**CST-341 Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | *Project Bookmark* | |
| **Date:** | *12/09/2018* | |
| **Revision:** | *Version 1.99* | |
| **Team:** | 1. Ramon Leon (Joe) | |
| 1. Jerran Fredricks | |
|  | |
|  | |
| **Weekly Team Status Summary:** | |  |  |  |  | | --- | --- | --- | --- | | **User Story** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | Design Report | Joe | 1 | 0 | | Responsive Design | Joe | 3 | 0 | | NavBar | Joe | 2 | 0 | | Images | Jerran/Joe | 5 | 0 | | Pojo | Jerran/Joe | 2 | 0 | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |
| **GIT URL:** | *https://Skie@bitbucket.org/Skie/projectbookmark.git* | |
| **Loom URL:** | *https://www.useloom.com/share/f427ebe463554d59b1dcdb1f3f36af40* | |
| **Peer Review:** | *Y* | We acknowledge that our team has reviewed this report and we agree to the approach we are all taking. |

**Planning Documentation**

**Agile Scrum Product Backlog:**

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Type | Status | Estimate Hours |
| Login/Registration | Requirement | Finished | 8 |
| Home Page | Requirement | Finished | 3 |
| Product Page | Requirement | Updated | 5 |
| Database | Requirement | Finished | 1 |
| Security | Requirement | Not Started | 2-4 |
| Book Add | Requirement | Finished | 6 |
| Book Remove | Requirement | Finished | 6 |
| Page Tracker | Requirement | Finished | 2-4 |
| Admin Functions | Requirement | Out of scope | 4-5 |
| Book Badges | Idea | Out of scope | 2-4 |

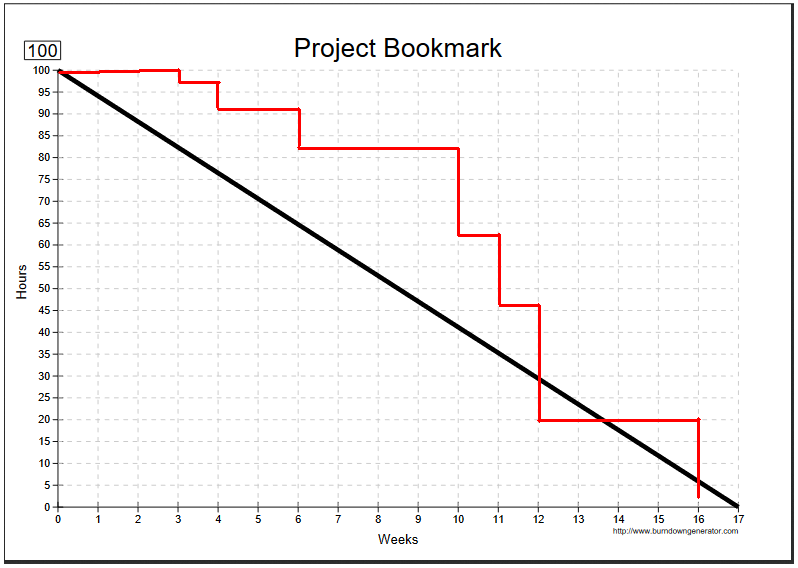
**Agile Scrum Sprint Backlog:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tasks | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 |
| Login/Registration | **0** | **0** | **1** | **2** | **0** | **0** |
| Home Page | **0** | **0** | **1** | **2** | **0** | **3** |
| Product Page | **0** | **0** | **1** | **2** | **0** | **1** |
| Database | **0** | **0** | **0** | **0** | **0** | **0** |
| Security | **0** | **0** | **0** | **0** | **0** | **0** |
| Book Add | **0** | **0** | **0** | **0** | **0** | **3** |
| Book Remove | **0** | **0** | **0** | **0** | **0** | **0** |
| Page Tracker | **0** | **0** | **0** | **0** | **0** | **0** |
| Admin Functions | **0** | **0** | **0** | **0** | **0** | **0** |
| Book Badges | **0** | **0** | **0** | **0** | **0** | **1** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tasks | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12 |
| Login/Registration | **0** | **0** | **0** | **8** | **0** | **0** |
| Home Page | **0** | **0** | **0** | **3** | **0** | **0** |
| Product Page | **0** | **0** | **0** | **5** | **12** | **8** |
| Database | **0** | **0** | **0** | **1** | **0** | **1** |
| Security | **0** | **0** | **0** | **0** | **0** | **0** |
| Book Add | **0** | **0** | **0** | **3** | **0** | **0** |
| Book Remove | **0** | **0** | **0** | **0** | **2** | **2** |
| Page Tracker | **0** | **0** | **0** | **0** | **2** | **0** |
| Admin Functions | **0** | **0** | **0** | **0** | **0** | **0** |
| Book Badges | **0** | **0** | **0** | **0** | **0** | **0** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tasks | Week 13 | Week 14 | Week 15 | Week 16 | Week 17 | Week 18 |
| Login/Registration | **0** | **0** | **0** | **0** |  |  |
| Home Page | **0** | **0** | **0** | **3** |  |  |
| Product Page | **0** | **0** | **0** | **1** |  |  |
| Database | **0** | **0** | **0** | **0** |  |  |
| Security | **0** | **0** | **0** | **0** |  |  |
| Book Add | **0** | **0** | **0** | **1** |  |  |
| Book Remove | **0** | **0** | **0** | **0** |  |  |
| Page Tracker | **0** | **0** | **0** | **3** |  |  |
| Admin Functions | **0** | **0** | **0** | **0** |  |  |
| Book Badges | **0** | **0** | **0** | **0** |  |  |

