VENS

Online Bookstore Design Report

**Student Name: Student Number**

Varsha Ragavendran 213193065

Soo Bae 212209045

Nusayba Moftah 212507091

Enxhi Haxhiu 212418588

**Table of Contents**

Introduction .................................................................................................................................. 1

How does our design facilitate the client and meet all her/his needs?........................................1.1

Who can access and use our website?.........................................................................................1.2

Architecture...................................................................................................................................2

UML Diagram..............................................................................................................................2.1

Sequence Diagrams for use cases.................................................................................................2.2

Used pattern……………………..................................................................................................2.3

Implementation details.....................................................................................................................3

Classes Implementation................................................................................................................3.1

Tables and Database.....................................................................................................................3.2

Team member contribution............................................................................................................4

1. **Introduction:**

Web development is a field of developing web sites for the Internet; it can range from simple static single pages into complex web-based electronic businesses. An online bookstore is a web application where the customer can browse and purchase books online.

* 1. **How does our design facilitate the client and meet all her/his needs?**

We have a very ‘user-friendly’ and simple design. When the user first accesses our website he/she are automatically directed to our welcome page in which he/she can search for a book, sign in/ register into an account etc. Depending on what the user is looking for he/she might also want to try to search by the name of the author, title or in all. Also to make it easier for the user, we have an advanced search in case there is a specific book the user is looking for. In the search result page, the user can sort by title, author and price. Furthermore, the user is able to narrow down the search by selecting a specific category or price range.

Once the search results appear the user can select a book for more details about the book including user reviews. The user is then able to add a specific book into the shopping cart, and from there is able to order the book by entering his/her credit card information. Furthermore, users are able to rate and review a specific book they have read.

Only signed in users are able to purchase a book or add a review. Any user is eligible to create an account by registering on our page. Our website is also equipped with resources to assist our users. For example, there is a help page where we have answered most frequently asked questions. Moreover, if the user has a question that is not answered they can contact us using the information in Contact Us page. We also have a page that informs our users about our company and design team.

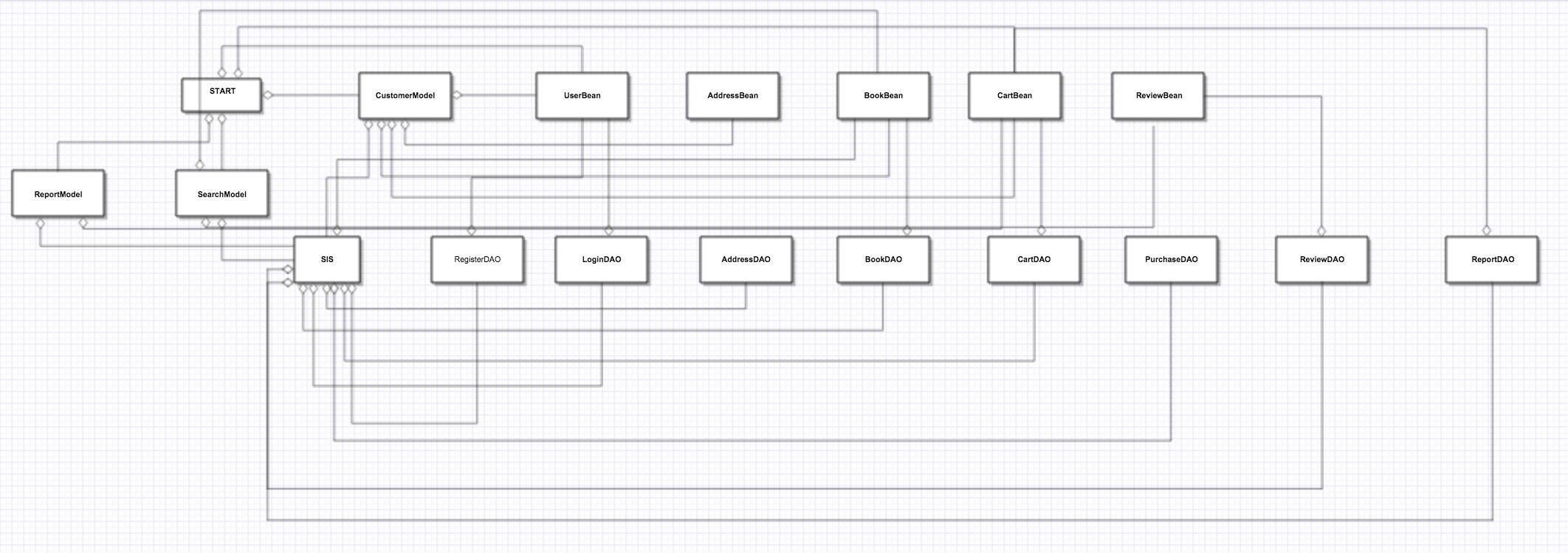
**1.2 Who can access and use our website?**

