, 5, 0));
  labelDate = new JLabel("Date: ");
  labelDate.setBorder(BorderFactory.createEmptyBorder(0, 0, 5, 0));
  labelAvailable = new JLabel("Available: ");
  labelAvailable.setBorder(BorderFactory.createEmptyBorder(0, 0, 5, 0));

  panel4 = new JPanel(new FlowLayout(FlowLayout.LEFT));
  panel4.setPreferredSize(new Dimension(75, 300));
  panel4.add(labelBox1);
```

```
    panel4.add(labelBox2);
    panel4.add(labelName);
    panel4.add(labelSurname);
    panel4.add(labelEmail);
    panel4.add(labelDate);
    panel4.add(labelAvailable);
}

private void buildCenterPanel()
{
    textName = new JTextField(20);
    textSurname = new JTextField(20);
    textEmail = new JTextField(20);
    String[] items = {"Book", "Article"};
    String[] search = {"Title", "Author", "Magazine", "ISBN"};
    itemBox = new JComboBox(items);
    itemBox.setPreferredSize(new Dimension(225, 21));
    searchBox = new JComboBox(search);
    searchBox.setPreferredSize(new Dimension(225, 21));
    textDay = new JTextField(2);
    textMonth = new JTextField(2);
    textYear = new JTextField(4);
    textAvailable = new JTextField(12);
    textAvailable.setEditable(false);

    panel5 = new JPanel(new FlowLayout(FlowLayout.LEFT));
    panel5.add(itemBox);
    panel5.add(searchBox);
    panel5.add(textName);
    panel5.add(textSurname);
    panel5.add(textEmail);
    panel5.add(textDay);
    panel5.add(textMonth);
    panel5.add(textYear);
    panel5.add(textAvailable);
}

private void buildEastPanel()
{
    panel6 = new JPanel(new BorderLayout());
    panel7 = new JPanel();
    panel8 = new JPanel();
    panel8.setBorder(BorderFactory.createEmptyBorder(0, 0, 15, 0));
    studentList = new JList<Item>();
```

```
    studentList.setVisibleRowCount(6);
    lendButton = new JButton("Lend");
    lendButton.addActionListener(new MyButtonListener());
    backButton = new JButton("Back");
    backButton.addActionListener(new MyButtonListener());

    DefaultListModel<Item> listModel = new DefaultListModel<Item>();

    studentList = new JList<Item>(listModel);

    panel7.add(studentList);
    panel8.add(lendButton);
    panel8.add(backButton);

    JScrollPane studentListScrollPane = new JScrollPane(studentList);
    panel7.add(studentListScrollPane);
    panel6.add(panel7, BorderLayout.CENTER);
    panel6.add(panel8, BorderLayout.SOUTH);
  }

  private void buildSouthPanel()
  {
    labelDueDate = new JLabel("Due date: ");
    textDueDate = new JTextField(20);
    textDueDate.setEditable(false);

    panel9 = new JPanel();
    panel9.add(labelDueDate);
    panel9.add(textDueDate);
  }

  private class MyButtonListener implements ActionListener
  {
    public void actionPerformed(ActionEvent e)
    {
      // Search button to look for an item with possibility
      // to search for a book or an article and to search
      // by title, author, magazine or isbn
      // returning the string with the info of the item found

      if (e.getSource() == searchButton)
      {
        if(itemBox.getSelectedItem().equals("Book"))
        {
```

```
if(searchBox.getSelectedItem().equals("Title"))
{
  String str = textSearch.getText();

  Inventory inv = adapter.getAllItems();
  ArrayList<Book> books = inv.getBooks();

  DefaultListModel<Item> listModel = new DefaultListModel<Item>();

  for(int i = 0; i < books.size(); i++)
  {
    if(books.get(i).getTitle().equals(str))
    listModel.addElement(books.get(i));
  }
  studentList.setModel(listModel);
}
else if(searchBox.getSelectedItem().equals("Author"))
{
  String str = textSearch.getText();

  Inventory inv = adapter.getAllItems();
  ArrayList<Book> books = inv.getBooks();

  DefaultListModel<Item> listModel = new DefaultListModel<Item>();

  for(int i = 0; i < books.size(); i++)
  {
    if(books.get(i).getAuthor().equals(str))
    listModel.addElement(books.get(i));
  }
  studentList.setModel(listModel);
}
else if(searchBox.getSelectedItem().equals("ISBN"))
{
  String str = textSearch.getText();

  Inventory inv = adapter.getAllItems();
  ArrayList<Book> books = inv.getBooks();

  DefaultListModel<Item> listModel = new DefaultListModel<Item>();

  for(int i = 0; i < books.size(); i++)
  {
    if(books.get(i).getIsbn().equals(str))
```

```
            listModel.addElement(books.get(i));
        }
        studentList.setModel(listModel);
    }
}
else if(itemBox.getSelectedItem().equals("Article"))
{
    if(searchBox.getSelectedItem().equals("Title"))
    {
        String str = textSearch.getText();

        Inventory inv2 = adapter.getAllItems();
        ArrayList<Article> articles = inv2.getArticles();

        DefaultListModel<Item> listModel = new DefaultListModel<Item>();

        for(int i = 0; i < articles.size(); i++)
        {
            if(articles.get(i).getTitle().equals(str))
            listModel.addElement(articles.get(i));
        }
        studentList.setModel(listModel);
    }
    else if(searchBox.getSelectedItem().equals("Magazine"))
    {
        String str = textSearch.getText();

        Inventory inv2 = adapter.getAllItems();
        ArrayList<Article> articles = inv2.getArticles();

        DefaultListModel<Item> listModel = new DefaultListModel<Item>();

        for(int i = 0; i < articles.size(); i++)
        {
            if(articles.get(i).getMagazine().equals(str))
            listModel.addElement(articles.get(i));
        }
        studentList.setModel(listModel);
    }
    else if(searchBox.getSelectedItem().equals("ISBN"))
    {
        String str = textSearch.getText();

        Inventory inv2 = adapter.getAllItems();
```

```
        ArrayList<Article> articles = inv2.getArticles();

        DefaultListModel<Item> listModel = new DefaultListModel<Item>();

        for(int i = 0; i < articles.size(); i++)
        {
          if(articles.get(i).getIsbn().equals(str))
          listModel.addElement(articles.get(i));
        }
        studentList.setModel(listModel);
      }
    }
}

// Back button to close the current window and
// go back to the main one

else if (e.getSource() == backButton)
{
  dispose();
}

// Lend button to lend an item to a customer;
// the user select the item wanted and pressing "Lend"
// the information of the customer, written in the fields
// are registered in the PersonList and together with the
// info about the item are added to the Borrow List and
// written in a file

else if (e.getSource() == lendButton)
{
  String firstName = textName.getText();
  String lastName = textSurname.getText();
  String email = textEmail.getText();
  String type = "Student";
  int day = Integer.parseInt(textDay.getText());
  int month = Integer.parseInt(textMonth.getText());
  int year = Integer.parseInt(textYear.getText());
  Item selection = (Item) studentList.getSelectedValue();

  MyDate date = new MyDate(day, month, year);
  Person person = new Person(firstName, lastName, email, type);
  Borrow borrow = new Borrow(selection, person, date);
```

```
    PersonList list = adapter2.getAllPersons();
    BorrowList list2 = adapter3.getAllBorrows();

    if (selection instanceof Book)
    {
      MyDate d = new MyDate(day, month, year);
      textDueDate.setText(d.dueStudBook().toString());
    }
    else
    {
      MyDate d = new MyDate(day, month, year);
      textDueDate.setText(d.dueStudArticle().toString());
    }

    if (list == null)
    {
      list = new PersonList();
    }

    if (person.getEmail().equals(email))
    {
      JOptionPane.showMessageDialog(null, "The user already exist in the system", "Warning",
      JOptionPane.WARNING_MESSAGE);
      textName.setText("");
      textSurname.setText("");
      textEmail.setText("");
    }
    else
    {
      list.addPerson(person);
      adapter2.savePersons(list);
      textName.setText("");
      textSurname.setText("");
      textEmail.setText("");
    }

    if (list2 == null)
    {
      list2 = new BorrowList();
    }
    list2.add(borrow);
    adapter3.saveBorrows(list2);
  }
}
```

