**SPRINT PLANNING MEETING**

**Sprint 1 : (1.11 – 7.11)**

**Backlog story**

*As a customer or a guest I want to be able to search for a book (by inputting the name, author, isbn, year or category, filtered by (specific) bookstore/library) so that I can find desired books*

**Tasks**

* Class diagrams : Model, Bookstore/Library, book service
* Activity diagram : search for a book
* EER diagram : database – book
* Database structure for books
* Domain table and Logical model
* Implement model
* Communication protocol : search and advanced search
* Library expose REST for searching the book
* Book service redirecting and gathering book queries
* Search engine on database server
* Database server
* Library/Bookstore connect to database
* GUI : search page (main page)
* GUI : showing results on web site

**Sprint 2: (8.11 – 14.11)**

**Backlog story**

*As a user I want to be able to access a book so that I can see details of it*

*As an administrator of institution I want to be able to manage books in my institution (add, search and delete) so that the state of books is updated in the system*

**Tasks**

* Class diagram : update
* Database diagram
* Use case description for those stories
* Find out if ISBN is good for id
* Library/Bookstore : CRUD
* Tests for CRUD in Library/Bookstore
* Book service : book details
* Database : availability
* Database : update connection
* Configure travis for C#
* Set up deploy
* GUI: book detail page
* GUI: administrator page

**Sprint 3: (15.11 – 21.11)**

**Backlog story**

*As a guest I want to be able to create an account with my name, surname, address, e-mail address and phone number so that I can be a customer*

**Tasks**

* Process report
* Class diagram : update
* Database diagram
* Check backlogs and diagrams
* Use case diagram
* Library : documentation of API
* Book service : diagram update
* Book service : registration test – controller
* Book service : registration request – POST
* Book service : registration communication with database
* Library : change to new URI
* Database server : registration request
* Database server : users table + hibernate
* Database : test request handling
* GUI: Registration page
* Website : Registration page connect form with API
* GUI: Website : navbar
* GUI: Finish tasks from last sprint (book detail page, administrator page)

**Sprint 4: (22.11 – 28.11)**

**Backlog story**

*As a customer I want to be able to buy books from the bookstore so that the administrator of the bookstore can accept my order and I can buy the book.*

*As a customer I want to be able to borrow a book from the library so that this book is temporarily assigned to me*

**Tasks**

* Process report continues
* Use case
* Class diagram : book service update
* Sequence diagram : update
* Documentation of borrowing algorithm
* Documentation of API update
* Documentation of database : update
* Book service : borrow book from the library request
* Book service : buy book from bookstore request
* Book service : book borrowing algorithm
* Book service : test of algorithm
* Hibernate : query builder
* Communication : borrow order / buy order
* Database : add borrow order / buy order
* GUI : update details page with buy and borrow buttons
* Website : advanced search

**Sprint 5: (29.11 – 5.12)**

**Backlog story**

*As a customer or an administrator I want to be able to log in to the system so that I have more functionalities*

**Tasks**

* Use case
* Update diagrams : Book service, Book store, Library, Database
* Documentation of login in
* Sequence diagram : login
* Find out how to make login
* Library : implement authentication
* Book service and Bookstore : implement login in
* Database : add password
* Services : add password
* GUI : login layout
* Website : find out how to login / session / cookies

*\*On project period sprints takes 3 days*

**Sprint 6: (6.12-8.12)**

**The main goal:**

*Finish the previous backlog story and unfinished tasks*

**Unfinished tasks**

* Website : connect both admins, connect book details list to API, registration page connect to API
* Website : storing cookies, adding cookie in request
* Website : changing layout on login
* Database documentation
* Change process report
* Document authentication
* Test library authentication
* Login encryption
* Add checking session key before request in Bookstore, Book service and Library

**Backlog story**

*As an administrator of bookstore I want to be able to confirm book orders so that the order is completed and the user gets a confirmation email*

**Tasks**

* Website : Bookstore admin layout for order
* Website : Bookstore admin orders – connect to API
* Bookstore confirm order request
* Database : handle order request + communication
* Use case
* Class diagrams update

**Sprint 7: (9.12 – 11.12)**

**Sprint 8: (12.12 – 14.12)**