**Agile Scrum Burn Down Chart:**

**

**Agile Retrospective Results:**

|  |
| --- |
| **What Went Well** |
| Bootstrap and adding a responsive design. |
| Formatting webpages and add in a navbar. |
| Ensuring all links and functions still worked as intended. |
| Images. |
|  |

|  |  |  |
| --- | --- | --- |
| **What Did Not Go Well** | **Action Plan** | **Due Date** |
| The “Hamburger” button refused to work for the longest time. | Got help from Lewis Brown. | 12-9-2018 |
| Images to display correctly. | Accidently stumbled the correct way to render them in jsp files, however, it went unnoticed since the correct tags were used in the footer. | 12-9-2018 |
|  |  |  |
|  |  |  |

**Design Documentation**

**Install Instructions:**

Clone the repertory and then create a new Dynamic Web Project within the IDE of your choice. Create a new package and call it “com.gcu” and then create a new “model”, “controller”, and “dto” package within the com.gcu package. Copy over the classes from the repository into their respective folder and then copy the WebContent folder and replace any preexisting files with those from the repository.

To the run the application, start up the server and open a web browser. Use this URL to start the application:  
http://localhost:8080/PROJECTNAMEHERE/user/home

**General Technical Approach:**

The approach Joe and I will follow when designing the Project will be as follows, Login/Registration, Database Connectivity and security, then the pages the user uses to see, add, and remove their books, and finally admin functions. Discussion during brainstorming, allowing users to track their book progress, they will track the pages they read per day, and may be awarded for certain page amounts per month or day.

One issue that became apparent while setting up the MVC framework was that when using validation annotations, if there were fields used during registration that were not needed for login, the program would always fail the validation check. To get around this, we split up the User Model class into two separate classes, a normal User which is used for login, and a Register User which includes additional information that involves that user along with a super in the constructor so that variables from the normal User can also be accessed.

**Key Technical Design Decisions:**

The program will be developed using the Spring framework, allowing for easier maintained and additional coding later, separation of classes, DAOs and user interfaces. As well as easier to access database through use of a method class so that the database username and password can’t be easily accessed.

Using a datatable built with JavaScript has proven to be very helpful to create a dynamic list of books, as well as a means of linking to each individual item for further editing or deleting.