```
  }

  public static void main(String[] args)
  {
    JFrame frame = new LendStudent();
    frame.setVisible(true);
  }
}
```

6. Testing

6.1. Testing methods according to Use case

Testing the program, the main purpose was to check every stage of the work, to validate the design of the GUI and check all the implementations, in order to be sure that everything works like it is supposed to do, or in the worst case scenario, to be able to identify the mistakes and the errors occur during the running of the program.

Add item

This function allows us to add items to the system, in this case allows adding all books and articles in possess by the library. The function is the base of the software since it's not possible to do any operation if the system is empty of items; however it's not the main function of the software.

Here a short description of the steps performed:

1. Choose if add a book or a article
2. Insert the information required
3. Press "Add" and add to the system

Lend item

This is the main function of the software since it's the main activity of a library. It allows lending books and articles to students or lecturers, storing the information in specific lists and files.

Books and articles can be searched in the system by title, author, magazine or ISBN

1. Select who to lend to (Student or lecturer)
2. Search the item you are looking for
3. Write the information of the borrower
4. Press "Lend" and store information

Return item

This function gives the possibility to return an item previously lent. The software should compare the date of the day which the customer is returning the item and the due returning date in

order to make pay a fine in case of late returning. Once returned, an item is removed from the BorrowList.

1. Search the item to return by ISBN
2. Select the item found
3. Press “Return” and delete from the file

Reserve item

The function allows the customers to reserve an item that is at the moment not available, to reserve it for when is going to be available again, after a previous borrower has returned it.

1. Search for the item
2. Write info of the reserving customer
3. Press “Reserve” and store the info in files

Search item

The function makes possible for the system to search for items and display all the ones that meet the searching criteria. Librarians have then the possibility to remove from the system some of them, if wanted.

1. Select searching criteria
2. Search the items wanted
3. Select one or more items and delete them (optional)

Below we are going to show a table with all the function and actions that the system is supposed to be able to accomplish; for each action, status and results will be shown.

Add item	Tested	Passed
Lend item	Tested	Passed
Write in customer list	Tested	Passed
Write in borrow list	Tested	Passed
Check if item exist	Tested	Passed
Check if available	Tested	Passed
Return item	Tested	Passed
Check due date	Tested	Passed
Reserve item	Tested	Passed
Write in reserve list	Tested	Passed
Check if lent	Tested	Passed
Search item	Tested	Passed
Check if in the list	Tested	Passed

Remove item	Tested	Passed
Message for returning	Not implemented	
CDs and DVDs	Not implemented	

The tests made on the classes show a general success in the methods we implemented.

6.2. Test of GUI

During the complete test using the Graphic User Interface we found out the GUI needed some modifications and in some cases some methods were not working as well as they was supposed to do.

Testing the GUI it required sometimes the addition of some feature like the Search button in order to submit to the system the text to search, since we didn't implement the possibility to achieve the same result with the Enter key.

7. Results

The software has been created in order to make easier the daily task of the librarians working in FairytaleLib in Ping Pong Mesa. While previously all the information where stored in excel sheets making the organization of all the data and information sometimes extremely confused, the aim of the software is make all the operations easier and faster.

The main purpose is to create an easy way to lend books, storing information of the customer that is borrowing and of the item that the customer is borrowing. Storing in files, the lists of persons, items, borrowings and reservations, it is easy for librarians to obtain all the information they need, whenever they need. The system offer an easy way to be update and always aware of the situation of interaction between them and the customers.

Furthermore the software gives an easy way to store new items and to remove in case of necessity, search following different criteria like searching by title or author or ISBN. Last, but as important as the possibility to lend, is the possibility to register the returned item.

The GUI offers an easy navigation, structured with a menu that open new different windows depending on the action selected. It might be improved, but being a beta version that has to meet only the requirements of a specific library it is able to perform its tasks.

7.1.Known errors in the system

Unfortunately we noticed there are some errors in the system. Most of them happen in the return window, where the ComboBox and the CheckBox are not doing anything but should work as information for the users, furthermore in the same windows, when searching for an item to return, it appears like there is a problem in the exact case that the borrow object was a lecturer borrowing an article, while in all the other cases it works fine.

Another known error is in the lending windows, the field “Available” that should shows the number of available items with something in common. It doesn’t do anything because it has not been implemented.

8. Conclusion

The main goal of our project was to create software for a library, a small one, run by only two librarians, providing electronic storage of information and with an easy to you interface, to make the software appealing and intuitive to use. The program is designed for the only use of the librarians.

Our goals were reached concerning the basic requirements we have been asked; even though we were not able to implement the “extra” ones, all the main requirements are complete and the system is ready to use.

Throughout the period which we have been working on the project, we faced different problems, most of those regarding Java code and in general about the logic behind it; behind the disposition of the elements on the GUI, since many times we had to change design, because of the understanding that in that way was not working properly.

9. References

9.1. Books

- Tony Gaddis, "Starting Out with Java early objects", 3rd edition

9.2. Additional Resources

- Presentations from classes
- JAVA 1 and UML (SDJ1)

10. Appendix

10.1. Appendix 1 – Use Case Description

Add Item

Use case	Adding item
Summary	Librarians can add items like articles and books for either students or lecturers.
Actor	Librarians
Precondition	Who use the software needs to have access
Postcondition	Librarians have written the information of the items in the system. The item is added to the system
Base sequence	1. Librarian check what item is 2. Librarians register items information in the system 3. Check if item already in the list (Use case: "Check if existing") 4. Press "add" 5. The item is added to the system 6. The software displays " The item is added" message
Exception sequence	Item already in the system: 1-4 As base sequence 5. Add 1 unit to quantity of item already existing 6. The item is added to the system 7. The software displays "The item is added" message Use case ends
Sub Use Case	Check if already existing
Note	

Use case	Lend item
Summary	Librarians can lend items to either students and lecturers for a certain period depending on who is the borrower and which item is lent
Actor	Librarians
Precondition	The item required has to be in possess of the library
Postcondition	The item has been let to the customer which it has been assigned a due date to return the item
Base sequence	1. Librarian search the item asked 2. Librarian check if the item exist (Use case: "Check if existing") 3. Librarian check if the item is available (Use case: "Check if available") 4. Librarian registers in the system customer's information such as first name, last name and email address. 5. Software assign a due date depending on the borrower and the item 6. The item is marked as lent 7. The item is added to the list of items lent 8. Software print "Item lent successfully"
Exception sequence	Item doesn't exist in the library: 1-2 As base sequence 3. Software printing "Item doesn't exist" Use case ends Item not available: 1-3 As base sequence 3. Software printing "Item it's available for reservation" Use case ends
Sub UseCase	Check if exist; Check if available
Note	At the end of the exception that see the item not available but existing, the customer will be able to reserve it, moving in the diagram for reservation

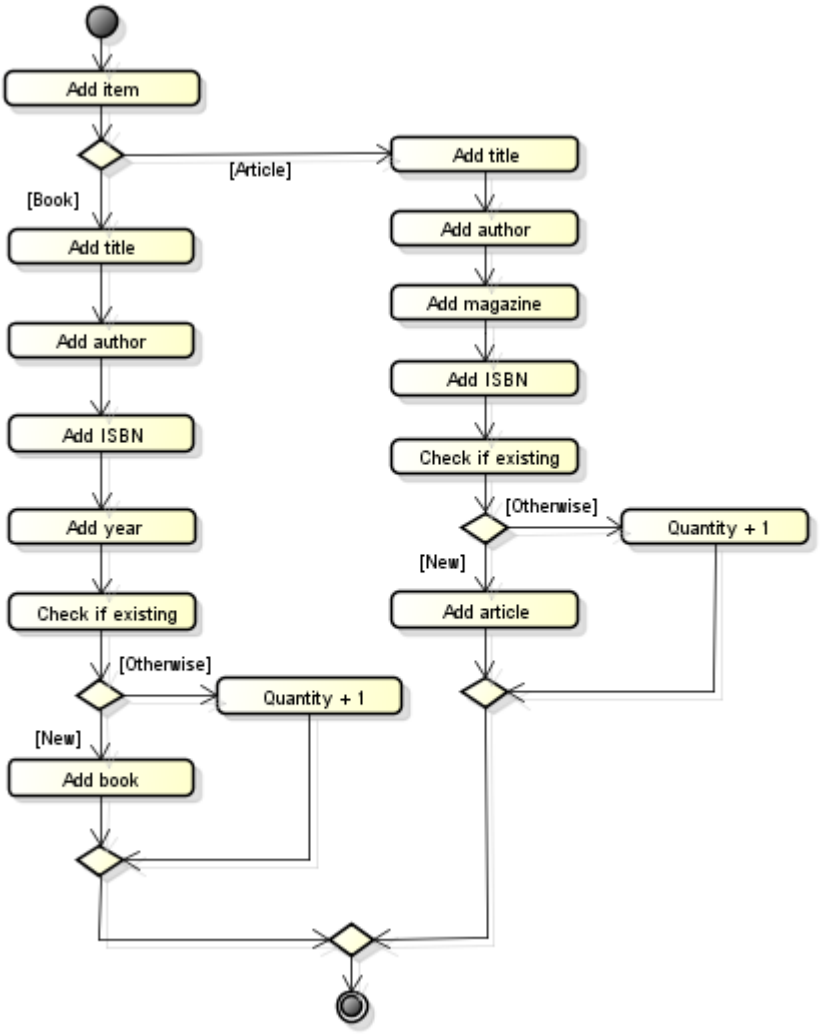
Use case	Return item
Summary	Librarians can annotate the returned items previously lent, in the system
Actor	Librarians
Precondition	The customer borrowed an item at the library
Postcondition	Librarians have written in the system that the item has been returned.
Base sequence	9. Librarian receive the item 10. Librarian check the lent list 11. Librarian check if on time (Use case: Check due date) 12. Librarian delete customer information 13. Press “Returned” 14. System set item available
Exception sequence	Item not on time: 1-3 As base sequence 1. Print “Returned late” 2. Librarian ask the payment of the fine 3. Press “Returned” 4. System set item available Use case ends
Sub UseCase	Check due date
Note	

Use case	Reserve item
Summary	Librarians can reserve an item for either students or lecturers in case the one they need is already lent out.
