

**Vilnius University
Faculty of Mathematics and
Informatics
Prof. Agne Brilingaite**

Modern Library(501) -Report-

**Alexandru Mitu, Antonia Carasel
Internatinal Group**

1. Abstract

The project simulates the daily activities that take place in a modern library. There are three categories that can be found in the program: Borrowing a book, Renting a room and Participating at an event. Based on this classification, the frames and the code were divided in sections accordingly.

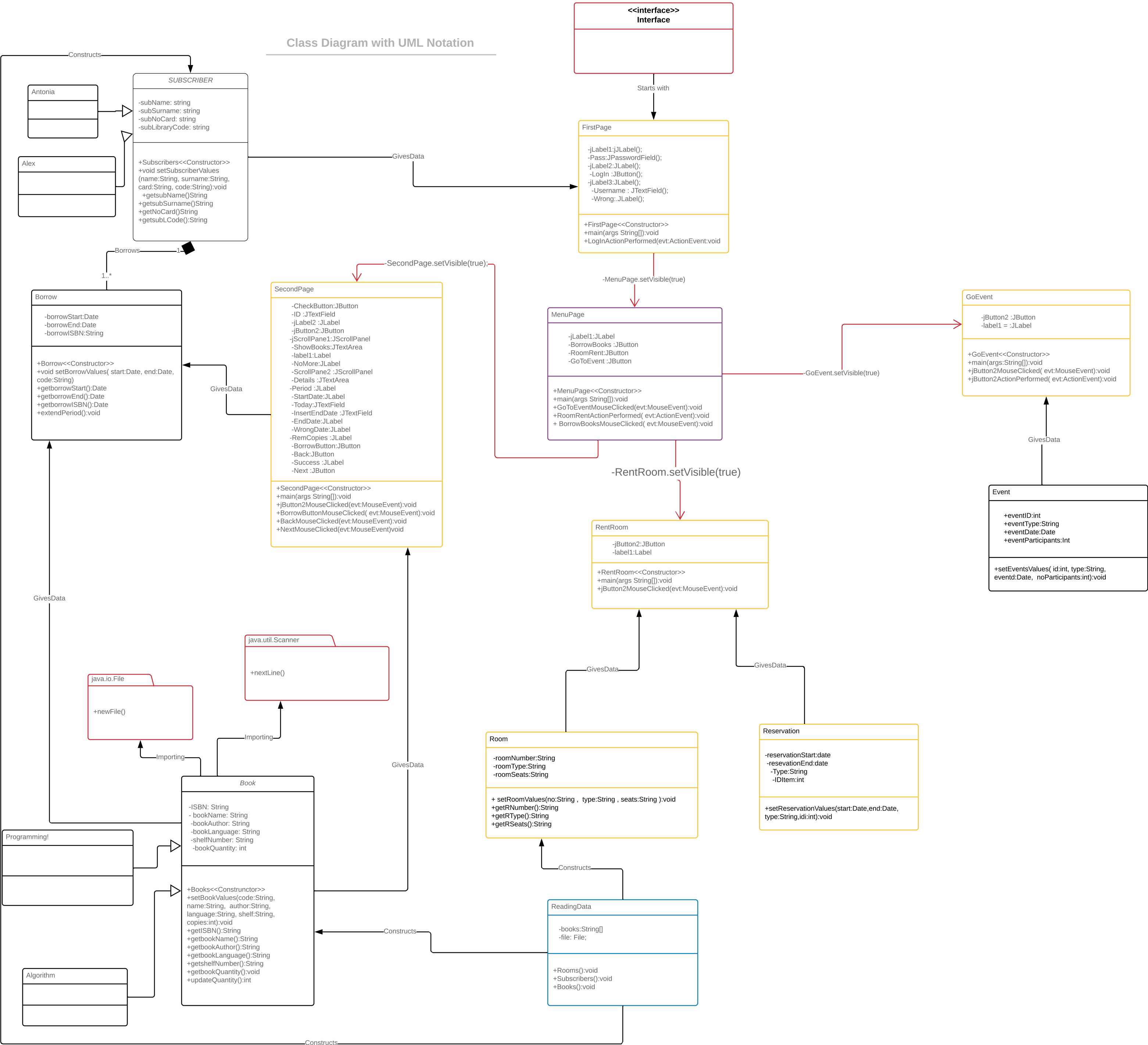
2. Program Presentation

This project has as its main functionality the process that needs to be followed in order to borrow a book from the library.

It was developed using **NetBeans**. This allowed us to work better with the graphical interface as we were able to develop a new frame for each of the functionalities. Furthermore, it contained the drag-and-drop feature for text fields, button etc which made it easier to work with these commands.

3. Program Structure

Class Diagram with UML Notation



4. Program Parts

The program can be divided in 3 parts, based on the activity in the library they are referring to.