Users can access our website by logging into it, purchasing books, registering, searching. One login information for user is: username: scott, password: kscott.

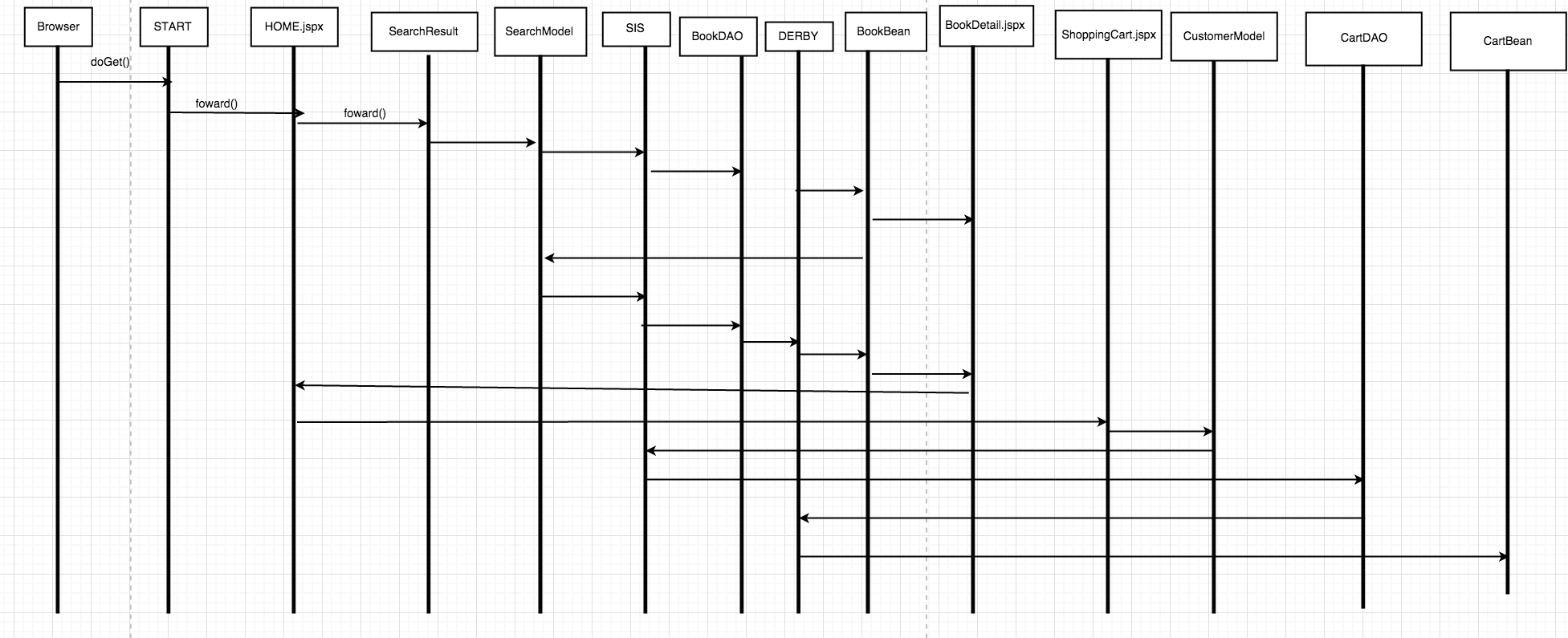
Admin is one of the principal users who is able to generate reports based on books sold each month. The login information for admin is as follows: password: admin, username: admin.