Actor	Librarians
Precondition	The customer is part of the school involved in the system
Postcondition	Librarians have written the information of the customer that wants to reserve the item
Base sequence	15. Librarian search the title asked 16. Librarian check if the item is available (Use case: "Check if lent") 17. Librarian registers in the system customer's information such as first name, last name and email address. 18. Press "Reserve" 19. The software display "Successfully reserved" message
Exception sequence	Item available: 1. As base sequence 2. Software displays "The item is already available" Use case ends
Sub UseCase	Check if lent
Note	At the end of the use case in the exception sequence, the sequence is moved to "lending"

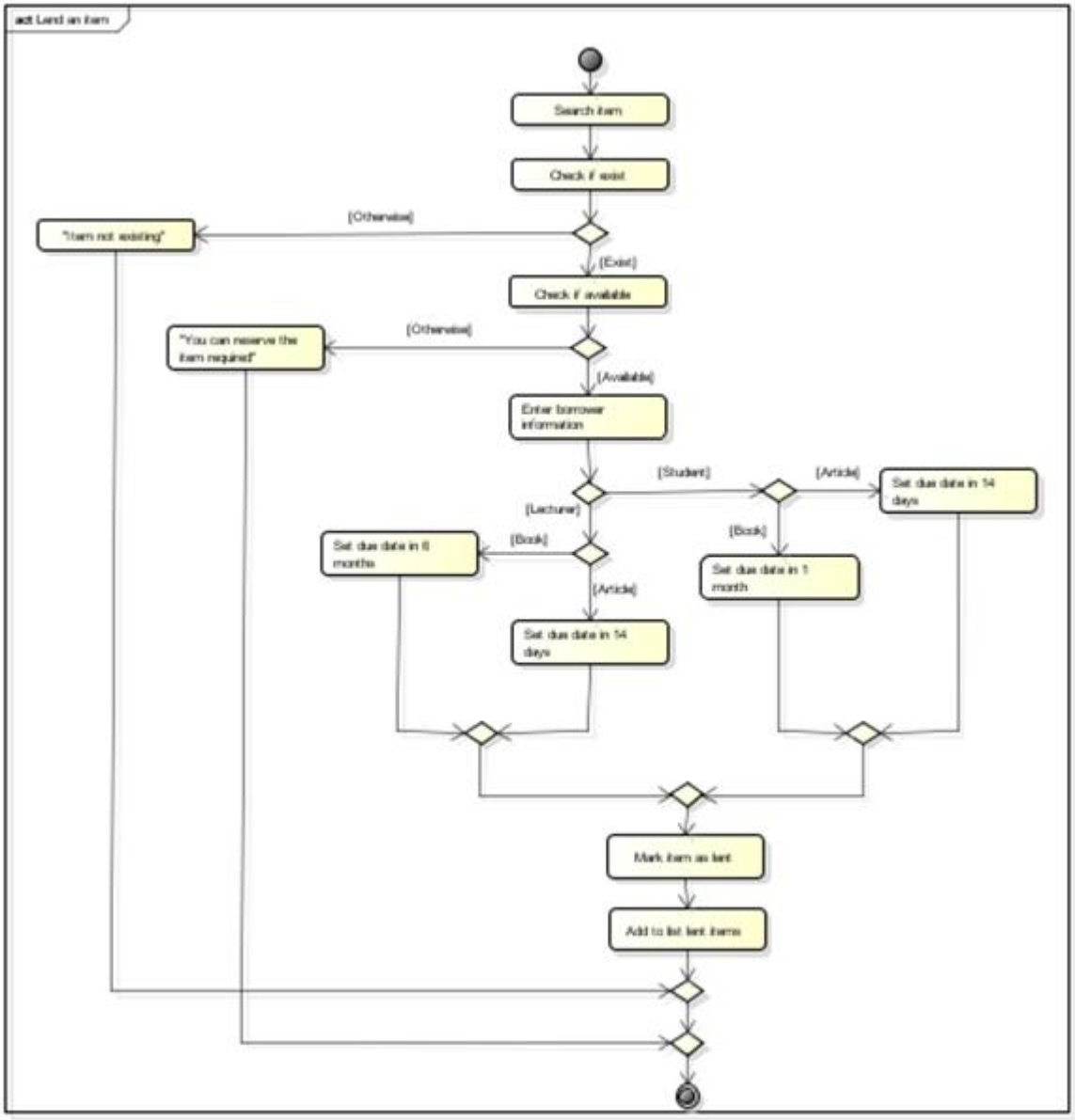
Use case	Search
Summary	Librarians can search for the items stored in files with the possibility to delete them
Actor	Librarians
Precondition	The list where items are stored is not empty
Postcondition	Librarians see the list and they remove in case they need
Base sequence	20. Librarian decides what to search for: (title, author, etc.) 21. Librarian press "Search" 22. The list shows the result of the research Optional: 23. Librarian select an item 24. Librarian press "Remove" 25. The items is removed from the list
Exception sequence	Empty list 1-2 As base sequence 5. System shows "There are no items in the list" Use case ends
Sub UseCase	Check if in the list, Remove
Note	

10.2. Appendix 2 – Activity Diagram

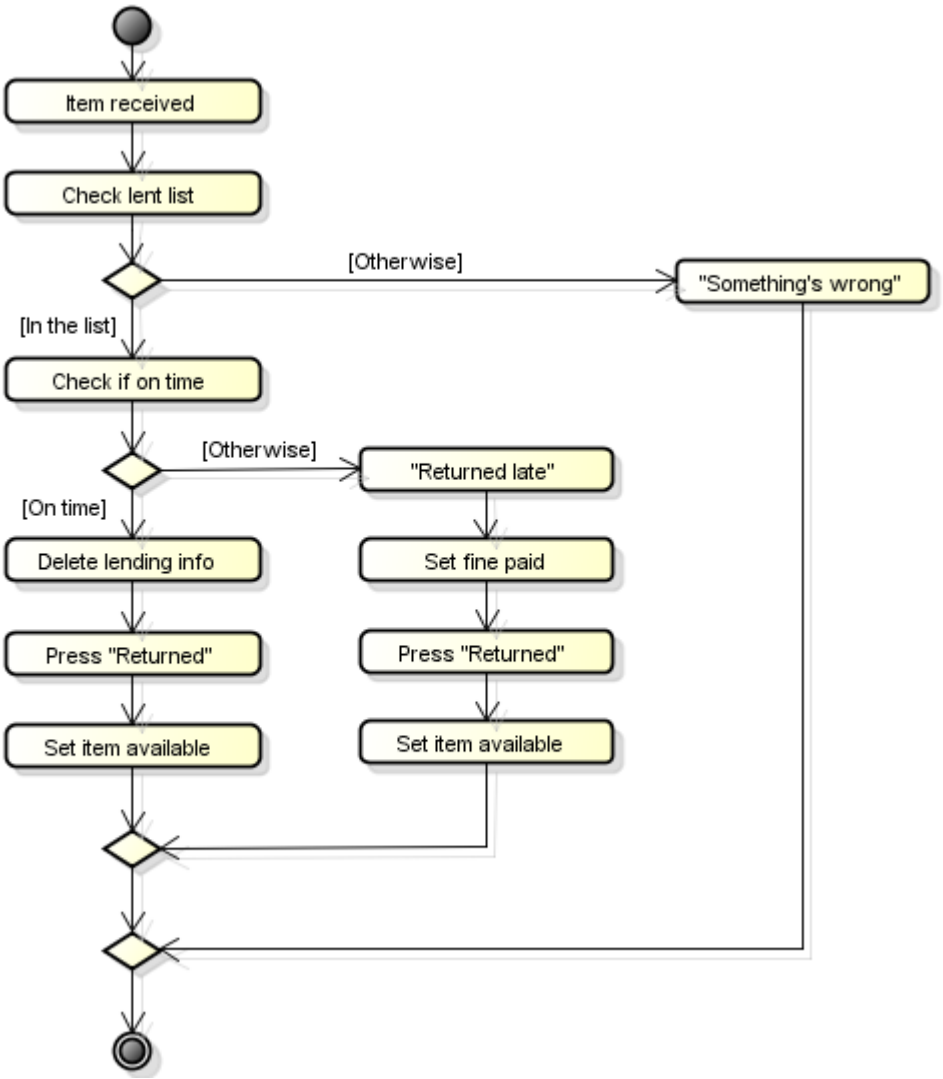
Add Item



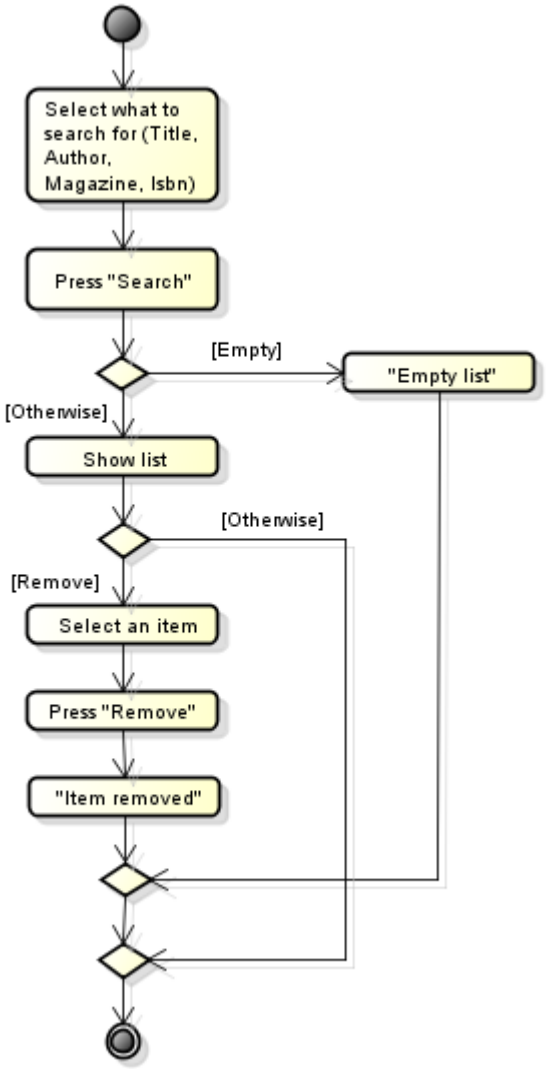
Lend Item


act Lend an item
Search item
Check if exist
[Otherwise]
"Item not existing"
[Exist]
Check if available
[Otherwise]
"You can reserve the item required"
[Available]
Enter borrower information
[Student]
[Article]
Set due date in 14 days
[Lecturer]
[Book]
Set due date in 6 months
[Article]
Set due date in 14 days
[Book]
Set due date in 1 month
Mark item as lent
Add to list lent items

Return Item


Item received
Check lent list
[Otherwise]
"Something's wrong"
[In the list]
Check if on time
[Otherwise]
"Returned late"
[On time]
Delete lending info
Set fine paid
Press "Returned"
Press "Returned"
Set item available
Set item available

Search Item


Select what to search for (Title, Author, Magazine, Isbn)
Press "Search"
[Empty]
"Empty list"
[Otherwise]
Show list
[Otherwise]
[Remove]
Select an item
Press "Remove"
"Item removed"

USER'S MANUAL

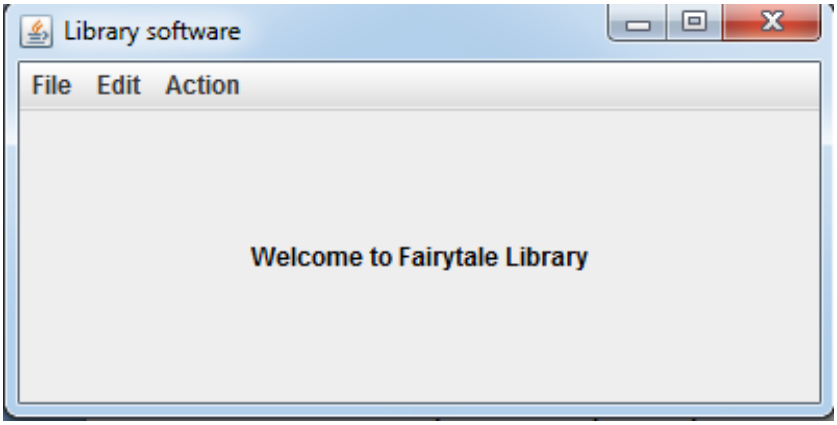
Fairytale Library Administrative Software

Software Version: 1.0

Software developers: Raffaele Brivio, Javeed Mahmoudi, Alin G. Cortel

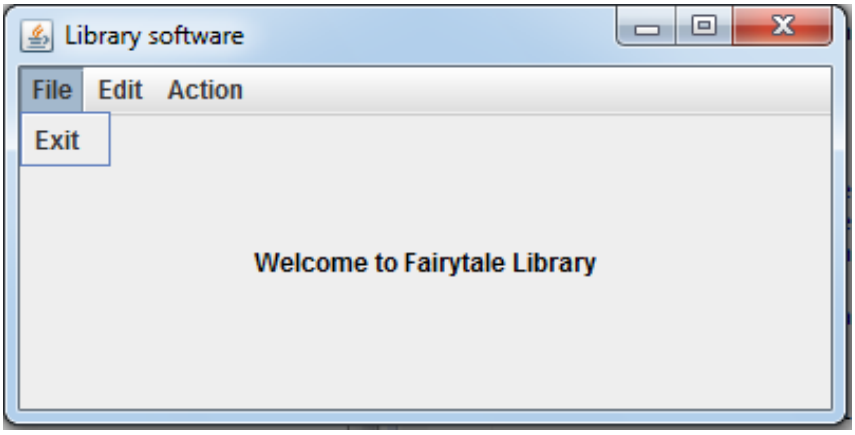
1. General Information

The current software has been developed in accordance to requirements of which librarians assessed as in vital in order to manage a library. All functions required by librarians, will be accessed through a Graphical User Interface, G.U.I, which it will be covered in current user's manual. Overview of home framework consists of title of software."Library software" and a menu bar which has the following options to choose from "File", "Edit" and "Action".



2. Software Usability

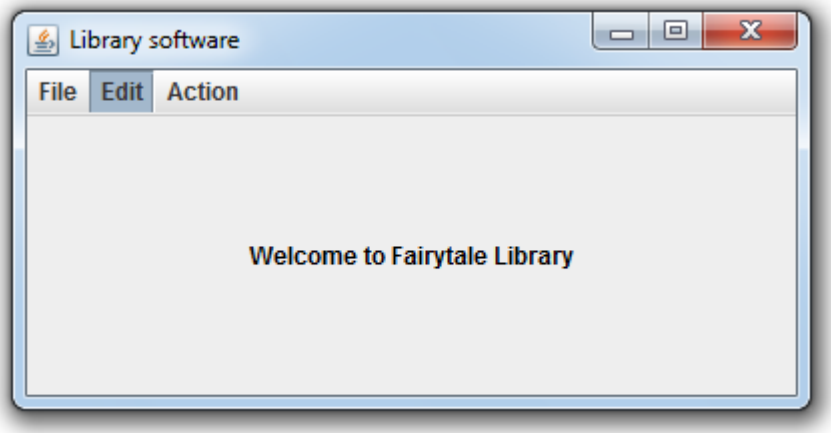
2.1 File



As seen in the G.U.I, librarians are able to select the "File" bar, and have the option to select Exit, and close the program.

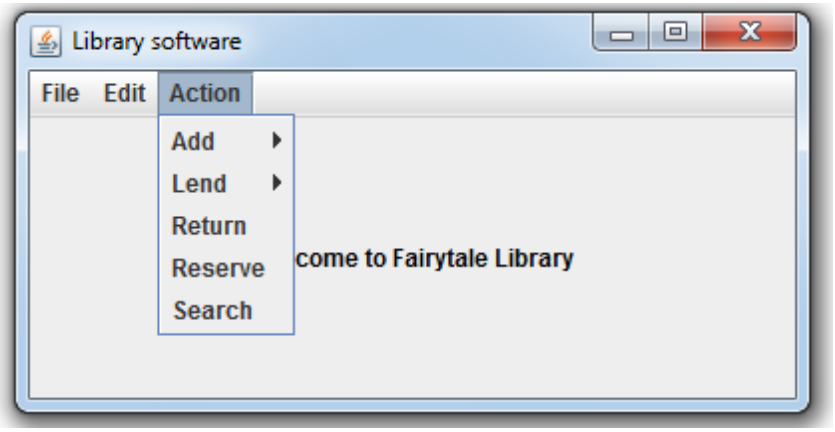
2.2 Edit

When selecting "Edit", librarians should be capable of editing the font of menu and colors of background. The current functionality is not implemented for this version because this option is meant to be available for users that pay extra, to receive this customizable G.U.I, along with updates.



2.3 Action Function

The action submenu is composed of these current Functions: Add Lend, Return, Reserve and Search. Each function will be explained further more in greater detail.



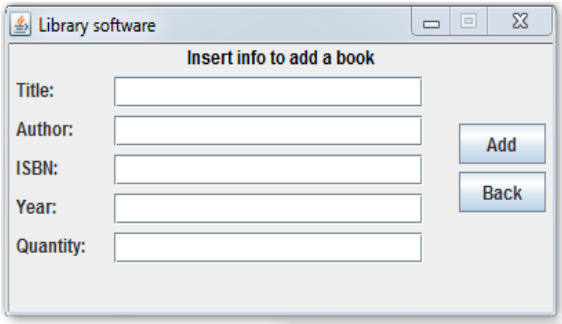
2.4 Add Function

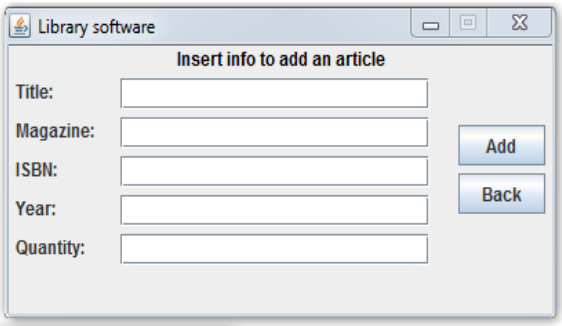
"Add "submenu consists of choosing which type of item you wish to add. Currently, you can choose between Book and Article.

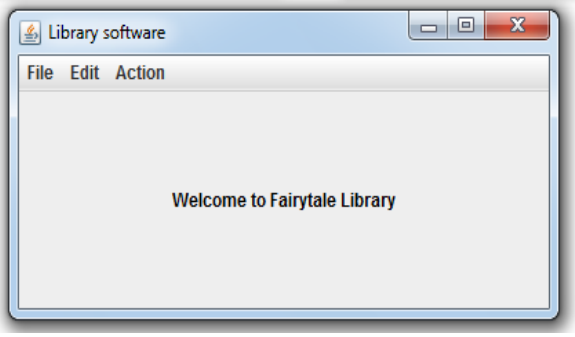
By selecting adding a book, (Picture on the Left Side), you will have the menu for choosing what information is associated with added book, information such as: Title, Author, ISBN, Year and Quantity.

By selecting adding an Article, (Picture on the Right Side), you will have the menu for choosing what information is associated with added book, information such as: Title, Magazine, ISBN, Year and Quantity.

You also have the option to add simultaneously, more than one book or article; a Pop-UP menu will not cancel the previous one.

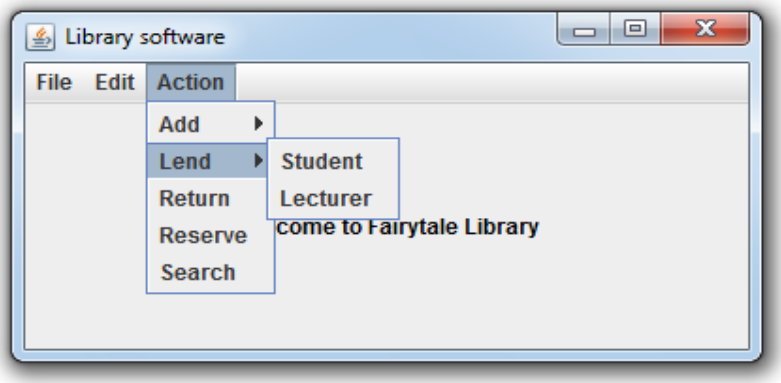




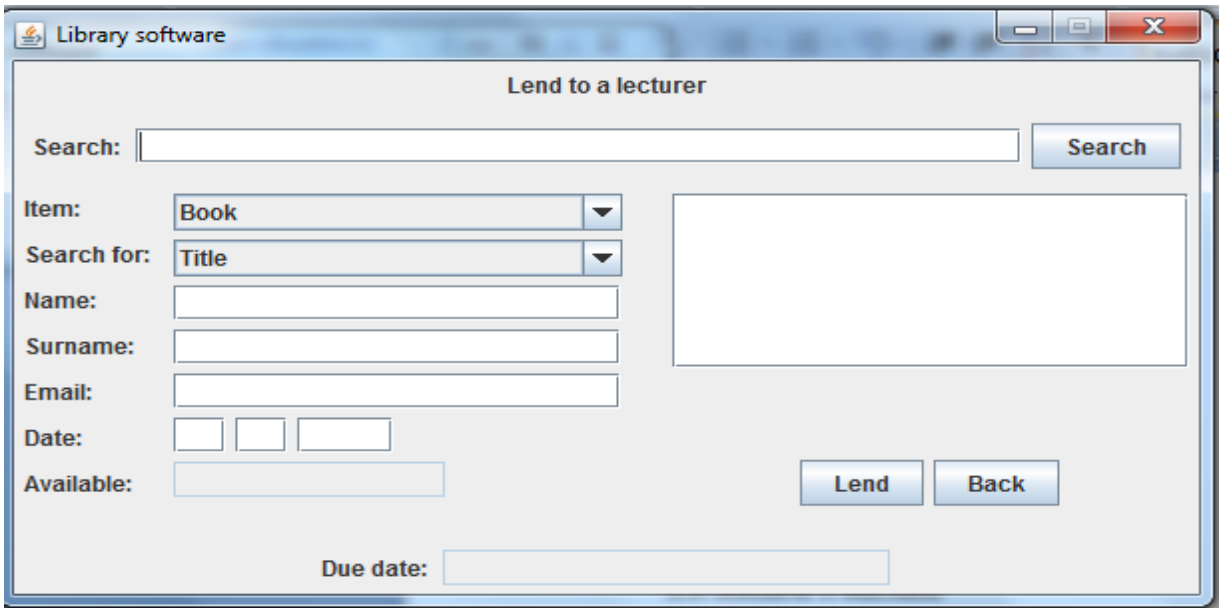


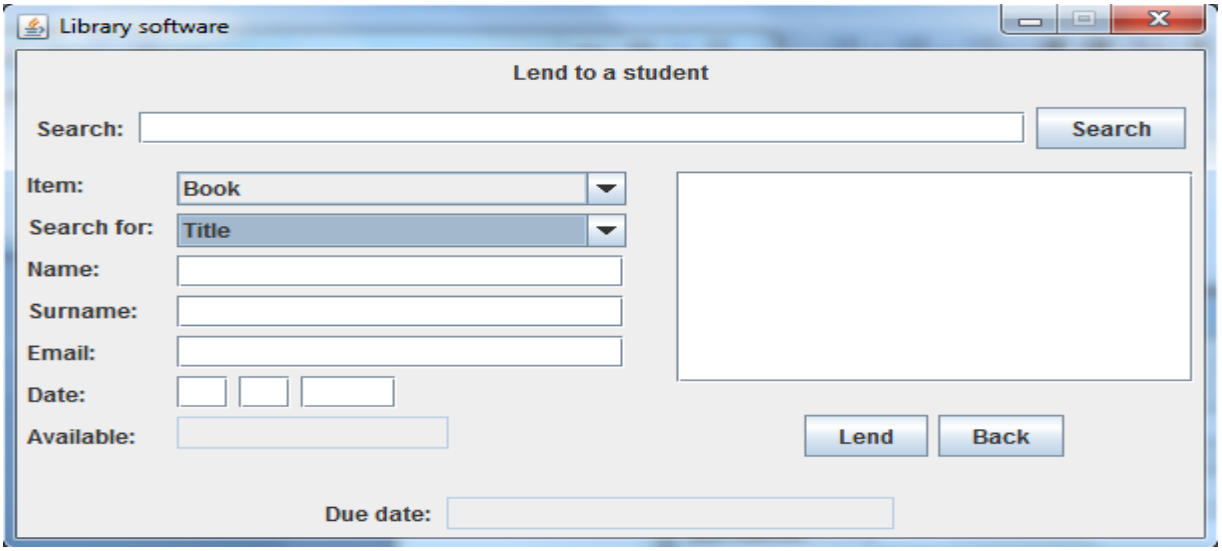
2.5 Lend Function

Lending submenu consists of whom librarians wish to lend items too; in this case, it is taken in consideration if it is either a student or a lecturer. By selecting either lending to a student or lecturer, another frame will appear which consists of multiple options. One of these options is a search function for which item is wished to be borrowed. The search function can be done based on Title, Author, Magazine and ISBN.



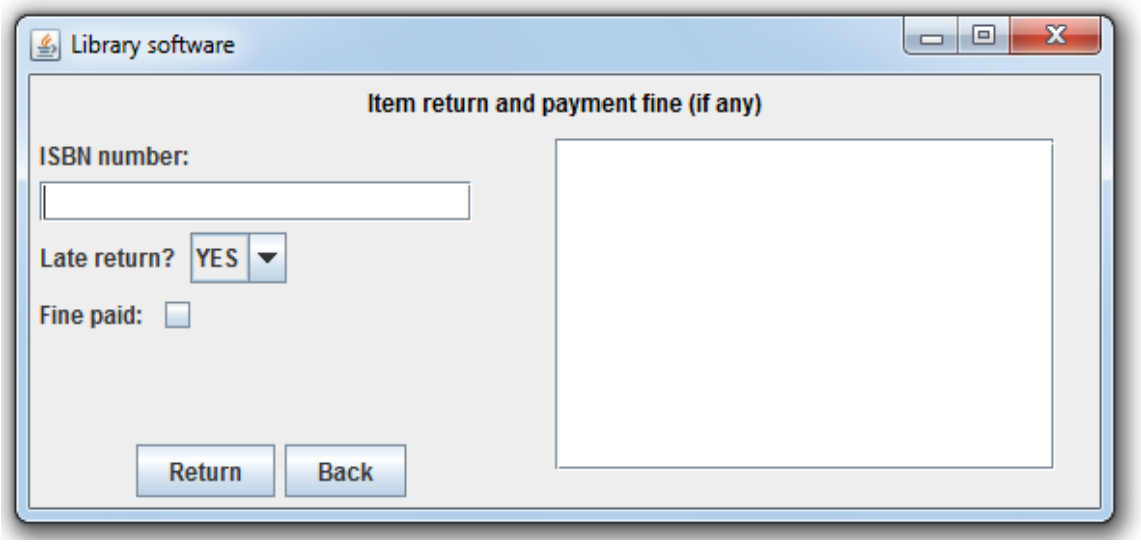
Down, there is Name, Surname, Email, Date information regarding person whom borrowed that needs to be added to lent item
Another option is related to which type of item is borrowed, if it is a Book, or an Article. The Availability of items is shown, if quantity of items is bigger than 0.
Besides these functions, there is the option to see the due date, and the option to see the list of items searched on the right. On the right, we also have the "Lend" and "Back" buttons, for lending and returning to previous "Home" interface.





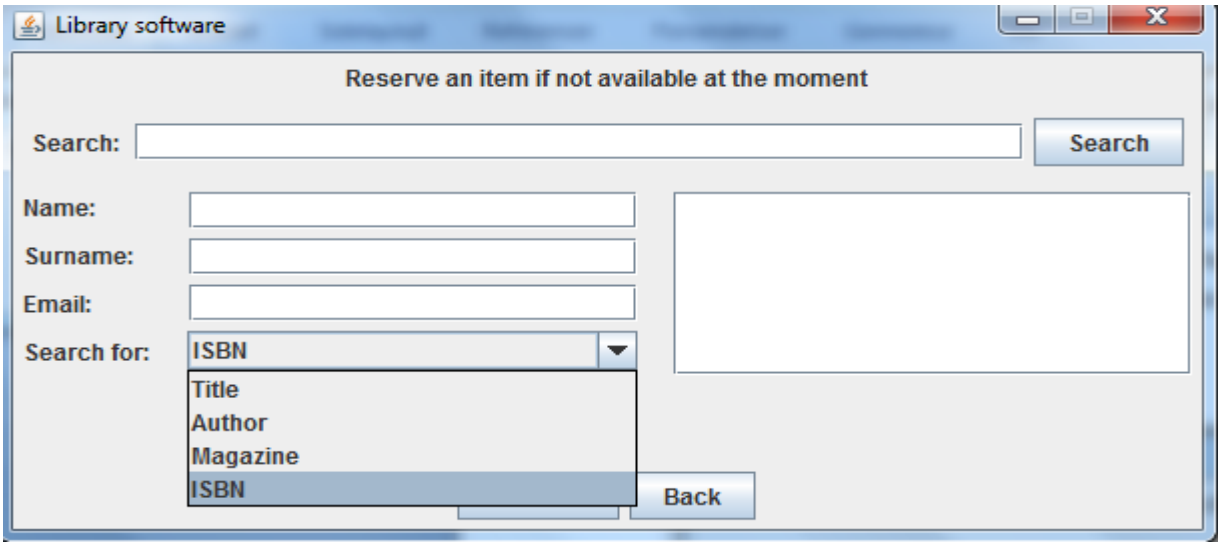
2.6 Return Function

The return function is meant so librarians are able to mark as returned items brought back from students or lecturers. The item is being added back in the system as returned by adding its ISBN number. Based on the fact of the item being returned late or not, there is a box that must be checked if the fine was paid or not, in case the item is late returned.
We have the buttons “Return” and “Back” meant for returning the selected item and returning to previous home interface. On the right, there is the list of items.



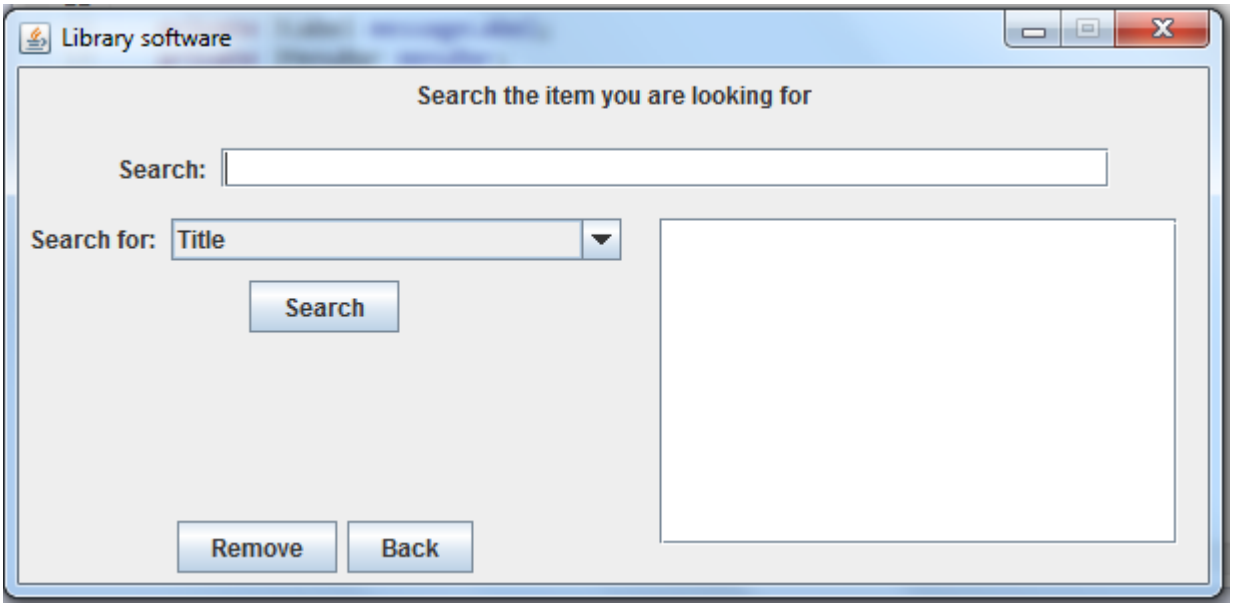
2.7 Reserve Function

When reserving an item, a frame appears with the option to search the item you wish to reserve, by title, magazine, author and ISBN. There is also the option to add the Name, Surname and Email of the person who wishes to reserve. On the right, there is the panel where you have the list of reserved items. Last buttons are “Reserve” and “Back”, which have the purpose of adding to reserve list the selected item, attached to it information of person who reserves and the other button returns the librarians to the home interface

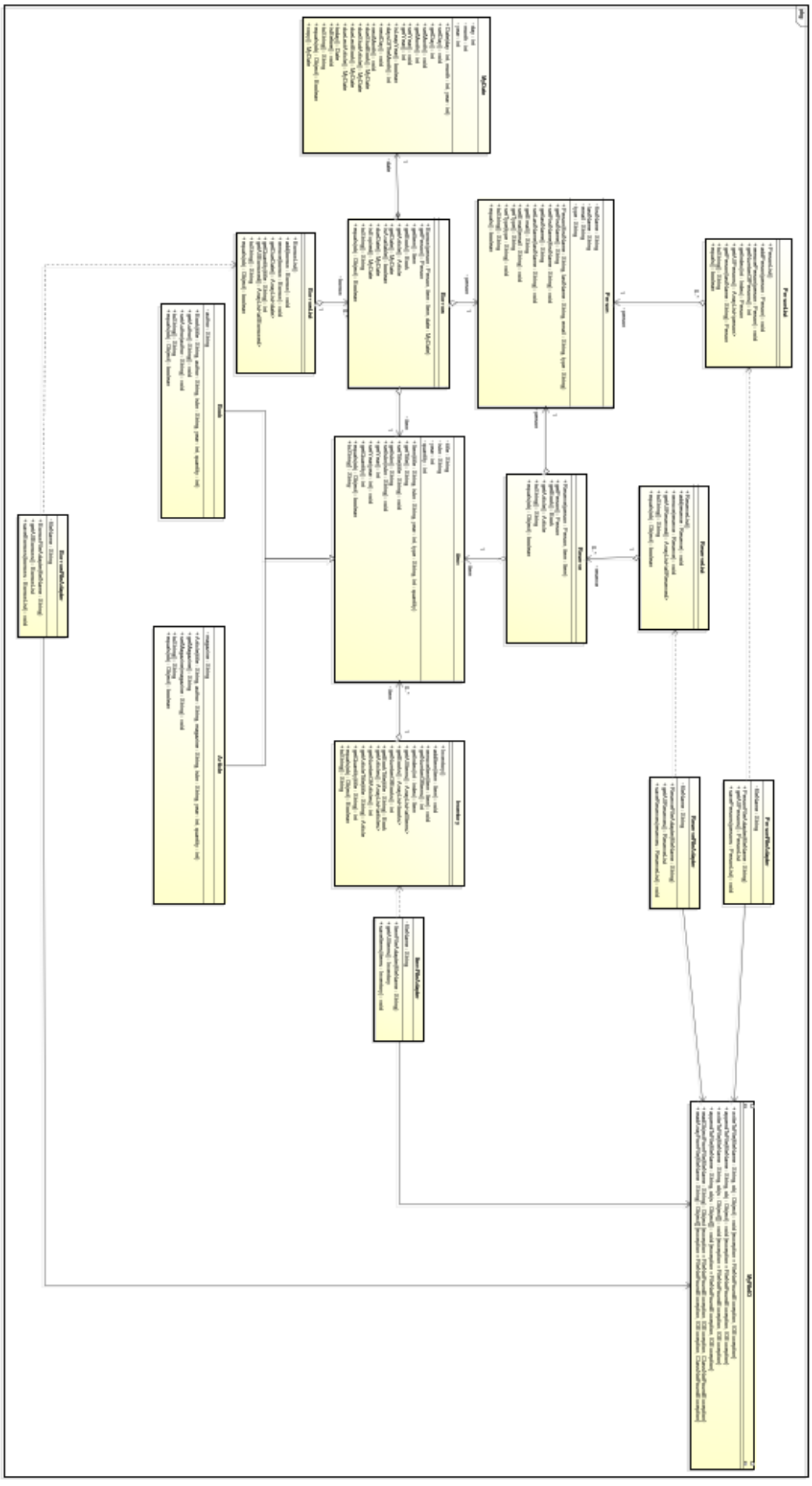


2.8 Search Function

The search function is meant to search in inventory, by Title, Author, Magazine, and ISBN; the window gives also the possibility to remove an item from the system in case needed.



10.4. Appendix 4 – Extended Class Diagram

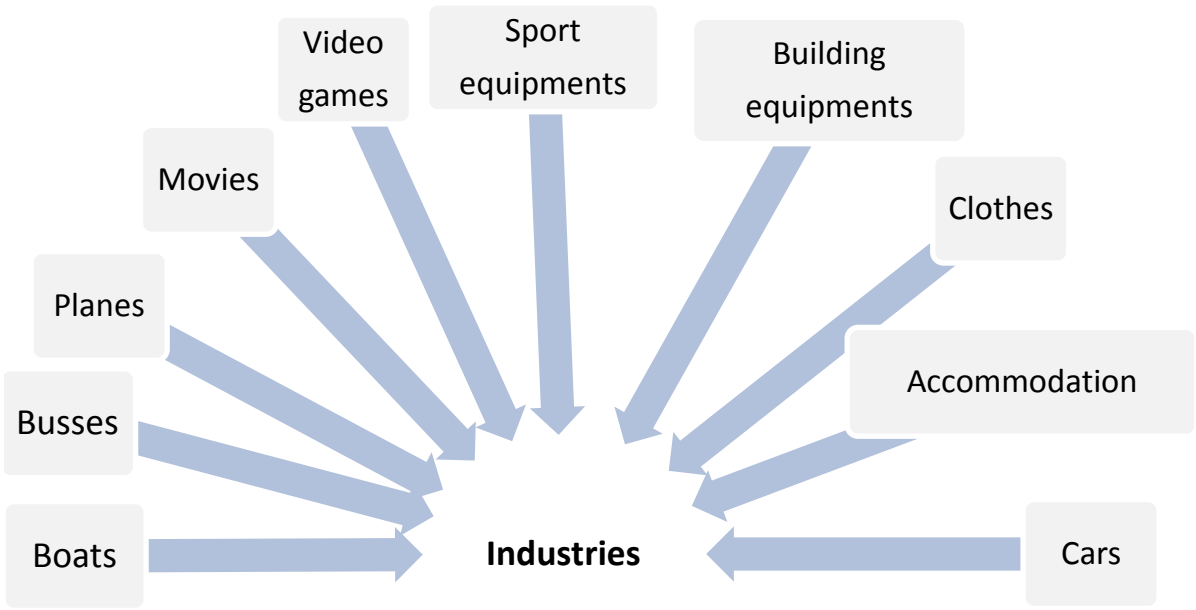


11. Industries

In order to see if there is the possibility for the system developed by us, to be expanded in other sectors, we need to undertake an analysis; looking at the general situation of the market and in particular looking for industries that might share the same basic idea, the same basic way in storing rental information than our system.

To find out which industries might be the most suitable for us, the first approach we used was to take a general look in the entire market and to think which industries work in a way similar to a library. The answer is that in general, in every industry, each business that has as main activity to rent some goods, have the potential to be analyzed by us.

The different markets we took into consideration at first, to have an idea of the potential number of them, without excluding any, were the most various; from car, busses, boats and planes rental, to other areas like videogames, movie, accommodations or sport equipments. Including also particular ones like clothes and building equipment.

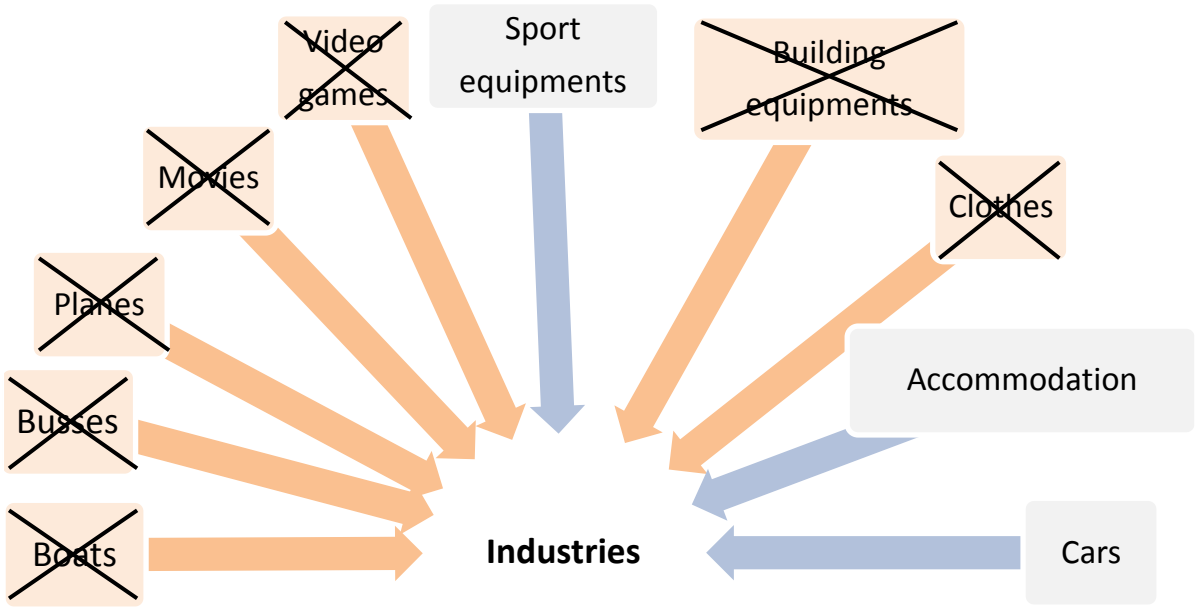


Starting from that point, we tried to figure out which among those, are the most suitable for our research. Assuming that we would like to maintain the structure of our software as much as possible similar to the original one, with the possibility to change features and functions but trying to reduce those at the minimum; we tried to focus more on industries that for characteristics, needs and operations are the most similar to us.

Having in mind the characteristic of our software, where we can store information of various items, that customers can lend or reserve for a fixed period of time, paying a fine in case of late returning, we found ourselves to exclude in advance some of the industries we took into consideration at the beginning, because of the differences from our starting system.

