VIA UNIVERSITY COLLEGE

The Best Library

a Web Engineering Processes project

Baptiste Finck, student number 310572
Tommaso Brumani, student number 310561
Lucie Naffien, student number 310573

Supervisor : Allan Henriksen

Table of content :

List of figures and tables :

I. Description

This project has been made as part of the Web Engineering Processes course from the ICT programme for international students of Via University College, in Denmark.

Because of the current health situation, it is very difficult to go to the library to borrow books. One solution could be to make reservations for them online, and then pick them up when they are available, so it would avoid a lot of direct contact and help to improve the current health situation while still letting people borrow books they need for their studies or to have a nice reading time at home. That is the reason why we decided to create a website for libraries.

II. Process

A. Use case

First, we made a use case diagram and description, so we could later define all the requirements our website will need to have. We used Google Drive's tools diagrams.net and GoogleSheet to make them (links can be found in the appendices).

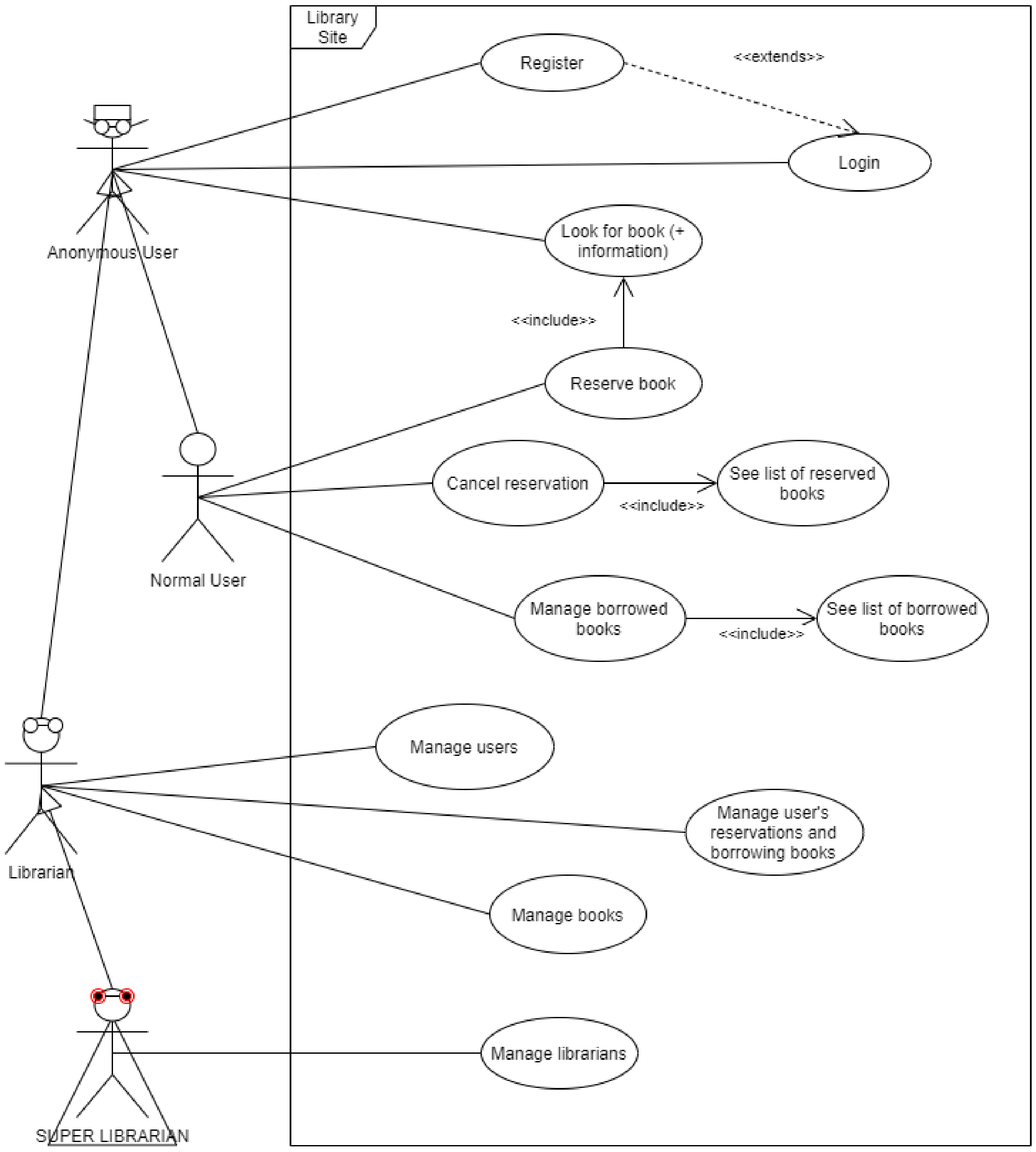
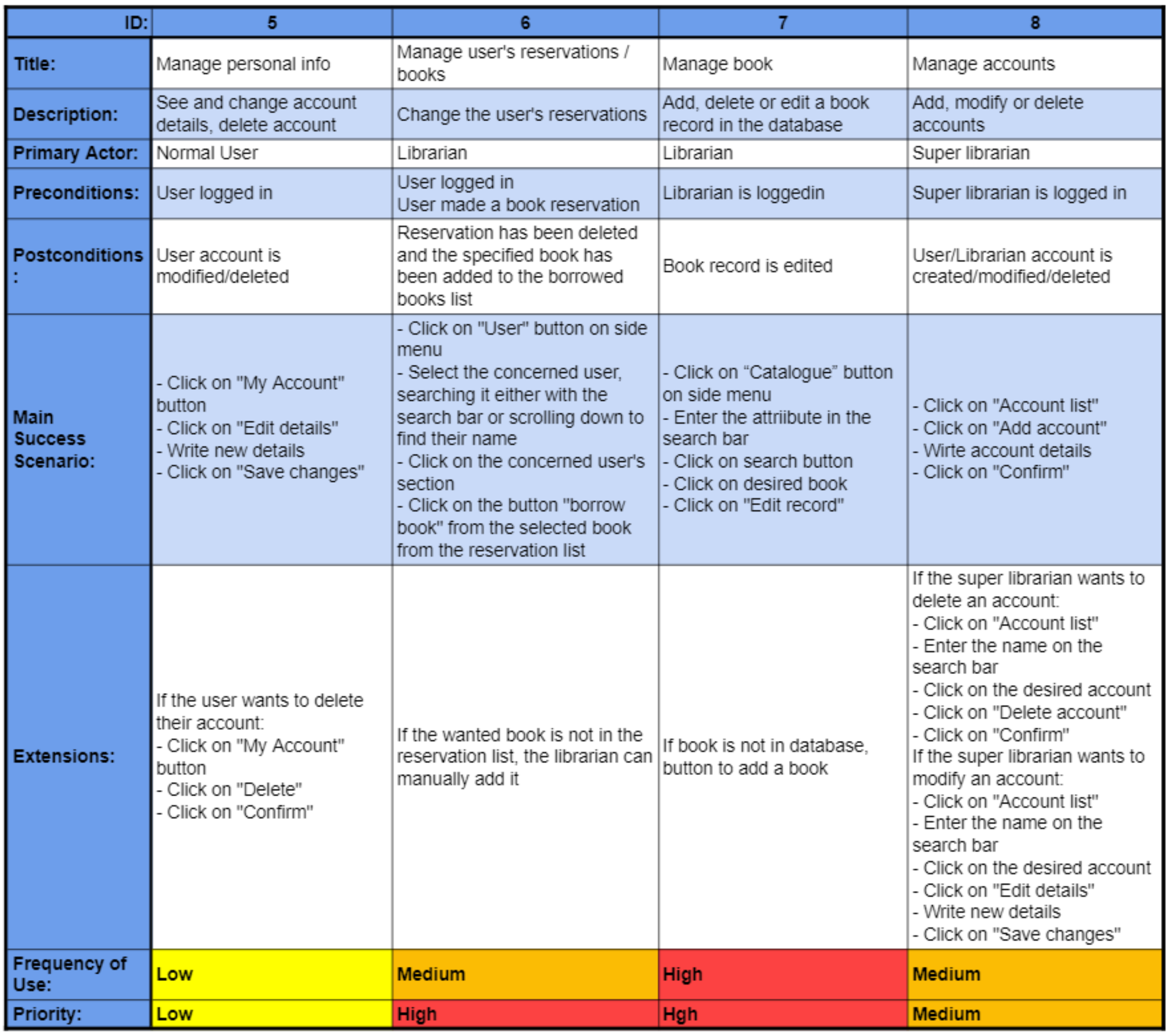