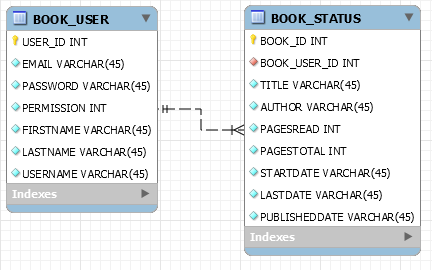
**Known Issues:**

*The navbar worked for awhile when the code was freshly implemented, however, upon demonstrating the website for Loom, the navbar stopped working.*

**Risks:**

Navbar does not function correctly, potentially cutting off individuals with mobile devices from accessing any major features.

**ER Diagram:**

**

**DDL Scripts:**

CREATE TABLE BOOKSTRACKER.BOOK\_USER(

USER\_ID INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY(START WITH 1, INCREMENT BY 1),

EMAIL VARCHAR(45) NOT NULL,

PASSWORD VARCHAR(45) NOT NULL,

PERMISSION INTEGER NOT NULL,

FIRSTNAME VARCHAR(45) NOT NULL,

LASTNAME VARCHAR(45) NOT NULL,

USERNAME VARCHAR(45) NOT NULL,

CONSTRAINT user\_key PRIMARY KEY (USER\_ID)

);

INSERT INTO BOOKSTRACKER.BOOK\_USER(EMAIL, PASSWORD, PERMISSION, FIRSTNAME, LASTNAME, USERNAME) VALUES('TestUser01@email.com','testUser01',1,'Test','User','TestUser01');

INSERT INTO BOOKSTRACKER.BOOK\_USER(EMAIL, PASSWORD, PERMISSION, FIRSTNAME, LASTNAME, USERNAME) VALUES('TestUser02@email.com','testUser02',1,'Test','User','TestUser02');

INSERT INTO BOOKSTRACKER.BOOK\_USER(EMAIL, PASSWORD, PERMISSION, FIRSTNAME, LASTNAME, USERNAME) VALUES('TestUser03@email.com','testUser03',1,'Test','User','TestUser03');

INSERT INTO BOOKSTRACKER.BOOK\_USER(EMAIL, PASSWORD, PERMISSION, FIRSTNAME, LASTNAME, USERNAME) VALUES('Admin@email.com','adminPassword',0,'Admin','Admin','Admin');

CREATE TABLE BOOKSTRACKER.BOOK\_STATUS(

BOOK\_ID INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY(START WITH 1, INCREMENT BY 1),

BOOK\_USER\_ID INTEGER NOT NULL,

TITLE VARCHAR(45) NOT NULL,

AUTHOR VARCHAR(45) NOT NULL,

PAGESREAD INTEGER NOT NULL,

PAGESTOTAL INTEGER NOT NULL,

STARTDATE VARCHAR(45) NOT NULL,

LASTDATE VARCHAR(45) NOT NULL,

PUBLISHDATE VARCHAR(45) NOT NULL,

CONSTRAINT userid\_key FOREIGN KEY (BOOK\_USER\_ID) REFERENCES BOOKSTRACKER.BOOK\_USER(USER\_ID)