Some like busses, airplanes, boats and clothes have been excluded because in a first place those are not that common, there are shops used to that, but are particular shops and with a limited potential due to limited customers. Building equipment has been excluded because even though there are companies renting those, the target is construction companies that rent the equipment they need for the period that works last; there is not reservation, differentiation of items, fines in case of late payment. Basically it's quite different from our original system. Between our other choices there are the entertaining games that is common to rent in particular during parties, but that business differs from our mainly because of the fact that it involves delivery with relatively costs for the company to keep track on the system; these kind of games usually require also high cautions making the software required, more particular and complex. For last, about videogames and movie for the modality in which they are rented in shops, the software needed to keep track of those might be really similar to the one already developed for the library; unfortunately, if we pause a moment to focus on the potential in that market, we can easily see that in the past years the rents of movies and videogames decreased considerably due to the online piracy, that play a big threat for businesses already in the market making it risky and unattractive for us.

The three last choices remained are cars, sport equipment and accommodation rental. All those three are appealing to us since they all share some basic characteristic that also our system has.



In case of car rentals, every rental shop will have different models of car, or maybe motorcycles also and customers will have the possibility to rent or reserve for a specific day or period. You can

see here the similarities from the software we have for the library; in the same way customers will have to pay a fine in case of late return. The main differences are the flexible period which people can rent the car and the possibility for the renter to ask for an initial caution.

About of sports equipments there might be shops renting those equipments for every kind of sports, for example if you go in the mountain there will be shops renting all the equipments for skiing or at the see for surfing. With these kinds of equipments the main difference is the length of the rent; usually it doesn't extend more than a week and usually is not allowed to reserve equipments, but it mostly depends on the shop. Despite the previous differences, all the others main functionalities are the same as for in a library.

Talking about the last of the three possible industries taking into account from us, accommodation rental, also in that case some of the characteristics are really similar to the one needed to manage items in a library. Having like our potential customers hotels, bed and breakfast, hostels etc. all this, usually need a system to be able to keep track to their customers; having different kind of rooms, customers can book for an undetermined period of time.

The main difference to software used to manage a library is that if the customer pay, he can potentially keep the rented room forever; there is not a fixed or a maximum amount of time that he can keep it. Even if not common, also for hotels or any other kind of accommodation is possible to rent without booking or reserve even if it's not common because the risk that everything is taken is usually high.

Despite the assumptions made so far until this point, deeper analyses have to be made in order to clear what is better for us to be able to find a suitable possibility to introduce our product in the market.

The three choices of markets we find suitable for a possible penetration in the market with our software, basing on the depth of analyses made so far, they are equally suitable. All of them are really similar to our already existing software and all with small and easy modifications and implementations to do on the software.

At this point we would be able to choose one of the three with no difference, but we would like to go more in depth with the analyses and see if one of them reveals more advantageous than the other two; for this reason we will leave the decision after further investigations.

12. Segmentation

A general way to understand where we want to place our product is to divide the market into segments and analyze which might be the most appealing for us, considering the characteristics of our product, the dimension that we can reach and the nature of the segments studied.

Basing our division on the software previously developed for the local library, the first observation, related on the geographical segmentation, is that we want to focus on a small area. The best for our software is to be limited to a maximum extension within the boarder Denmark, without entering the international market.

An additional reason why Denmark as a country is suitable for the product we offer is cultural. Denmark is known to be a vanguard country, one of the most innovative countries in the world and we believe that software to automate registrations in different industries is well welcomed.

Another important delimitation is that we want to remain into the B2B, that will allow us to remain focused in the sector we will choose, while in case of starting developing for privates, would means to create an entire new software each time, potentially never related to our original one.

Said that, we should focus on small businesses where is not required the use of complex databases for the number of information stored, but that the storage in files is enough to serve the purpose without slowing down or complicate the use of the product; in case the use of databases is strictly necessary, we will consider to implement it.

Examples related on the previous mentioned possible industries are, to focus on single rental shops, where we would have the possibility to adjust the software according to the request of the owners of the shops. Or single sport equipments rental shops, specialized in lending equipment for only a determined sport. Or again, in case of accommodations rental the focus would be on local Hotel with the possibility to involve also private owners that own Bed & Breakfast.

To summarize, the most suitable segments in which we would like to focus on are small and local businesses within the borders of Denmark.

13. Porter's 5 forces

At this point we need to analyze the micro environment that surrounds those industries we looked at, in order to understand which is the real potential of the market and to figure out which is the most suitable for us or in case, to figure out that more than one is suitable to try the penetration.

For this purpose we will undertake the one known as Porter's 5 forces analysis that, taking a close look into threats of new entrants, threats of substitute products or services, bargaining power of customers, bargaining power of suppliers and intensity of competitive rivalry.

Threat of new entrants

We will start looking at the threats of new entrants, analyzing the three industries we are working on, under this aspect. The analysis of the possibility of new entrants is going to be made at the same time for cars, accommodation and sport equipment; the reason is that for the purposes of our system, those three industries are similar.

In an age where the IT sector is one of the most advanced among all, companies learned to develop new systems with a really high pace, often updating them, increasing the features or improving the functionalities. The market is really active, characterized but a huge differentiation of products.

The barriers that a new business can meet are especially in case if the business doesn't exist yet; in case of Denmark the costs to meet are to start the business and to have the CVR. The costs to start a business in Denmark are really small compared to other countries all over the world. The barriers a new business can meet usually are found out when the business is still running and those are the high amount of taxes to pay annually and the high salary to pay the workers. Those last two have to be analyzed carefully because even though as just said, the general costs to start a business are low, taxes and salary can act as an annoying barrier if not took into considerations carefully in advance.

The costs for developing a system are various, it depends on the software required by the customer and it can range from a really simple and cheap one to the most complex, updated, more expensive one; this make possible to a big number of potential entrants to approach the market since the possibility of deciding how much to spend in it.

Although in this kind of market, customers, or in our case other businesses, tend to become loyal to certain suppliers, avoiding to change software often, having then to re-train the workers to use the different ones, the threat of new entrants, for all the industries is really high.

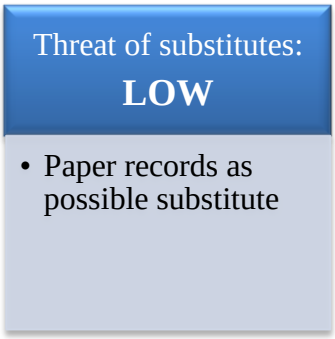
Threat of substitutes

Moving on the threat of substitute products, here we need to be careful on what we analyze; thinking about a general system that allow businesses to store information about renting, the potential substitute are all the other existing systems with that purpose and therefore, a really big amount, including also online software. However our idea is to aim at small businesses, with a close relation of cooperation; we would like to be in the segment where businesses ask for customized software, where we can meet all their requirements, creating a bond between us and our customers with loyalty towards and from them.

With this in mind, once a customer is asking for a personalized system for their business, they generally develop a lasting relationship with the supplier, not easily willing to change, having to bare new costs for the new system and the training to users.

Just to mention the possibility, a possible substitute, for really small businesses with scarce resources to invest in software, is to keep track of what they need in paper records. Although this might be a possibility, it's really old-fashioned; it's maybe possible to find it in really old businesses or extremely small, but in general, in an innovative country like Denmark it's really hard to find that solution.