Figure 1 : Use case diagram

ID:	1	2	3	4
Title:	Get book info	Reserve a book	Cancel reservation	Manage borrowed books
Description:	Get information about a specific book	Make a reservation for a book	Cancel book reservation	See user's borrowed books, the due date and extends it if needed
Primary Actor:	Anonymous user	Normal user	Normal User	Normal user
Preconditions:	User on main page of the website	User logged in	User loggedin User has reservation	User logged in
Postconditions:	Info displayed for desired book	Book added to user reservations (for 2 days)	User reservation deleted	Book's due date is extended
Main Success Scenario:	- Click on “Catalogue” button on side menu - Enter the attribute in the search bar - Click on search button - Profit	- Click on “Catalogue” button on side menu - Enter the attriibute in the search bar - Click on search button - Click on desired book - Click on "Add to reservations"	- Click on "Manage my reservation" button - Click on a reservation - Click on delete	- Click on "Borrowed books" button - Choose the book you want to extend the due date - On the right side of the book's informatios, click on the button "Extend due date" - Select the time you want to extend it amough the suggested ones
Extensions:	- If book doesn't exist in database display: We don't know this book	If book is not available: user is added to reservation queue for that book, is alerted when their book becomes available	If user don't have reservation display no reservation	If the user didn't borrowed any books : "You don't have borrowed books currently"
Frequency of Use:	High	Medium	Low	Medium
Priority:	High	Medium	Low	High

Table 1 : Use Case description part 1



ID:	5	6	7	8
Title:	Manage personal info	Manage user's reservations / books	Manage book	Manage accounts
Description:	See and change account details, delete account	Change the user's reservations	Add, delete or edit a book record in the database	Add, modify or delete accounts
Primary Actor:	Normal User	Librarian	Librarian	Super librarian
Preconditions:	User logged in	User logged in User made a book reservation	Librarian is loggedin	Super librarian is logged in
Postconditions :	User account is modified/deleted	Reservation has been deleted and the specified book has been added to the borrowed books list	Book record is edited	User/Librarian account is created/modified/deleted
Main Success Scenario:	- Click on "My Account" button - Click on "Edit details" - Write new details - Click on "Save changes"	- Click on "User" button on side menu - Select the concerned user, searching it either with the search bar or scrolling down to find their name - Click on the concerned user's section - Click on the button "borrow book" from the selected book from the reservation list	- Click on “Catalogue” button on side menu - Enter the attriibute in the search bar - Click on search button - Click on desired book - Click on "Edit record"	- Click on "Account list" - Click on "Add account" - Wirte account details - Click on "Confirm"
Extensions:	If the user wants to delete their account: - Click on "My Account" button - Click on "Delete" - Click on "Confirm"	If the wanted book is not in the reservation list, the librarian can manually add it	If book is not in database, button to add a book	If the super librarian wants to delete an account: - Click on "Account list" - Enter the name on the search bar - Click on the desired account - Click on "Delete account" - Click on "Confirm" If the super librarian wants to modify an account: - Click on "Account list" - Enter the name on the search bar - Click on the desired account - Click on "Edit details" - Write new details - Click on "Save changes"
Frequency of Use:	**Low**	**Medium**	**High**	**Medium**
Priority:	**Low**	**High**	**Hgh**	**Medium**

Table 2 : Use Case description part 2

B. User stories

In order to have a clearer idea of the requirements, we made the below user stories. As shown on the use case diagram, our website needs to handle 4 kinds of users : anonymous user, user, librarian and admin.

Anonymous user :

- An anonymous user wants to see the library's catalogue. They can go to the website and see the whole catalogue.
- An anonymous user wants to see more details about.
- An anonymous user wants to register at the library. They go on the library's website and register their mail address, last name, first name, age, address, phone number (optional).
- An anonymous user wants to be able to log in.

User :

A user can do everything an anonymous user can do.

- A user wants to borrow a book, they go to the library and the librarian adds it to their account.
- A user wants to see a book's availability. They go to the website, can search for the specified book and see if it is currently available at the library or not.
- A user wants to reserve a book. They go to the website, can search for the specified book, connect to their account and reserve it. Then they can see if they can go to the library to take it with them, and if it is not available, the supposed date it will be available.
- A user wants to see which book he has borrowed. They connect to their account and can go to the list of books they borrowed.
- A user wants to know when they will have to return a book. They connect to their account, go to the list of books they borrowed and can see when is the due date. If needed, they can search for a specified book.
- A user wants to extend the borrow date of a book. They go to the borrowed books list of their account and can extend the date of a specified book.
- A user wants to cancel a reservation. They can go to the reservation list of their account and cancel it.
- A user wants to delete their account. He goes to the website, connects to their account and can delete it in the account's parameters. It will not be possible if they still have borrowed books. If they have reserved books, they would be canceled.