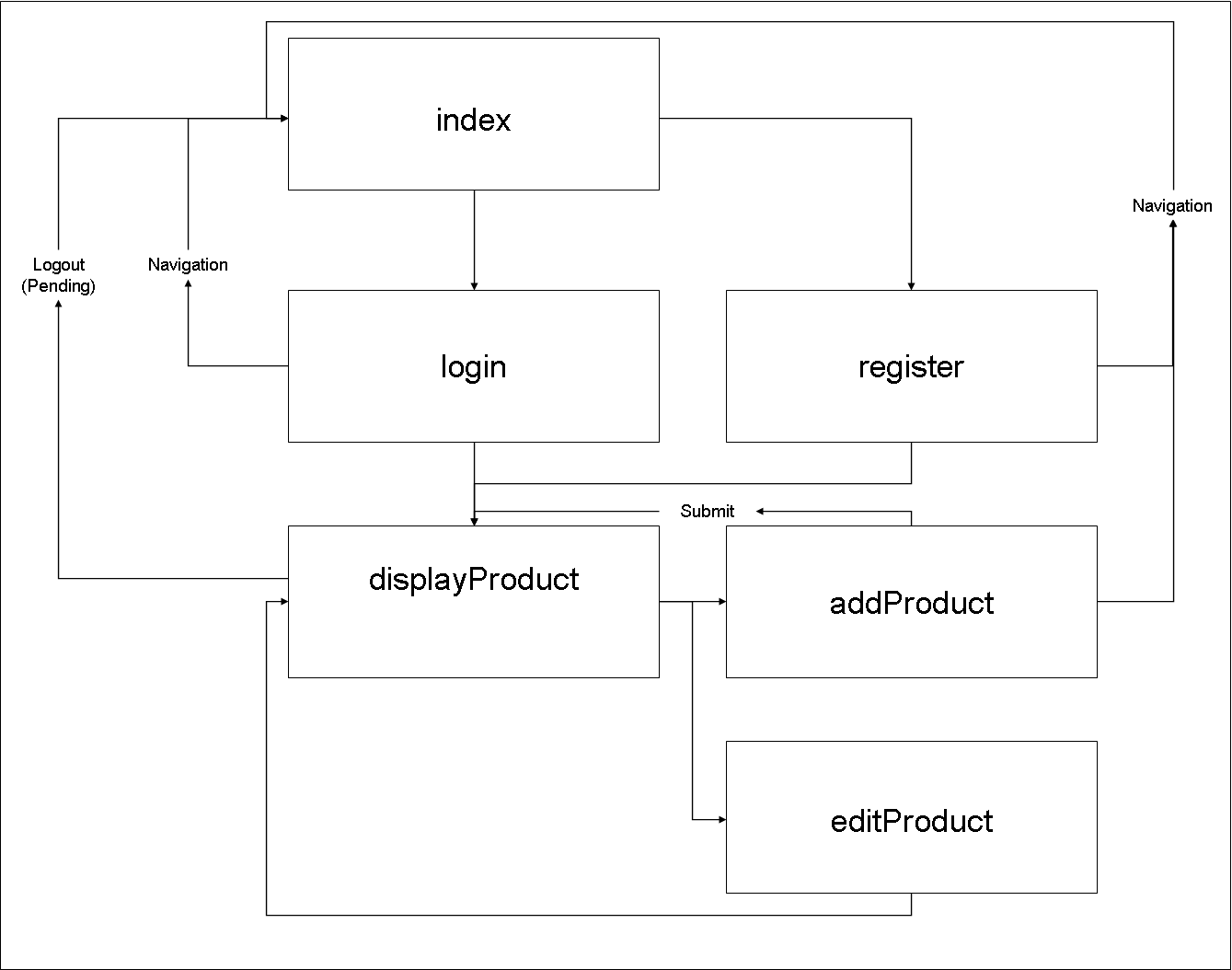
);

INSERT INTO BOOKSTRACKER.BOOK\_STATUS(TITLE, AUTHOR, PAGESREAD, PAGESTOTAL, STARTDATE, LASTDATE, PUBLISHDATE, BOOK\_USER\_ID) VALUES('Test Book01','Test Author01',1, 50,'01/01/01','01/01/01','01/01/01',1);

INSERT INTO BOOKSTRACKER.BOOK\_STATUS(TITLE, AUTHOR, PAGESREAD, PAGESTOTAL, STARTDATE, LASTDATE, PUBLISHDATE, BOOK\_USER\_ID) VALUES('Test Book02','Test Author02',2, 50,'02/02/02','02/02/02','02/02/02',1);