For the reasons just analyzed, the threats of substitute's products, according to our product and the segment we would like to focus in, it's relatively low.



Bargaining power of buyers

Now it's time to analyze the bargaining power of the customers and in this case we need to differentiate a bit between the 3 sectors we are looking into. As first factor to take into account, there is the number of choices, of different possibilities that buyers have; here we can already differentiate the chosen sectors.

Without forgetting the choice on focusing in a restrict area of the market, with a maximum expansion to the whole country of Denmark, with regard to car rental shops and accommodation rental, the number of our potential customers is high, usually many in every city, making for our business relatively easy to find customers to serve, being able to face the competition without expecting major obstacles.

On the contrary, regarding to sports equipment rental, it's a less common business, at least compared with the other two; the presence of shops in the Danish territory is not as widespread as the other two sectors.

As mentioned before analyzing the threat of substitutes, customers are not that used in this market to switch often to other, different products; one of the reason is that regarding software, usually, the main cost is paid at the beginning while once bought, the subsequent costs are relatively small. It is true however, that new potential customers, or customers already in the market that have to

purchase a new software, they find themselves with a huge choice, due to the large number of software developer in the market; this, independently on which of the three different industries we talk about.

Due to the factors just seen, we can conclude that regarding the car rental and accommodation rental, the bargaining power of buyers is moderate, balanced from the large number of potential buyers on the market on one side, and from the large choice the buyers have on the other side; while regarding the sports equipment rental the bargaining power of buyer is higher compared to the others two industries.

Bargaining power of customers:

MODERATE (cars, accommodation)

HIGH (sport equipment)

- Large number of customers (cars, accommodation)
- Restrict number of customers (sport equipment
- Loyal
- Have a large choice in different suppliers
- Have more costs in changing suppliers

Bargaining power of suppliers

Analyzing the bargaining power of suppliers, we need to think that to develop software, the "raw materials" are limited to the workforce and the processor used to program. This analysis is valid at the same way for all the three industries from us considered.

Starting from the assumption that processors, computers to program, to create software, are largely available of every prices, every kind, with every possible features in the market and sold indistinctly to private and businesses, the bargaining power of those over customers is virtually null.

The other fundamental element to develop software is the labor. Even though it required a specific education, the IT sector saw an expansion in the last years, with an increasing numbers of workforce specialized. Even analyzing this factor, we can see that for a company it's relatively easy to replace the workers in case of need.

To summarize, the general bargaining power of suppliers toward a business of software developers, it's generally low.

Bargaining power of suppliers:

LOW

- Large number of different suppliers
- The sector require skilled labor
- It is not hard to replace the labor

Intensity of competitive rivalry

Talking about competition, as mentioned in a precedent section, the market is full of software developers of every kind, also of rental software like ours.

There is a general strong competition in this kind of market due to different factors. First of all we need to consider that software are more and more part of the everyday life of people and businesses in particular; making any kind of tasks easier compared with the past.

In the market we can find big software developers that deal with different kinds of software at the same time, meeting the demand of different segment of the market, or software developers specialized in particular kind of software, like us.

Every business in this market has to face a close competition from all the others due to different factors. The most important ones, where the competition is the strongest are related to price and innovation. Regarding the price, for every need you have there are infinite different software that will allow you to accomplish your task, from the most expensive ones with all the possible features, to the free ones you can find online that allow you to do only the basic operations. It is important for a business, in this case, to understand which position it wants to be positioned at; depending of the nature of the business and the level of knowledge among the workforce.

Regarding innovation, the IT market in general, is one of the most active markets in the world and innovation is the key to be competitive and to survive in it. A business can decide to react to innovations of other businesses to adapt their own product or they can be a proactive business, being the one introducing the innovation in the market, forcing the other to adapt to keep pace with all the others. Whatever is the choice of the business, it's not possible not to innovate, adapt and add features and functions to these kinds of products, pretending to remain competitive in the market.

Although usually there might be loyalty from the customers towards a business, they would like to be always at the same level of the others, requiring innovative software when possible. It is easy to understand that innovation is a key factor.

To conclude the analysis, all we saw here about the competition, it applies for all the three industries we are studying in the same way. We can see from what said that the businesses in this sector have to face a high level of competition.

Intensity of competitive rivalry:

HIGH

- Large amount of competitors
- Products from every range of price
- High presence of innovation

14. Target group

At this point of the analyses, we have a broader view of the market and we've been able to analyze some key factors and aspects of the market in a deeper way.

Thanks to the information we gathered looking at the threat of the new entrants on the segments of the market studied by us, at the threat of the possible substitutes that might push customers to choose a different, maybe cheaper product, at the power the buyers have on the market towards us, at the bargaining power of suppliers that might influence our prices and our freedom of choice in case too influent on us, and at the rivalry in general in the chosen market; we are now able to decide which is the target group we aim to sell our product to, the one where the potential is the highest obtainable for us.

We started looking into the market in his whole, starting from all the industries that we thought were suitable for us and for our product; narrowing down the choice, selecting only the most similar to the criteria we want to satisfy.

We ended up with three possible industries to take into consideration in a deeper way in our analyses and at that point we studied, which were the segments our business is interested into entering the market.

After careful analysis of the Porter's five forces about the three industries we chose, related to the segments of the market we want to focus on, we can see from the result that the market is promising and there are the possibilities to enter the market and develop a strategy to do it at best.

However, even though the condition to enter the market are present in all the three different industries, we can see that in particular regarding the bargaining power of buyers, the sport equipments industry is slightly in a position of disadvantage compared to the other two.

It is for this reason that we believe that, in order to focus only where the possibilities are the best, we should exclude the market involving sport equipments rental. On the other side, because of the good results of the analysis and because of the high competition in particular, we decided that the best option is to develop a strategy to enter the both remaining two industries.

This choice was dictated by the fact that since we want to focus on small customers that require customized solution and as already said, in a sector with high competition, being present in different markets allow us to take a smaller risk in case of market saturation in one of the two sectors; moreover, since, as previously analyzed, the industries are really similar concerning the software that might require, there is the possibility to save costs, developing systems with only small differences. In this way we won't have to create completely new software but there will be the possibility to enter both markets, just adapting the product they require.

As it is easy to understand, as stated above, it means that basically the costs incur in the major part in one of the markets while we will have the possibility to exploit both market to cover the costs we incurred on.

15. 4 Ps

Product

The products we provide are software that allows businesses to store data related to their business.

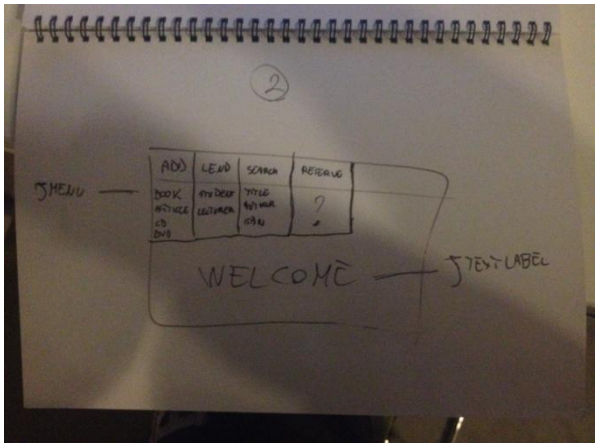