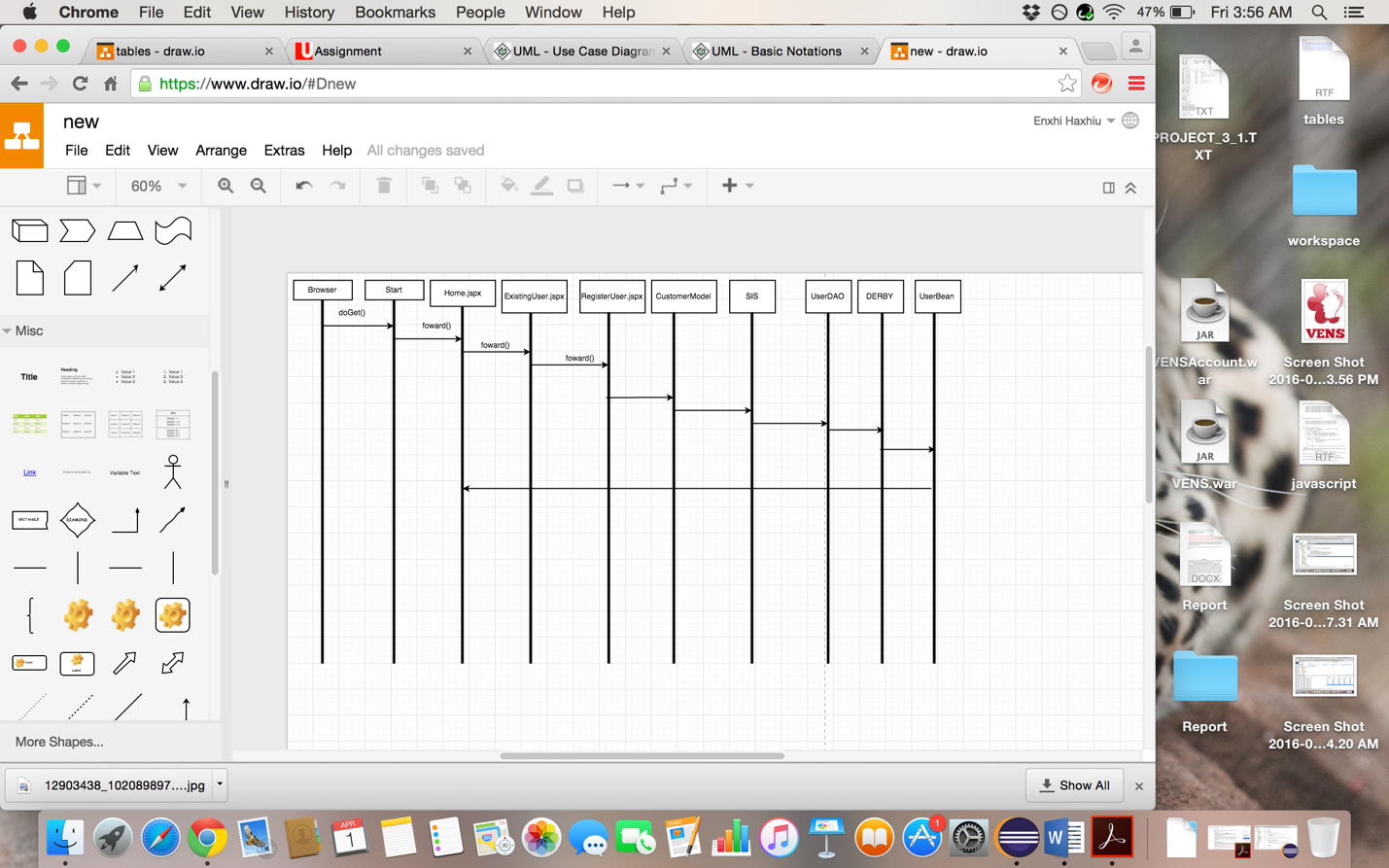
1. **Architecture**
   1. **UML Diagram**

The following pictures shows the UML of each implemented class and the relationship between each other.