Librarian :

- A librarian wants to be able to add a book to a user's account
- A librarian wants to see information about a user. They can connect to their account and select a user from the user list, or search for a specified one.
- A librarian wants to block the ability of a user to borrow a book, because for example they didn't bring back a book in time, or extend the due date too many times. They connect to their account, select a user from the user list and block their borrow ability.

Admin :

- Create librarians

C. Requirements

With the help of the use case diagram and description and the user stories, we were able to define this project's requirements. In order to do so, we applied the MoSCoW method. You can find below what we came up with :

1. Must have :
    - Possibility to log in and log out
    - Books' catalogue
    - User's management by librarians and admins and librarian's management by admins
    - Possibility for an user to reserve and borrow book, see when it should be available and they reserved and borrowed books' list
2. Should have :
    - Research and filter books, users and librarians
    - Book's details
    - Cancel a reservation
    - Extend the duration of a book's borrowing
3. Could have :
    - Advanced research
    - Account management (change password and mail)
    - A page with a tutorial about how to use the website
    - A page with library's useful informations
    - Availability for librarians and admins to prevent a normal user from reserving and borrowing books
    - Extend the due date to a borrowed book
    - Books' management by librarians and admin

4. Won't have :

- Friends, Chat, Messaging and Comments features
- Help page
- Automatic mails (reserved books available, confirm email address, password recuperation, connection from a new device...)
- Library's actualities
- Features for visually impaired people
- Mobile version

D. Design

To decide on our website's design, we used the collaborative interface tool Figma (you can find a link of it in the appendices). We decided to make it look like the overview below :