The first version developed by us, was custom made software for a library where librarians needed to store information about books, articles (CDs and DVDs). The functions required for the system were that the librarians should have been able to add new items, write the information of the borrower with the due date (the software is able to calculate the return date differently depending on the item borrowed and who was borrowing it), reserve the item in case was already lent when asked to be borrowed, and search for an item in the system, looking for the title (author, ISBN or magazine). The software has been created with the storage of data in files, without using any databases since the limited dimension of the library.

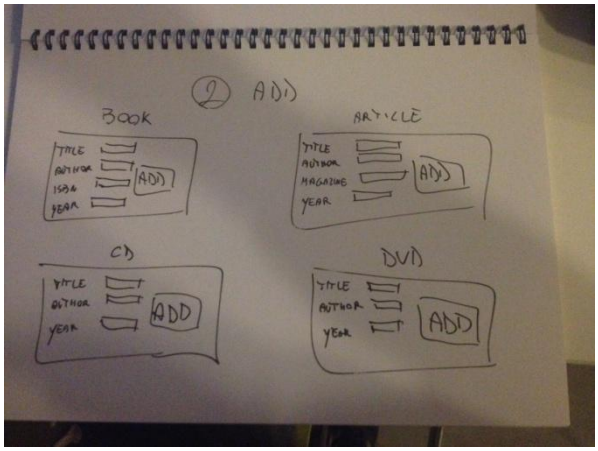
Every customers usually have different needs and it's for this reason that instead of creating products that might be used by everyone, making them adapt their needs to the software we

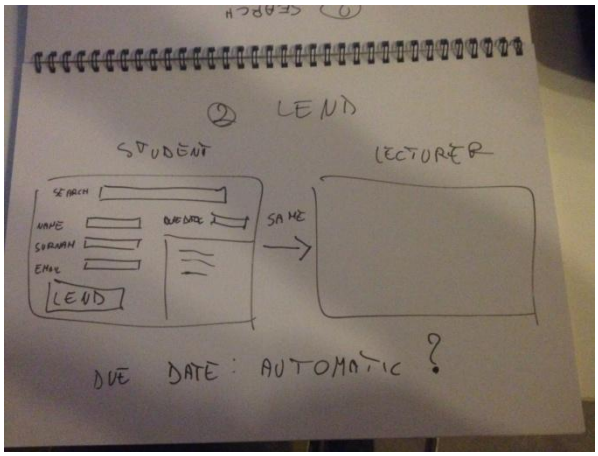
produce, we decided to make only custom made software, in order to be able to meet exactly all the requirements of every single customer.

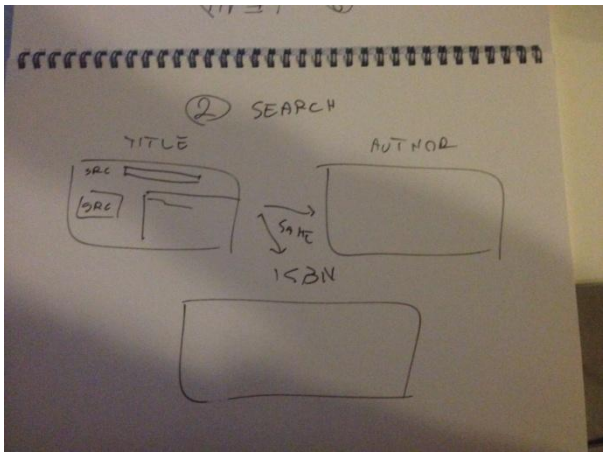
Deciding to target our products to businesses that are similar in operations and functions needed to the ones of a library, we will be able to use the general layout for the software we used for the library (see below), implementing and/or modifying the functions that the new customers require.

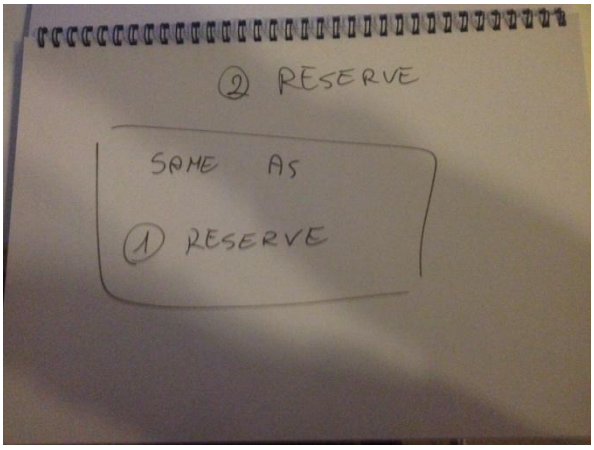
The software can be always implemented with a database in case the different customers would require that.











Taking into examination the target markets we've chosen, we should take a look on how the software would change for these businesses.

Starting from the car rental market it is easy to understand that as a first thing to change are the items that we need to give possibility to store in the software. The structure and the functions basically remain the same, there will be changes in the information to store and for example it will need fields to write the brand of the car, the model, power, color, year etc. and the items themselves like cars, bikes, trucks.

Talking about the functions, except for special requests, they remain the same, with the possibility to add items if new, to book a vehicle, to reserve in case the vehicle required is already booked and the borrower want it after it has been given back, and to search in the software everything stored in it. As previously said, it is possible to implement the software with a database storing there all the data instead of in files.

Moving to the accommodation market the items that could be stored in the software are the different rooms like, single room, double room, triple room etc. and for each of these the information to store are related to the characteristic of the room like the number of the room, the floor, the places in it and other different optional that the room might have.

Regarding the functions needed for a software that needs to deal with information about accommodations, once again, like in the car rental market are: the possibility to add items in the system, in case the hotel for example expand, to book a room for one of their customers in case they will appear with no reservation (rare case), to reserve a room for customers for a determined period of time, and to search in the system all the items inside it. Same as cars rental, the system might be implemented with a database.

Even though the structure of the functions remains generally the same in all the cases just mentioned, some function might need a different behavior; while for example the software was giving the possibility to lend books and other items for a determined period of time, depending on different factors, in the car rental and accommodations, if the customer pays, he can potentially keep the item forever. Independently on the fact that usually cars in the car rental market are kept for a short period of time, while rooms might be kept for longer.

Regarding the possibility for the software to see if the due date is past and proceed to calculate a fine, in case requested by customers, it has to be implemented and/or modified.

The last main difference between the two new markets and the library is that for the new ones it will be most certainly required that the software and the database in case, it's possible to reach also online, and be used directly also from their customers. This is due to the fact that nowadays most of the businesses are present also online and give the possibility to do operations without being physically at the shop; booking operations are among the most common online.

Our product presents itself as efficient software, capable to fully satisfy our customers needs and at the same time with a user friendly interface, that makes the software easy to use, to understand, making also all the possible functions really simple interact with.

Price

Our main pricing objective is focusing initially on survivability, in order to penetrate the market by adapting a low pricing strategy, while steadily increasing the price with a direct proportion to quality offered.

The strategy of increased price is implemented in time, after customers get accustomed to our product, while they keep in touch with our customer relation service.

Before talking about the price, we need to analyze the value that the product offered by us has for the potential customers.

Our software allows businesses to simplify their tasks in their everyday activities, making the work easier and in particular faster. Making the work faster it means for a business to be able to save costs, and to do more tasks in the time that previously was just enough for one.

In order to create a benchmark for setting a minimum and a maximum price, first we need to consider man hours in developing the product, along with updates and maintenance of the product which stand for our costs, and we also need to consider cost savings of our customers, in relation to our service.

The hours involved in developing the actual software will not be taken in consideration because our strategy takes in consideration a long term financial stability. We will focus mostly on hours which are used to maintain, update, and sell the software.
We presume that the hours per month involved in maintaining and updating and selling the software represents X DKK. Based on this, our software might have a minimum price set around to 10%X DKK basic software, and monthly updates for 20%X.Based on these assumptions, we achieve double profit when reaching 10 customers, only by taking in consideration monthly updates.

On the other hand, we must consider what would drive the customers to pay this price, or even a higher price, or even take in consideration our product in the first place. The next question that arises is what are the cost saving of customers and how our product delivers a better service so customers save money instead of losing.

Let’s presume that we have no knowledge regarding other software that is meant to manage such systems as in car rental, or accommodation. The only remaining substitutes would be manually writing in notebooks all the functions implemented in the software.
If writing manually on hard paper, or on an excel sheet would consist on Y monthly hours, we wish that customers will achieve an efficiency of 150%Y with our product, which means that their efficiency would increase with 50%. Cost savings in relation to man hours will increase to 50%, because output of work done with our software is increased with 50%.

If customers wish a higher efficiency, as in a faster platform for the system, or more features as in customizable interface and efficiency related functions they need to pay monthly more for these updates and functions, as in a “Standard Plus” System.

A “Standard Plus” version of the software, could have functions such as customizing the interface of the software within fonts, colors or placement of different functions that users might think they

are more of a high priority within their work. This would increase the customer's value and their efficiency by lowering their stress level, and increasing their comfort at work. This could lead to an increase in efficiency up to 15-20% depending on the user, up to a total of 170%Y efficiency. Based on this, we could ask for a 25%X price for monthly updates.

Another pricing strategy could consist in a "Premium" version of the software, which consists in features such as a function that mimics the previous action of the user, acting as a mirror to what the user desires to do. But since mirroring could also create problems, this function should also give the user the ability to make small changes, as if he would create a script only by typing what he wishes to implement and store, such as type of item, or renting period. This means that the user might add 10 different items in the same amount of time he used before to add 3 items. The efficiency would reach 300%Y; so therefore, monthly updates could reach 35%X.

X- Developers Costs Y- Default customer efficiency	High Price - 10%X basic purchase and 35%X monthly updates	Medium Price- 10%X basic purchase and 25%X monthly updates	Low Price- 10%X basic purchase and 20%X monthly updates
High Quality-300%Y	Premium Version		
Medium Quality-170%Y		Standard Plus	
Low Quality-150%Y			Standard Version

If we presume that we will reach only 5 different customers, we will have a monthly income of 100%X, added with a 50%X from basic purchase, this only from Standard Version. With 100%X monthly costs, we still have a 50% profit.

Standard Plus will give us 125% monthly income and a 50% onetime income at purchase. If again we subtract 100%X monthly, we reach a 25% monthly profit, and a 50% onetime profit.

Obviously, Premium Version will give us the most profit, with an outflow of 100%X and inflow of 175%X monthly plus 50% onetime inflow. Profit should be 75%X monthly and 50%X onetime inflow.