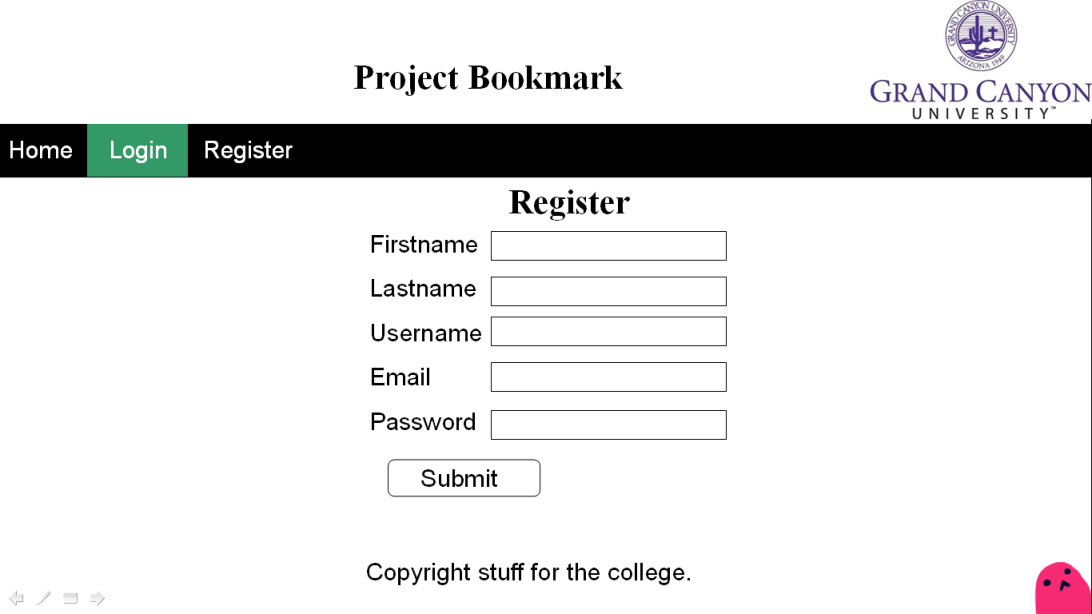
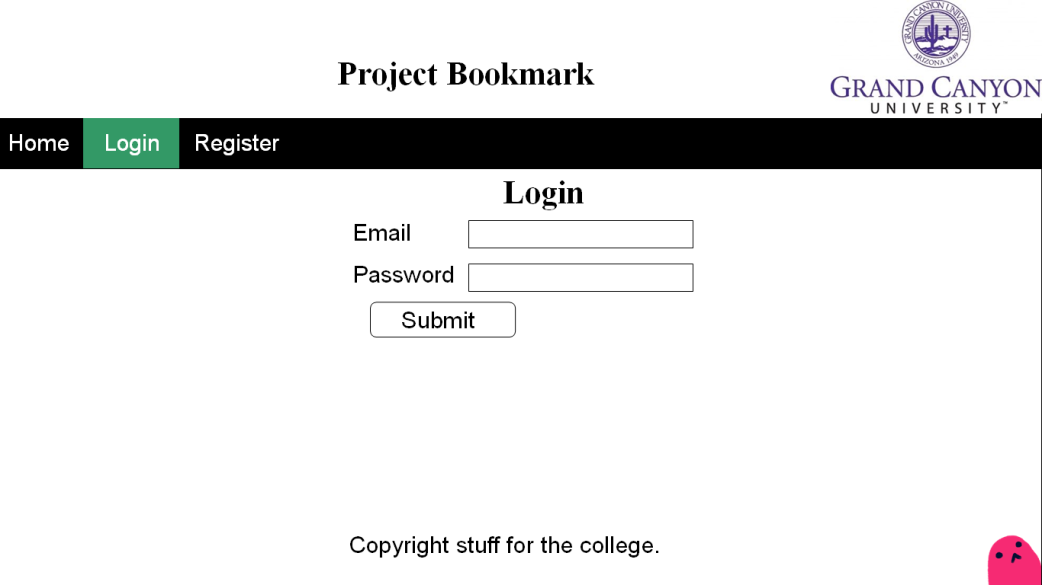
**Sitemap Diagram:**