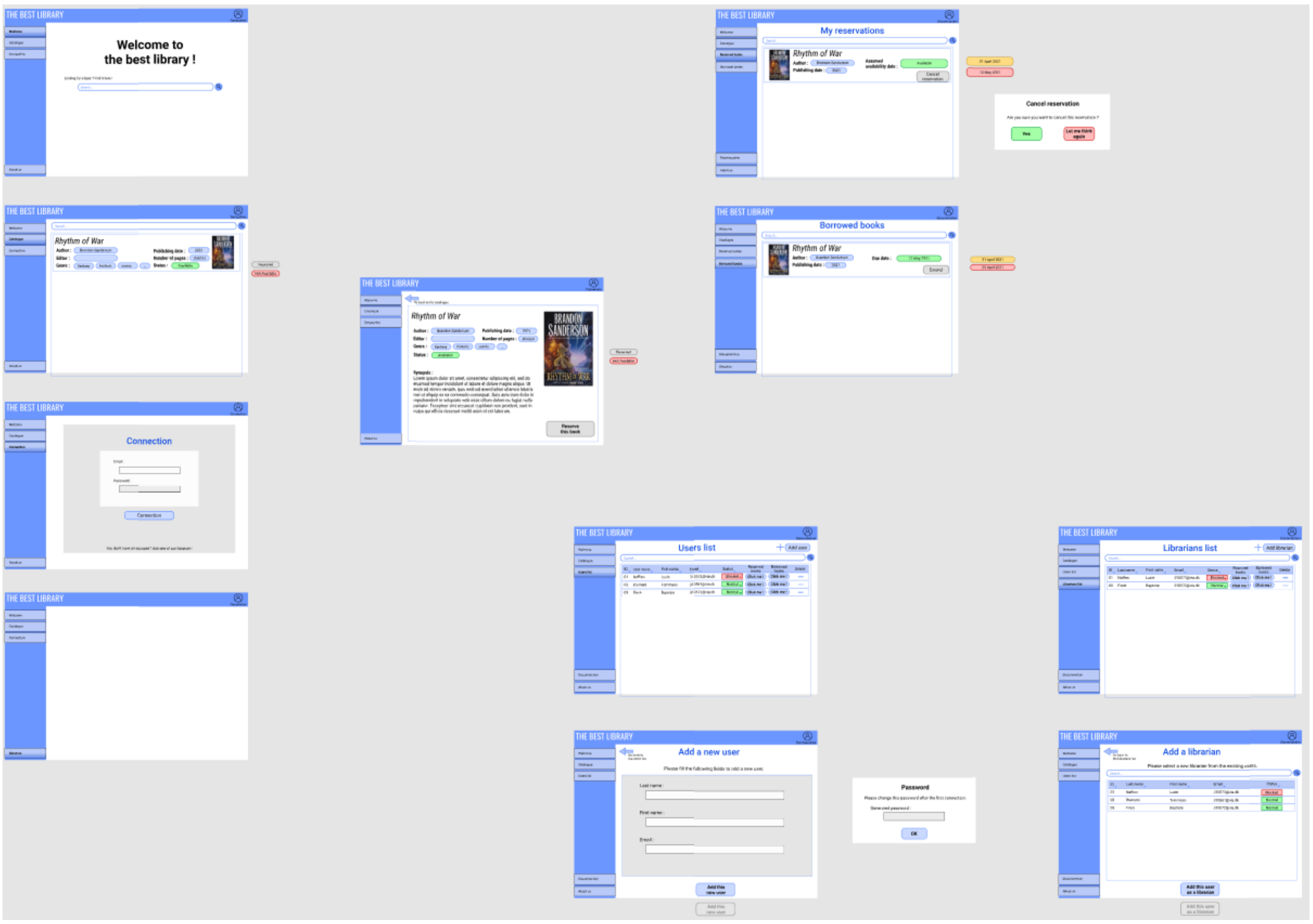


Figure 2 : Design Overview

E. ER Diagram

To be able to work, our website will need a database, so we made the below ER Diagram, with diagrams.net to define it :

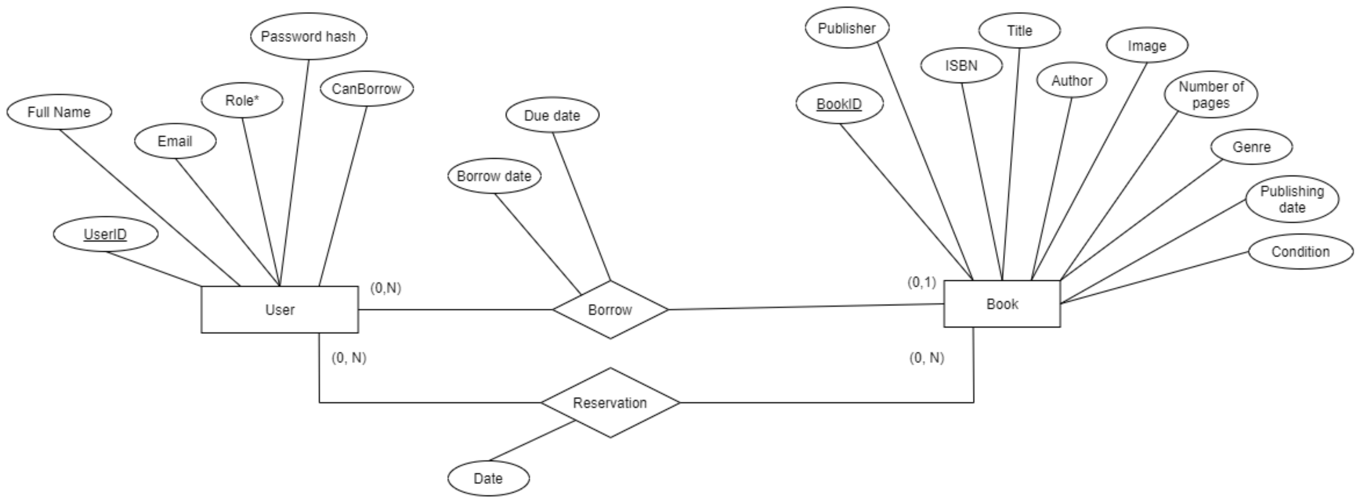


Figure 3 : ER Diagram

F. Implementation

To make our ASP.NET website we used Visual Studio Express for Web 2012, and to coordinate our work we created a shared git repository with GitHub.

Users and Roles

We decided to use ASP.NET's built-in user authentication and membership features, which take care of handling many aspects of user handling (such as passwords and some user details).

We also settled for using ASP's roles feature, which enabled us to assign specific roles to the various users, thereby providing them with different levels of access to the various pages and elements.