Since there is quite a big difference in efficiency from Standard Plus to Premium, and not as much in proportion of Quality and Pricing, from Standard to Standard Plus, users will most likely opt for a premium version of the software.

As mentioned when talking about the product, the software we develop is unique for each customer, shaping it accordingly to their requests. In this way they are sure to have a product that does exactly what they want, how they want it.

Said that, the market doesn't have barriers or limits about the price; you can find software in the market covering every range of products, from the free ones with few features that allow only the simplest operations, to the most expensive ones with all the possible features and functions.

The big difference of price it's possible to find in the market is due to the fact that depending on the different needs of the business, they might need something more or less complex. Depending also on the dimension of the business, it changes the sensibility on the price, with small businesses, our target, more sensible to price than the big ones.

For the reason mentioned above and because we need to be competitive to enter a new market, the price of our software will be low, affordable to almost every business. As already said, we aim to small businesses, either already in the market, or new entrants; for this reason we need to be appealing to them concerning quality and in particular price.

Our competitive advantage compared to the rivalry is that with us, our customers can have the software they required at the lowest possible price; they won't pay for anything additional unless they want to, making them completely free to decide what to pay for, depending on their needs. The software will be developed with the basic features and functions asked and some additional that they will have the possibility to purchase for additional costs.

This solution is preferred on making the software more expensive with included in the price additional features and functions that our customers might not need and maybe never use.

Apart from the price just discussed, we might offer discounts in particular occasions such us, welcome bonus or loyalty bonus, where we can provide some additional features included in the normal price, maybe for new customers or some features at half price in case of loyalty bonus.

Because of software of every price in the market, it's difficult to analyze a general trend of the market in order to be able to acquire useful information about the price. Every business has their own different strategy depending on their target and philosophy and same we need to have, not looking at other competitors.

Place

Now it's time to take a look into the channel of distribution for our software. Since we are not selling a material product, the most common solution is to use the web. There are websites used to upload every kind of software where whoever needs one can go and download.

What just mentioned is going to be taken more in consideration talking about the promotion. For now we just wanted to point out that possibility as a tool to reach potential customers.

However, due to the nature of our business and our targets, the best way to reach our potential customers it's through direct contact; contacting them directly, showing from our side interest in working with them, in having them as our customers. In this way we will be able in addition to contact the businesses that are the most similar to our target groups.

When we will be established in the market there will be the possibility for us to look into some sponsorship, maybe together with business reseller, reaching potential customers also thanks to the visibility of our sponsor.

Promotion

Recalling as mentioned before, software are promoted mostly online, in appropriate websites where software developers can upload their software and interested customers can download them and use them; sometimes for free, some other times not.

These websites are usually where most of the people find the software, since they are the first result of any search engine for anybody is looking for these kinds of products. Here, there is the possibility for developers to make their own product visible to the potential customers.

Two effective methods to promote a software, allowing the interested potential customers to familiarize with the product, are either to give them the possibility to download a free demo version, with limited functionalities, where they can have a glance of the product and its potential; or, give to the customer a free, complete version that they can enjoy for a limited period of time, after which they can continue to use it only if they buy it, otherwise all the data would get lost.

Those methods allow the users to have an idea of the features, functions and all they need to know about the software, making them able to know at the end of the demo version or the limited period of time, to decide, to know if they are interested in the software or not.

A more expensive way of promotion is to place advertisements online, in determined pages of interest where people can see our product. Usually, more than being more expensive, it doesn't give the results hoped. This is due to the fact that online advertisements usually pop out while you are surfing on the net and they are mostly annoying and distracting; people tend not to like them and most of the time they get completely ignored.

Recalling what said in the section about the channel of distribution, the direct contact it's for us the main way to promote our software, with the possibility after being established in the market to look and cooperate with possible sponsors.

Contacting directly potential customers, the best time for our business to promote our product it is when a new business enters the market, before the competitors, showing them our interest since the beginning, and our availability for them to help them in the best way we can.

By seeking science fares, and participating actively in such networks, we develop also our business network and our perceived image on the market, and therefore our influence should increase as players on the market, by associating ourselves with peers that are in similar domains.

As a conclusion, we will start with a low pricing strategy, with low quality, yet enough efficiency to attract the customer. Our purpose is to push the user into developing loyalty and trust towards our product, so we create a long term relation with our clientele. This will be done with free customer service regarding technical issues, loyalty points, social events where we encourage our customers to gather and discuss different functionalities which they prefer, and high quality service. High quality service is our agenda, which is why the difference from Standard Plus to Premium Version is quite high, in comparison to Standard to Standard Version. We will distribute our software mostly directly to business, but we offer also the possibility to buy online our software.

16. Project Planning

16.1. Purpose

The purpose of this project is to develop, design and implement a lending system for FairyTale Library, and afterwards develop a strategy to sell our product.

Objectives

- **Understand which are the requirements from an technical point of view**
- **Create Use case diagram, use case descriptions, and case diagrams**
- **Create activity diagrams for each case**
- **Create the class diagram**
- **Implement and develop the java program based on class diagram**
- **Design a Graphical User Interface**
- **Connect Graphical User Interface with Java code**
- **Create User Manual**
- **Define which industries are appealing for our software**
- **Segment the preferred industries**
- **Elaborate the strategy by working with Porter's 5 Forces**
- **Create an marketing mix, which consists of 4P's**
- **Develop a marketing Strategy plan based on our findings**
- **Sum everything up in a process and a project report**

Work Sections

a) Requirements, Diagrams, Coding, G.U.I, User Manual
b) Industries, Segments, Porter's 5 Forces, 4 Ps, Strategy Plan
c) Project and Process Report

16.2. Group Contract

Group Members:

Names	Phone Numbers	Mail	Student Numbers
Javeed Mahmoudi	+45 26 16 15 09	139487@via.dk	139487
Cortel Alin George	+45 91 86 72 32	164625@via.dk	164625
Raffaele Brivio	+45 91 98 83 12	185175@via.dk	185175

Time:

- Meet to the deadlines
- Do not be late for deals
- Make an effort to use time effectively during the project work

Goals:

- Make sure to solve eventual issues in a fair way – democratic voting etc.
- Be sure to make an end result that everyone can be okay with.
- Follow the Schedule

Schedule:

- Weekly contact until the project weeks
- Daily contact during the project weeks

Rules:

- Set up meeting dates and things that need to be done until that date
- Set up a calendar in which it becomes clear to everyone, what has been done
- and what needs to be done
- Discuss the group progress
- Keep everyone updated on the individual work
- Stick to the arrangements and deals.

1. **Daily log of group activity**

Date of Meeting	Purpose of Meeting	Actions developed	Objectives Achieved
1.10.2014	Marketing Strategies	Raffaele and Alin have had a meeting. Discussion was regarding sales strategies and marketing penetration. Talked about advantages and disadvantages of having an adapted system in different branches.	We got a better grasped of how to develop the marketing plan