**2.2 Sequence Diagrams for use cases**

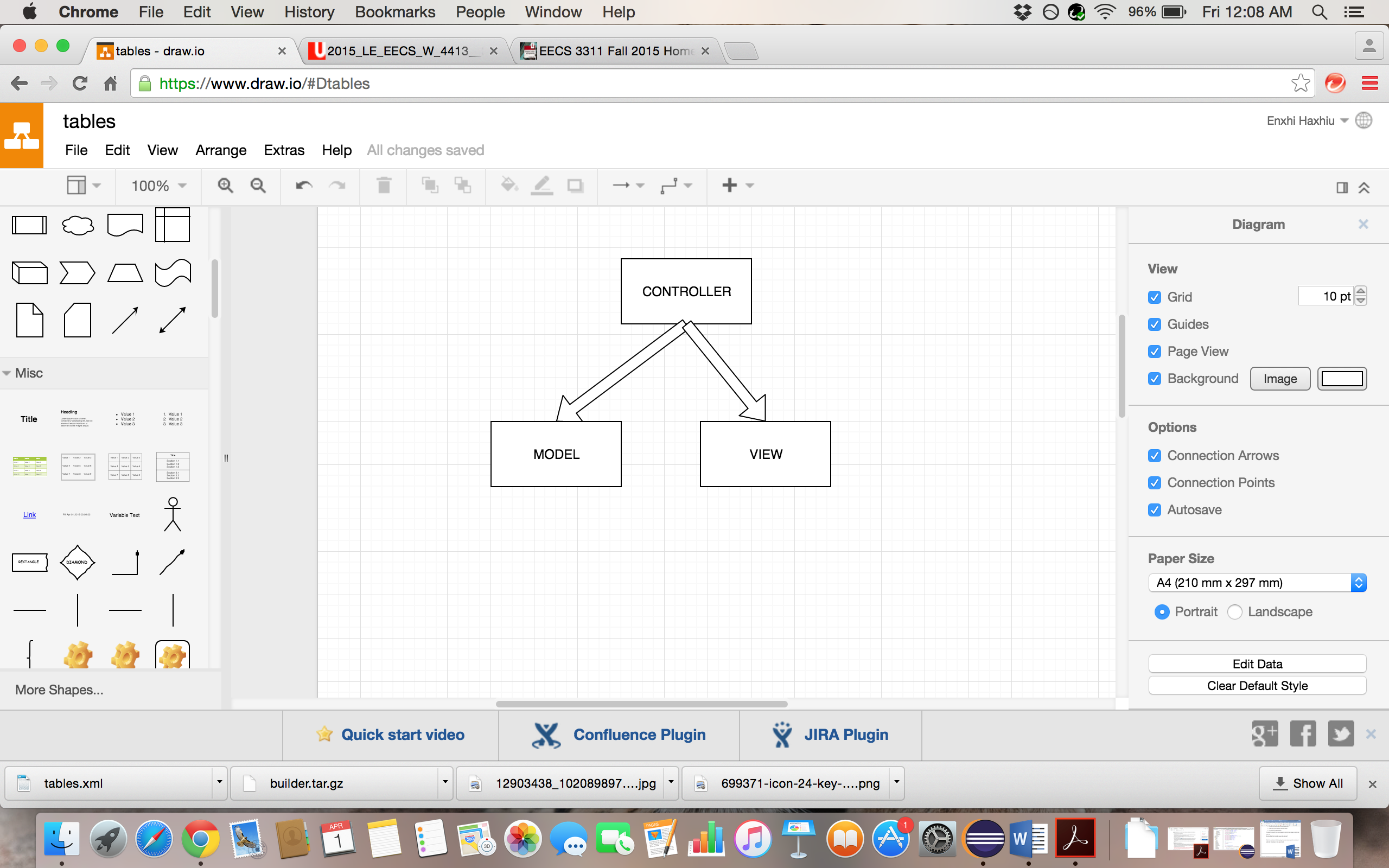
**First test case:** The user runs the servlet and goes to the home page. From there it searches for a book in thesearch bar and it is redirected to the book result page. After clicking on the book image or on the book title it is sent to the book detail page where the user can view the details of the book and to be able to add it to the cart. The user adds the book to the cart and goes to the purchase page to look for the books that he has there. The following is the class sequence diagram for the first test case.