Users can be created by Librarians and SuperLibrarians.
Librarians can be promoted from Users by a SuperLibrarian.
SuperLibrarians can only be created from the ASP role manager, as they are an admin role.

Master page

The site is built using ASP.NET's Master Page feature, which allows to make a page that acts as a framework upon which all other pages are built.

We used this feature to make the site's header and footer constant throughout the various pages, as well as an ASP Tree View on the left side to navigate it, which displays different links depending on the login status and role of the user (thanks to the Custom Site Maps feature).

A login/logout button was also placed at the top right corner, changing its function depending on the user's login status.

CSS

In order to style the site, we decided to make use of separate CSS files by assigning CSS classes to various HTML elements. In some instances we however decided to write the CSS directly in the HTML tags for simplicity.

Login

Thanks to ASP.NET's handling of user login, we were able to keep the Login page extremely streamlined, as it contains only the ASP login element.

The page is accessible to anonymous users by using both the left-side Tree View and the top-right hyperlink.

Users can easily logout by pressing the top-right hyperlink, and if necessary will temporarily be redirected to a Logout page, and then to the home page.

Home Page

The Home (Default) page of the site contains a search bar that can be used to search a book in the library database by its title. This was achieved using the code behind C# file, by saving the search string in a Session variable, and retrieving it in the Catalogue page after redirecting the user to it.

Catalogue Page

The Catalogue page contains an ASP List View of the books in the database. It displays some information on the books, such as their title, author and others, and displays whether the book is borrowed or available to logged in users.

The list displays 5 books per page, visible through a scroll bar, and features buttons to navigate the various pages at the bottom.

The page also includes a search bar for the books' titles, which is filled and triggered automatically with the data retrieved from the Home page if necessary.

The search bar was implemented through the page's C# file, by executing different SELECT commands on the database depending on the search bar's contents.

Every item of the list also includes a button to enable users to view the book's Details page: in the C# file, pressing the button saves the book's id in a Session variable and redirects to the appropriate page.

Book Details Page

The Details page makes use of a single-element List View to display the details for a book, whose id was retrieved from the Session variable it was stored into by the catalogue page, by binding its data to an appropriate SQL Data Source.

This page displays all of the book's attributes in greater detail.

For logged in (regular) Users, the page also displays a button that allows them to make a reservation for this book, if they haven't already. This is achieved by the C# file, by adding a row to the Reservations table with the current date as a date for the borrowing.

For Librarian and SuperLibrarian users, the page displays instead an ASP Grid View of the book's reservations and the details of the users who made them.

It is possible for them to delete one of these reservations through the use of the Grid View's DELETE feature.

If the book is currently borrowed by a user, they may also signal its return causing the corresponding row on the Borrowings table to be deleted.

If the book is instead available, they may signal its borrowing for one month by a specific user, selected through an ASP Drop Down List bound to an SQL Data Source, displaying the username of all Users whose CanBorrow attribute is set to "True".

It is also possible for them to delete the book altogether.

All of these features are achieved through database manipulation in the C# file.

Borrowings and Reservations Pages

The Borrowings page displays an ASP List View of the books a user has borrowed, along with some brief details about them and the date for which the return of the book is due, whereas the Reservations page does the same for those they have reserved, along with displaying their status (borrowed or available).

In both of these cases, this is achieved by binding the list to an opportunely parameterized SQL Data Source, and they both have a search bar for the books' titles, implemented similarly to the Catalogue Page.

For (regular) Users, the pages always display their own borrowings or reservations, but Librarian and SuperLibrarian users may switch between different Users'.

These pages are accessible only to logged in Users, Librarians and SuperLibrarians.

Add Book Page

The Add Book page allows a Librarian to add a book to the database. To do this, it makes use of ASP.NET's Text Boxes, as well as AJAX Control Toolkit's Filter Extenders for Text Boxes, which make it possible to blacklist certain characters.

To select the date of publishing of the book an AJAX Calendar Extender is used, and a picture of the book's cover may be uploaded.
If any of these fields, other than the title, are missing, they will be set to the default parameters in the database. There is no security if the book's image is corrupted or very large. In the first case, it will display the browser's default image when it can't find or access to it. Upon creation a book is always flagged as available.

The creation of the book is handled by the C# file by use of a parameterized INSERT command.

This page is accessible only to logged in Librarians and SuperLibrarians.