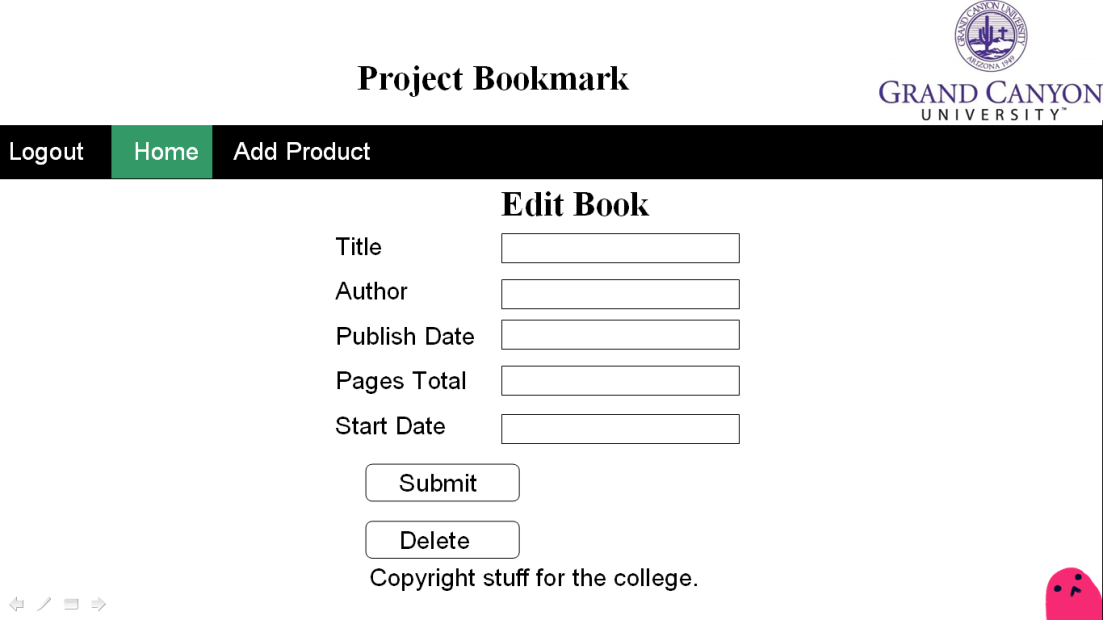
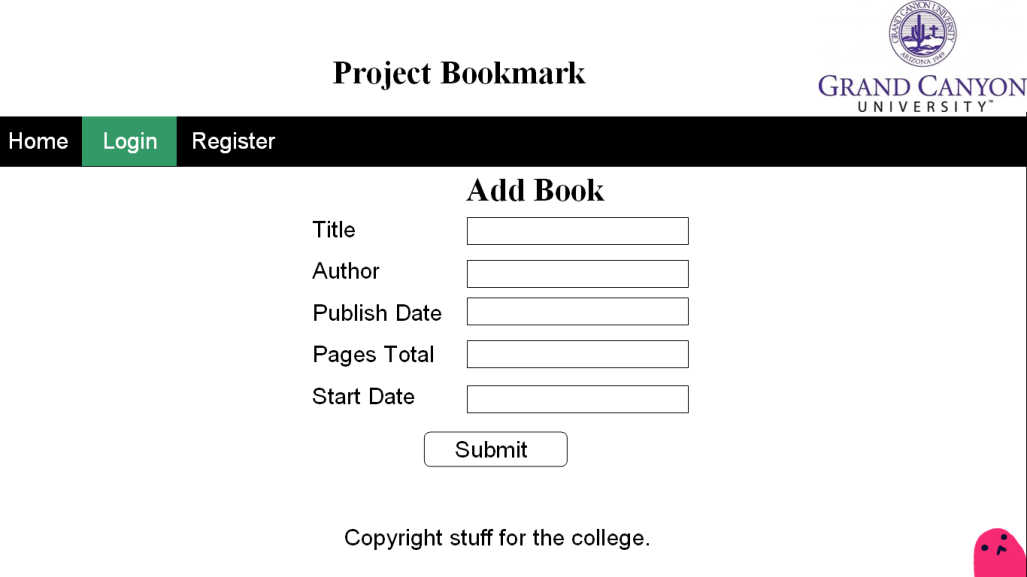
Current Sitemap:

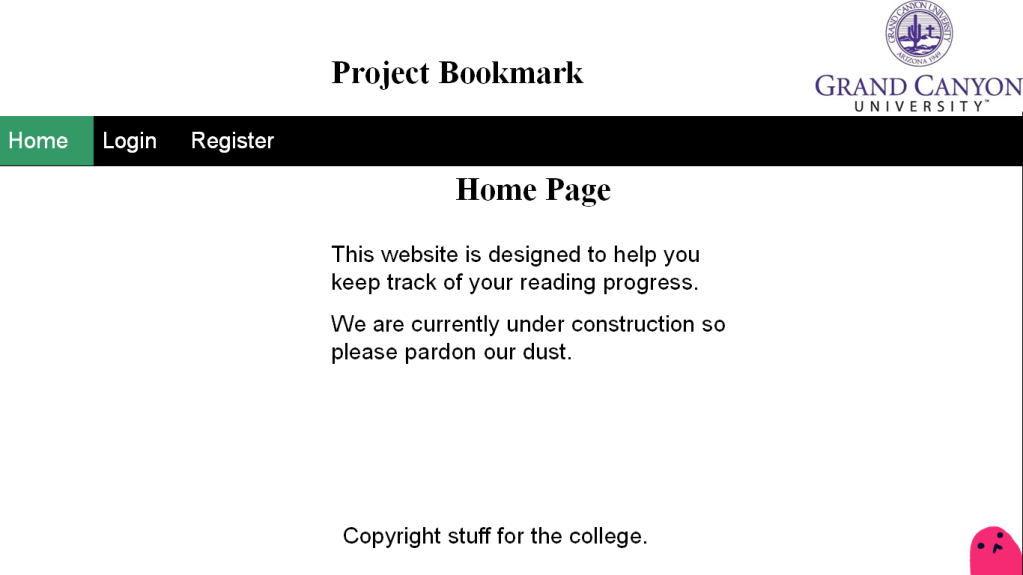
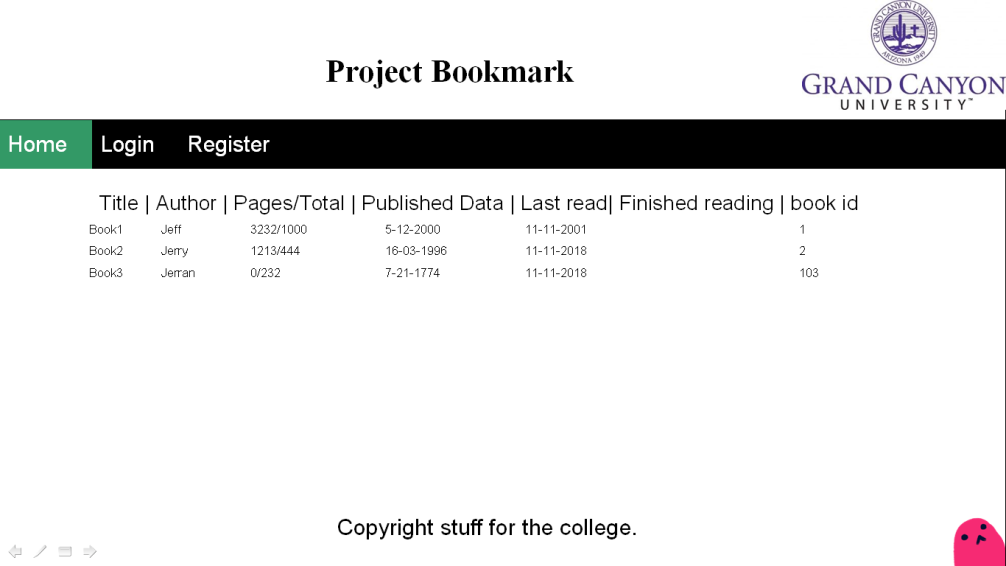