**Second test case:** The user runs the servlet and it is sent to Home page. He presses the ‘Sign in’ link and it is redirected to the Login page. The user clicks on the “want to create an account link?” and proceeds to the Register page and inputs his details to have an account. After he presses submit it is logged in and send to the home page. The following is the class sequence diagram for the second test case.

**2.3 Used Pattern**

We decided that for the project to use the MVC pattern. We used Model View Controller (MVC) pattern because:

1. It is an effective way to split the code and assign different tasks to specific packages.
2. It is easier to test the code and to track any possible errors.
3. It makes the code more organized, usable and readable.
4. It is suitable for big implementations.

****The tradeoff for using MVC was that changing the model will cause changes in controller and in the view.

1. **Implementation Details**

**3.1 Classes Implementation**

Model is the combination of different packages that contain classes responsible to maintain and update data. The packages are: bean, DAO and model.

The bean package contains classes that hold information about the tables in our database system. They contain private attributes, a public constructor that instantiates all the fields and public methods that give or update information about the fields.

The DAO package contains classes responsible for communicating with DBMS. All the classes contain the DataSource that takes the connection from the pool. Depending on the function of every class, they contain specific methods that retrieve/update the tables of the database.

The model package contains 4 classes: SIS, SearchModel, CustomerModel and ReportModel.

SIS class has as objects all the classes of the DAO package. It instantiates them in its constructor and it uses their methods to get the information from the tables in database.

SearchModel has an SIS object and a BookBean object. It has all the methods that the user needs to get the books information from the book table in the database and return the user what its needed. The search is done by whatever the user inputs, the title, the author or the category.

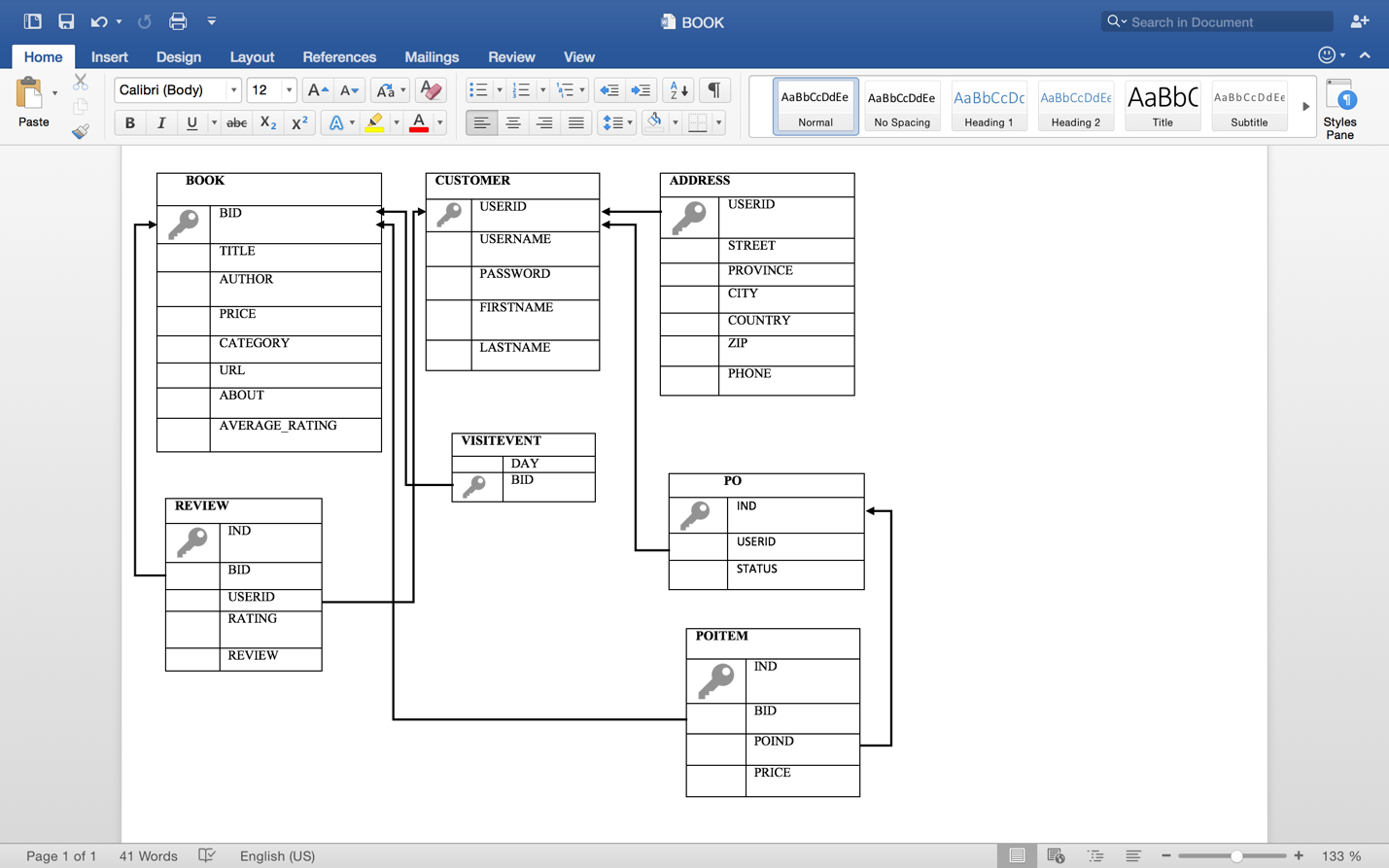
CustomerModel has an SIS object and other objects from the bean package like UserBean, CartBean and AddressBean. The CustomerModel is responsible for retrieving the information from the customer and address table.