User List page

The User List page contains an ASP Grid View of the Users in the database and many of their attributes. It allows a Librarian to modify some of the attributes and delete the users through the Grid View's DELETE and UPDATE features, and it contains buttons to reach their Borrowings and Reservations pages (through the use of Session variables in the C# file).

This page is accessible only to logged in Librarians and SuperLibrarians, and it also contains a button to reach the Add User page.

Add User page

The Add User page makes use of ASP.NET's Create User Wizard to create a new User account.

The First Name and Last Name tabs were added to fill the UserDetails table, and this is implemented in the C# file through an INSERT command. The CanBorrow attribute is set to "true" by default in the database, and can be edited from the User List page.

This operation is meant to be carried out by the library staff, and so the page is only accessible to logged in Librarians and SuperLibrarians.

Librarian List page

The Librarian List page contains an ASP Grid View of the Librarians in the database and some of their attributes. It features a button to allow a SuperLibrarian to demote them to the role of (regular) Users, which is achieved by modifying their role through the C# file.

The page also contains a button to the Create Librarian page, and is accessible only by SuperLibrarians.

Add Librarian page

The Add Librarian page contains an ASP Grid View of the Users in the database. It features a button to allow a SuperLibrarian to promote them to the role of Librarians, which is achieved by modifying their role through the C# file.

This page is accessible only by SuperLibrarians.

Custom Error page

The site contains a custom Error page that is displayed when something goes wrong with the site. From there, users can go back to the site's Home page.

Database

The database used is the default MS SQL database present on ASP.NET. It features numerous standard tables used by ASP to handle user authentication, membership and roles, as well as an additional UserDetails table for some extra user information.

The tables for Books, Borrowings and Reservations are also present, as well as a Statuses table to determine which values the Status attribute in the Books table can assume (it includes the Available and Borrowed values, as well as the Reserved and Unavailable values in case they ever became necessary).

Missing Features

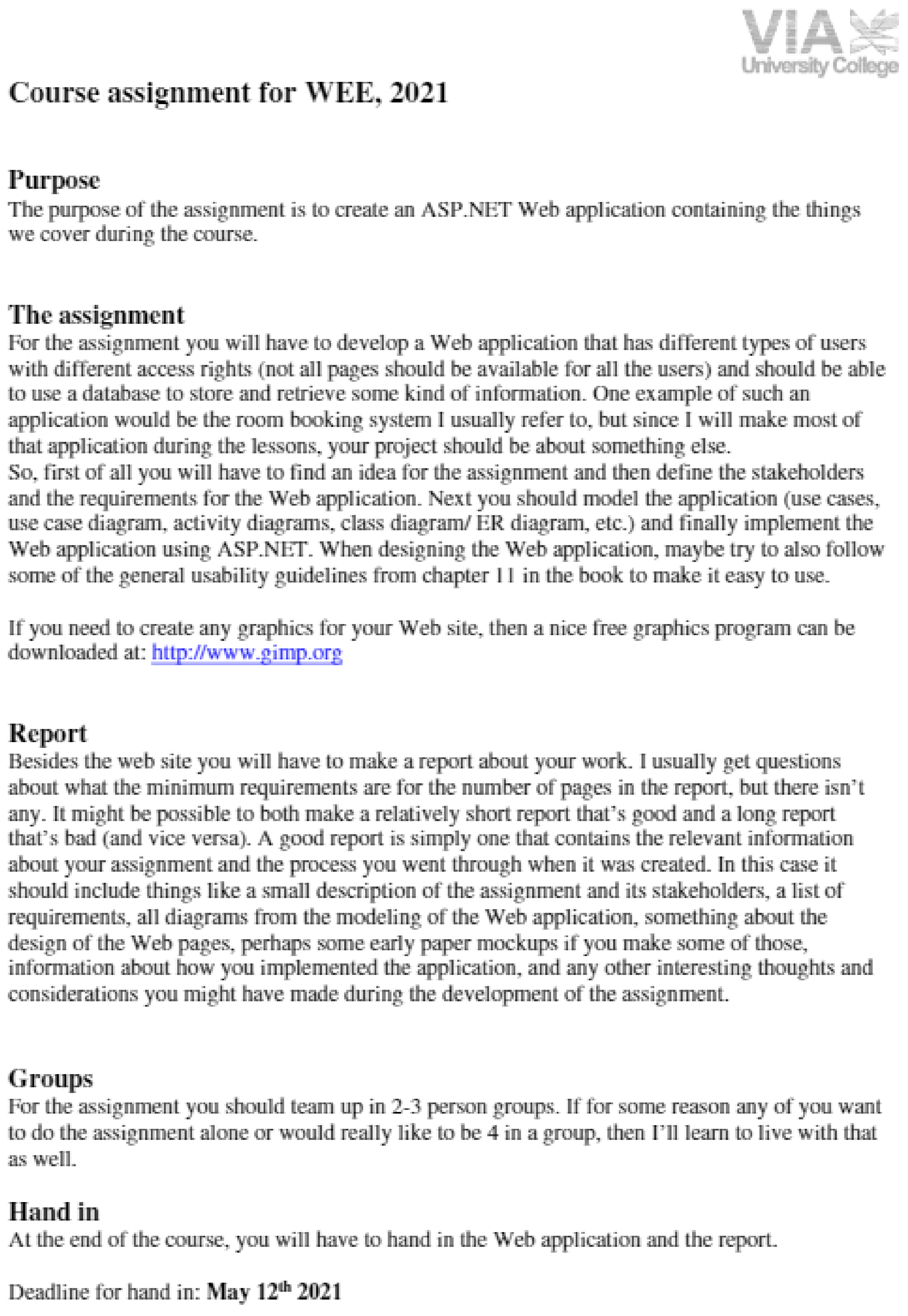
Due to time constraints, some features were cut in order to focus on more essential aspects of the site. These could be easily implemented with some extra time, but were deemed of secondary importance compared to the rest.