**User Interface Diagrams:**Created in PowerPoint.

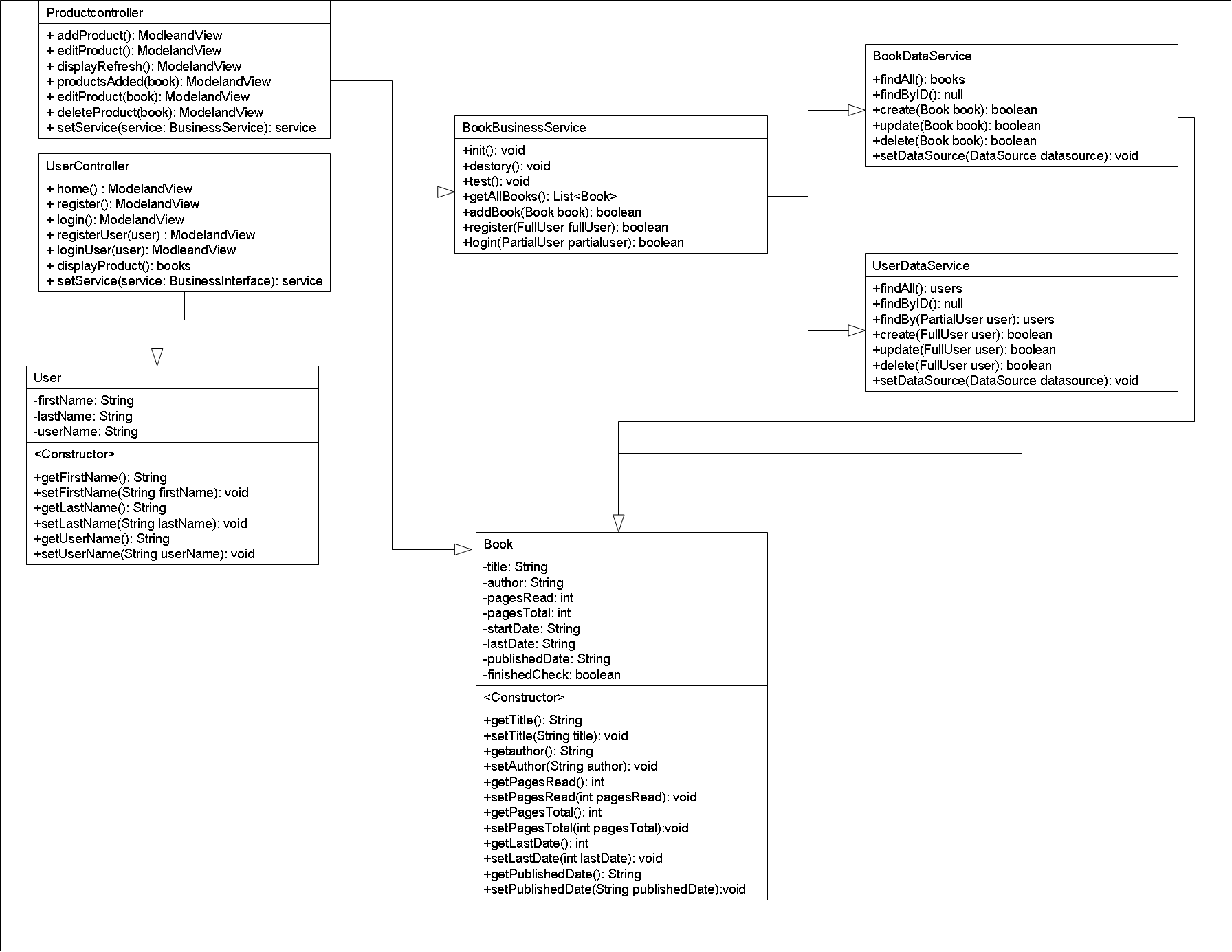
*.*

****

****

****

**Class Diagrams:**



**Service API Design:**

Bootstrap 3.3.7 is used for formatting the website and to provide a navbar.

JQuery and Oracle’s Datatables are used for displaying the product.

Inside the DisplayProduct table, the url inside the ajax call needs to be updated for the rest with the correct information for the table needs to be populated.

$.ajax({

type : "GET",

url : "/ProjectBookmarkRX/service1/books",

dataType : "json",

success : **function**(data) {

**Security Design:**

Login and registration both check the given email address within the JSP to the database to see if an existing match can be found. In the registration, it checks to see if the provided email address is valid. If it does not find one, then it will fill in the user’s information. If there is an existing email, then the website will reject the user’s information.

With login, it checks to see if the email and password match any records that are saved to the date base, if it returns with a match, then the user is taken the DisplayProduct page.

**Other Documentation:**