The view contains different jspx files that help show the user different parts of our website like for example the Home, Register, Login and so on. We have used CSS to style the pages and also javascript to check for errors from the user input and also to display some tags when necessary.

The Controller is implemented in the ctrl package and it contains one servlet: START. We decided to have one servlet for better performance of the code. The servlet contains CustomerModel and SearchModel objects and also objects from AddressBean, CartBean and UserBean. The controller uses the methods from the model to make it possible the display of the view.

In the following picture we created the simple UML diagram for all classes and how they connect to each other.

The Admin is able to generate reports based on books sold each month. . The Listener is used to add events to the VISITEVENT such as user viewing a particular book or adding a book to the shopping cart or purchasing a book. Based on the page visited and noticed by the listener, the VISITEVENT table event type column gets updated accordingly (cart, view, purchase).

**3.2** **Database and Tables**

To connect to the tables we used the open source relational database: derby. The following pictures show the schemas for every table and also the connection between them. The key image shows the primary key of every table and the arrows from one table to another show the foreign keys.

1. **Team member contribution**

Team VENS worked very hard throughout the semester to complete this project. We met very often, setting for over selves’ tasks and deadlines. When we did not meet we would communicate online to see how we progressed and what we had left to do. When we were together we spent most of the time combining the codes, talking about the difficulties that we had and how could we solve them. After our discussions, we tried to work on our own and help each other when was needed.

This group project gave us the ability to learn from each other and master skills we did not have. We were unfortunately unsuccessful in completing the Partners page using the SOAP/REST. That was mostly due to the limited time available.

This is how we split the work amongst ourselves:

**Varsha Ragavendran**: worked on connecting the following pages to the information in the database: ExisitingUser, RegisterUser, Purchase, Shopping Cart, Admin account. Varsha also worked on the Listener.

**Soo Bae:** implemented the home page, she made possible the search result and advanced search. She also combined most of the codes in an efficient and elegant way. She also worked register, login, and searching books. She worked on error checking for account pages. Organized the code in general.

**Nusayba Moftah**: worked on connecting the following pages to the information in the database: Account, ForgotPassword, ForgotUsername, Help, ContactUS, AboutUs pages. Nusayba also worked on the JUnit testing for the Beans and implemented the reviews/ratings, sorting, and narrowing down the search.

**Enxhi Haxhiu**: worked on the HTML design the jspx pages for: ExistingUser, RegisterUser, Account, Purchase, my404, ForgotPassword, ForgotUsername. Enxhi also did error-checking for all the previous mentioned pages and worked on the report and diagrams.