For Book Borrowing:

Class Book represents the object itself. It contains all the necessary details of the book. The variables used are ISBN, name, author, quantity and place in the library. The class contains “set()” method that adds initial values into all the fields and several “get()” methods that return the values from the variables so that they can be used in other classes of the project.

Class Borrow was implemented to retrieve the ISBN of the book and the borrowing period.

For Renting a Room:

Class Room works as well as an object that contains all the available workplaces in the library, stating the number of the room, its type and capacity. The same get() and set() methods are implemented.

Class Reserve is

For Events:

Class Event is the third object that we used in our work. It contains all the details about upcoming events that are held at this facility. The methods previously mentioned for the other 2 objects are also written here, to set and get all the data.

Class Reservation has the code of the event and the number of people that participate.

Another important class that we used is **ReadingData**. This class implements working with the files containing data for 3 other classes: Book, Room, Event. First of all, we declare an array of objects of the type Book. Method **ReadBooks()** takes data from the file as a string by using **Scanner()** method. After that it divides the string into an array(method **split()** whenever it finds symbol “,”) and copies each position from the array into the corresponding variable from the object. The while statement makes sure that all the Books are inserted into the objects. The same process is followed for the other two objects(Room and Event).

Last but not least, the project uses a total of **5 frames** that were declared separately as classes. **FirstPage** is responsible for the LogIn of the user, the application can not be used without the right set of credentials. After that, the **BorrowMenu** frame is displayed that requests the user to choose between the 4 types of activities offered by the library. The **SeconPage** is

The **most important part** of the project can be found in the **SecondPage Frame**. Its implementation is the key towards making the project functional as it allows the user to

borrow a book and updates the information from the objects once the user has finished the reservation process.

At first, the user can only see 4 commands

- **Search** button
- text box **ShowBooks** where the result of this search is displayed
- field **ID** where the user inserts this detail for one book
- **CheckButton** that shows the user id the book is available or not

The rest of the command are hidden using **HideCommands()** method. They are not yet necessary, so there is no point un displaying them.

We didn't manage to implement the Field yet, however its purpose was to compare the data inserted by the user with every field of the objects Book and select only those that contain those letters. After pressing the Search Button, all the books that match the search are displayed in the Text Box. The user inserts the ID in the field on the right part of the page and presses Check.

If the book is not available, a message stating this fact will be displayed inside **Label NoMore**. Otherwise, the next set of commands is shown by invoking method **ShowCommands()**.

- text box **Details** with all the information about the selected book
- fields **Today and EndDate** with the dates for the borrowing period
- **Borrow** button

The Start Date is automatically filled with the current date, while the End Date has to be written by the user. If the period exceeds 30 days, the book can not be borrowed. Otherwise, after pressing "Borrow" button, the program will show the message that your process was successful. The number of remaining copies of the certain book is updated by invoking method **updateQuantity()** which receives the new number as a parameter and sets the value to the bookQuantity variable from the object.

An additional button is shown that allows the user to borrow another book. In this case, all the commands that were previously shown will be hidden (method **HideCommands()**) and a new process will begin.

The Borrowing functionality was in fact the part of the project that required the most time.

The declaration of the classes, as well as data retrieval from files were developed by Antonia. Alexandru focused on implementing the frames and the LogIn option. The SecondPage, which, as mentioned before was much more complex than the others, was developed by both of us.

5. Team and Project Management

Overall, we believe that this project made us realize what developing a complex application would truly mean. It was an exciting and practical way of testing the

knowledge that we already have, while determining us to dig deeper and make some research on our own.

The team was well balanced and we managed to divide the work evenly. Probably the best part of this project was the fact that we tried to work together on each class, frame etc as to make sure that any of us understands each part of the project.

Looking back, we believe that we needed a little bit more time to finish the project. At first, we tried to use complicated methods or concepts, that were too difficult to implement and required more time and practice. After we managed to find the simplest way, our work was effective and the project started to take shape.

6. Work Distribution

As far as the tasks are concerned, we can not say that they were thoroughly planned. For the first programming part, one of us focused on declaring the classes, while the other was preparing the necessary data to test them. The second part we figured out while coding, based on the things that we needed to move forward. We realized that in order to check if the program is functional, we would need to insert some data and see the results. This led us to believe that the first task should be implementing the methods that read the information from the files. After that we focused on the graphical interface and added the first two pages(Login and Borrow Menu). The SecondPage came next, which took the most time to implement and was somehow the core of the project. After that we added the other two functionalities.

The time we did allocated for the project was well managed and we worked in an extremely efficient manner. However, we didn't realize just how many days we would need to finish the project, which was the biggest minus of our team work. We believe that better results could be achieved if we would have been able to eliminate this setback.

All in all, we consider this experience one of the best we could have had. We both believe that most important part in the learning process is not the amount of information you have, but your ability to actually put it into practice. This is exactly what we did during the past weeks and we are certainly more capable of applying everything that we learned during the semester.