These features include:

- the ability to delete one's own reservations on a user's part.
- the displaying of whether a user is the next in line in the reservation of a book.
- the ability for a user to change their username, email and password.
- the ability for a librarian to choose a variable amount of time for a borrowing.
- the ability for a librarian or user to extend the length of a borrowing.
- check on syntaxes, such as Email address syntax.
- instead of adding a corrupted or very large image as a book cover in the database, using our default book image.
- less memory leak. For example, data and variables for the page number are integers. However, a page number can not be negative, and the largest book in the world has 7312 pages, so using an unsigned integer on 16 bits - or an unsigned short integer - is enough.

III. Appendices

Project description



Course assignment for WEE, 2021

Purpose
The purpose of the assignment is to create an ASP.NET Web application containing the things we cover during the course.

The assignment
For the assignment you will have to develop a Web application that has different types of users with different access rights (not all pages should be available for all the users) and should be able to use a database to store and retrieve some kind of information. One example of such an application would be the room booking system I usually refer to, but since I will make most of that application during the lessons, your project should be about something else.
So, first of all you will have to find an idea for the assignment and then define the stakeholders and the requirements for the Web application. Next you should model the application (use cases, use case diagram, activity diagrams, class diagram/ ER diagram, etc.) and finally implement the Web application using ASP.NET. When designing the Web application, maybe try to also follow some of the general usability guidelines from chapter 11 in the book to make it easy to use.

If you need to create any graphics for your Web site, then a nice free graphics program can be downloaded at: http://www.gimp.org

Report
Besides the web site you will have to make a report about your work. I usually get questions about what the minimum requirements are for the number of pages in the report, but there isn't any. It might be possible to both make a relatively short report that's good and a long report that's bad (and vice versa). A good report is simply one that contains the relevant information about your assignment and the process you went through when it was created. In this case it should include things like a small description of the assignment and its stakeholders, a list of requirements, all diagrams from the modeling of the Web application, something about the design of the Web pages, perhaps some early paper mockups if you make some of those, information about how you implemented the application, and any other interesting thoughts and considerations you might have made during the development of the assignment.

Groups
For the assignment you should team up in 2-3 person groups. If for some reason any of you want to do the assignment alone or would really like to be 4 in a group, then I'll learn to live with that as well.

Hand in
At the end of the course, you will have to hand in the Web application and the report.

Deadline for hand in: **May 12th 2021**

Ressources :

- diagrams.net :
  https://www.diagrams.net/
- Google Sheet :
  https://www.google.fr/intl/eng/sheets/about/
- Figma :
  https://www.figma.com/about/
- Visual Studio Express for Web :
  https://docs.microsoft.com/en-us/previous-versions/aspnet/dd537667(v=vs.110)
- GitHub :
  https://github.com/about