5.11.2014	Check Project Description	Raffaele and Alin have had their first meeting with Anders regarding the first draft of Project Description, which included the background description, specification of purpose and problem formulation. We understood that the background can be more ample, and include more points regarding the whole view of what is the project about. We received more points in relation to how the software must adapt to a different segment of a market, such as a car lending software, or renting rooms or such, from a technical and a marketing approach. The overall structure of the problem formulation and specification of purpose had to be changed.	We changed the background, modified some problem formulated questions, and grasped a better understanding of our specification of purpose.
9.11.2014	Remaking Background	Meeting with Raffaele and Alin at school at 11.We have remade background description, specification of purpose and problem formulation with more understanding from Anders and Allan.	Refurbished background description, specification of purpose and problem formulation
4.11.2014	Marketing related issues	Raffaele and Alin had another meeting with Anders regarding 2 project reports that had to be combined. Topics such as industries, selected segments of specific industries, chosen segments and the relation of target group per segment. We have talked about the analysis of Porter's 5 Forces, and that after analysis and evaluation, SWOT analysis should be regarded as a conclusion on how to adapt our strategy. Another point was in relation to how can the software framework be adapted to which specific segments we will chose, and what changes needs to be implemented, in relation to either new features, promotions and such. It has been taken in consideration the product, place, promotion, and place and how is that related to our marketing strategy. At this meeting we have discussed regarding Javeed's problem, and his integration in the group!	Got a better grasp of how the marketing plan should be segmented and we discussed regarding the possibility of accepting Javeed in the group.
6.11.2014	Javeed Talk, and Requirements	We have accepted Javeed in the group, and we have spitted the work, within requirements.	Advancing in Requirements, and Javeed accepted in Group
8.11.2014	Project description	We have finished creating the project description by working together on choice of models and methods. Javeed has done the delimitation, Alin has done the choice of models and methods, and Raffaele has finished with deadlines, schedule and reference.	Project Description done
9.11.2014	Check up on what we have	Our meeting was regarding few I.T questions, and adjustments related to background description, purpose, problem	Remade I.T. and business questions that had needed

	and where we are heading	formulation and delimitation. Besides these, a few questions related to marketing were modified.	some changes.
1.11.2014	Case Diagrams, Case Descriptions and Requirements; Chosen industries, and segmentation	We have divided the use Case Diagrams, use case descriptions among us. Javeed has answered who are users and non functionality question, and Cortel has done functionalities and what are the requirements.	First version of use diagrams, use case descriptions, which are the users, requirements, and functionalities set up. Chosen industries, and segmentation has been worked through
4.11.2014	Recheck of what we have; Porter's 5 Forces	Talked about remaking the requirements as in bullet point's structure. Talked regarding the case diagrams that each should have its case. We have discussed that we can delegate the marketing part, since coding can be done separately also.	Started working points on choice of models and methods questions Porter's 5 Forces has been done
6.11.2014	Case descriptions recheck?	We had a meeting with Allan regarding requirements, and case descriptions which we refurbished.	Case descriptions, use cases done after being refurbished.
7.11.2014	Activity diagrams	We started checking overall diagrams, refine what we have, and finished return case diagram. After this we started to work on Class diagram.	Activity diagrams done, besides return and search activity diagram.
1.12.2014	Activity diagrams and class diagram	We met again at Javeed's place and we started again a brainstorm regarding how the class diagram should look. We made a first version of a class diagram in astah.	First version of Class Diagram Done, All activity diagrams are fully done.
3.12.2014	What is our target group? And which are the 4 P's? Developing Class Diagram	We met again and discussed the target group and the relation of the marketing mix with the target group. Besides business issues, we have worked with the class diagram	Target group done 4p's done Class diagram still in developing mode
4.12.2014	Class Diagram And Implementing Code, In parallel with GUI	We have had another meeting with Allan at 14:30 regarding our class diagram, and some functionality, such as "is before" method in my date and regarding inventory array.	Class diagram finished,Code 40%done, G.U.I. 50% done
6.12.2014	Coding G.U.I and Class Code	Working in parallel with coding the G.U.I. and the software coding.	75% both G.U.I. and Class coding
7.12.2014	Working on class code, and G.U.I.	Besides meeting regarding the class code and G.U.I., we have talked about taking in consideration Java doc, user Manual, and how to connect G.U.I. with functionalities with the adapter, and store objects in files.	G.U.I done Class code done, except functionalities such as adapter and file storage.

8.12.2014	Working on Code, Java Doc, User Guide	The meeting was meant to work on developing the code, and working on distributing the Java doc to Javeed, and User Guide to Cortel.	Java doc started, Code still developing, User guide manual Started
9.12.2014	Working with Code, and Java doc	The coding still has few breaches, in relation to adapter, storage in files and implementing the functions in relation to G.U.I.	Java doc is still in process User Manual is done
0.12.2014	Marketing Strategy And Coding	We have had a meeting with Anders related to what we are missing in our project. We realized we need to focus on making a conclusion, and that SWOT might be a good idea to sum up what we currently have. We should also be concerned with price strategy as in the efficiency of the program through an economical perspective of the buyer.	Started Developing a price strategy Java doc still developing Coding still developing
1.12.2014	Finishing Coding Marketing price strategy Java Doc	This is the meeting in which everything should be done, and ready to be implemented in the project report. We have finished the Coding, Marketing Price Strategy, and Java doc. Now we are summing everything we have and wrap it up in a project report.	Code Finished Business related issues Finished Java doc Finished Project report is being developed
2.12.2014	Project Report hand in	We have put everything inside, along the Group Contract.	Project report done

17. Personal Considerations

17.1. Cortel Alin George

My reflection towards this project consists in both positive and negative aspects. As how everyone is used to his own style of work, of course I had my expectation on how the work should be developed. A positive aspect of this project is that I learned, no matter what you are doing, the only you can truly do is only when you work together with the group.
I have been bothered 2 times that some actions have been furthered without group's approval, and I felt as if there was no group , but just a work load that had to be done no matter what, and at some point I felt disappointed and I let my expectations down regarding this project at some point.
But as how any cycle in life has bumps and downs, I also enjoyed the group when we acted as a group and worked together with brainstorming and agreeing together on defined terms

17.2. Javeed Mahmoudi

I came into the group a little late, they were done with some parts of the project description but still there was some important parts left to work on. From the first day I was so delighted to work with these guys. We were meeting almost every day to make the project description finish and start work on the project. The whole project period has going well, because we followed the schedule we made for our project and respected the

deals we made for us self. When we divided some tasks between us, then it should be done to the time we were giving by and I think we respected the deadlines. The best experience from this project was that we meet every day in the whole project period and we had a good teamwork.

17.3. Raffaele Brivio

First time for me working on a project where the major focus is on IT, programming in particular, instead of pure business. It has been a little strange at the beginning because of the difference of the subject, difference that came with us through the entire project. Different approach, different way of thinking, less hypothetical and more focused on a tangible result, where consequently there is less room for mistakes.
It has been challenging but interesting; maybe we met some problems at the beginning, in particular about comprehend each other, where everyone thought was right about something, but that was mainly due to the different background we had in a subject relatively new for all of us.
Even though I think that at the end of the project, being something new for all of us, we worked in a quite nice way, I think we could have done much more; first of all without waiting the last moment to do the most of our project and instead starting before and have time to figure everything out properly.
I know I put a lot of efforts in this project, because I liked it a lot and in general I like programming, but I know that I could have done much more and for sure I will next time, learning from my mistakes, starting before and having more knowledge of the field compared to this time that I was a newbie in this field.
Overall I really enjoyed it and I think he left a good experience and new skills learnt that will for sure be really useful in the future.