**PROJECT REPORT**

**Used Book Website**

**For**

**Amrita University (India)**

**by**

**Sreerag Sreenivasan (820911878)**

*Under the guidance of*

Dr. Carl Eckberg

Graduate Advisor, Dept. of Computer Science



**Department of Computer Science**

**San Diego State University**

**San Diego, CA, 92115**

**2018**

**SAN DIEGO STATE UNIVERSITY**

The Below Mentioned Evaluation Committee Approves the

Project by Sreerag Sreenivasan

**Used Book Website for Amrita University**

Dr. Carl Eckberg

Department of Computer Science

Dr. Roger Whitney, Graduate Advisor

Department of Computer Science

Dr. Xiaobai Liu, Professor

Department of Computer Science

**ACKNOWLEDGEMENT**

I express my sincere piece of gratitude to Dr. Carl Eckberg, Graduate Advisor, Department of Computer Science for his moral support, encouragement, suggestions to complete this project.

Furthermore, I would like to express my deep sense of gratitude to all the staff and faculties of Amrita University for having faith on me and giving me the permission to do this work as my capstone project.

I acknowledge the help of Prof. Krishnakumar Kalathil, Faculty, Department of Sociology, Amrita University for being a great force behind all my efforts through his advice, guidance, and encouraging nature throughout the course of the project.

Finally, I would like to express my gratitude towards my parents for their moral support for making this project a success.

**ABSTRACT**

The goal of this project is to develop first working prototype of a Used Book Website using advanced web technologies. It is aimed at students of Amrita University, India where they can purchase used books at a lesser price. Students can also submit their old book at the end of the semester to the library and the library officials finalizes a price for the book based on its condition. The book will be published in the used book website managed by the library. Students within the university will have access to the website and they can purchase it from there at the published price. The whole idea is a trial attempt by the university and if found successful, this project will be handed over to a commercial vendor and then they will operate on behalf of the library.

The website will have 2 parts. One will be an admin portal to which only the library officials will have access. The other one will be the actual website that looks something like amazon.com, which we can call ‘store-front’. Only registered students of Amrita University will have access to ‘store-front’.

**INDEX**

1. **Overview**
   1. **Introduction**
   2. **Tools used in this project**
2. **Analysis and Implementation**
   1. **Stacked Regression Model**
      1. **Dataset Definition**
      2. **Data Cleaning**
      3. **Feature Engineering**
      4. **Model Definition**
         1. **Lasso Regression**
         2. **Gradient Boosting Regression**
         3. **XGBoost**
         4. **LightGBM**
      5. **Result**
   2. **Image Analysis**
      1. **Scale Invariant Feature Transform**
      2. **Speeded Up Robust Feature**
      3. **BRIEF**
   3. **Android Application using Firebase**
      1. **Firebase Storage**
      2. **Firebase Authentication**
      3. **Firebase Database**
3. **Future Work**
4. **References**

**LIST OF FIGURES**

1. House in Boston
2. Comparison between Artificial Neural Network Model and Hedonic Model
3. Hedonic Pricing Method
4. Figure of Stacked Regression Method
5. Outliers present in datasets
6. Removal of Outlier
7. SalesPrice Distribution
8. Probability plot
9. Log Transformation of Target Variable
10. Showing the amount of data in each graph
11. Comparison between lasso and Elastic Net
12. Model Score for Enet, GBoost, KRR, lasso
13. Meta Model
14. Result of each feature
15. SIFT on Image 1
16. SIFT on Image 2
17. SIFT on Image 3
18. SIFT on Image 4
19. For Detecting feature points
20. Scaling of image
21. Feature Extraction using SURF
22. Image Surf Process
23. Activity Lifecycle
24. Firebase
25. Firebase Install
26. Login Page
27. Register Screen
28. After Image Taken
29. Image for Submission

**INTRODUCTION**

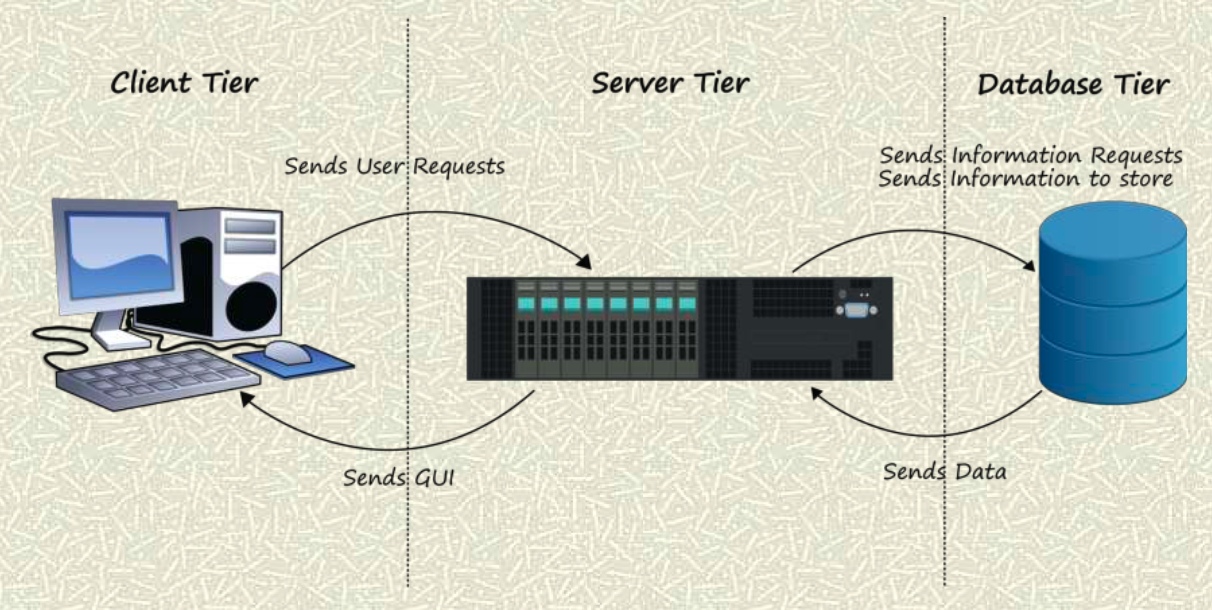
It is a known fact that the price of undergraduate and graduate level textbooks are quite high. Some students completely relay on the library for text books. Those people who buy new textbook used to abandon them the end of the year/semester. The thought about having a used book website where students can sell and buy books came from this realization.

This platform managed by the library will help students to get rid of their old books for a good price as well buy books at a reduced price, sort of win-win situation. The plan is to implement this on a trial basis with a working prototype, if the response is good then improve the website with more features.

To develop the website I was asked by Amrita University to use the most advanced web technologies and hence this application’s core technical components include Angular 5 for front-end and Java based back-end using couple of open-sourced Spring technology projects like Spring MVC, Spring Security, Spring Boot, Spring Data etc. On the Database side I am using MySQL and an ORM tool called Hibernate to map Java Objects to database.

The Amrita University officials are more concerned about the core functionality of the web application than the CSS part of it. So I am using a CSS package just for the demo purpose which I will remove when I handover the project to them. Their internal IT team will work on it with some tweets to bring it to their standards.

**HIGH LEVEL ARCHITECTURE DESIGN**



The project uses a 3 tier architecture. Three-tier architecture is a client–server software architecture pattern in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms

Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or technology. For example, a change of operating system in the *presentation tier* would only affect the user interface code.

In the web development field, three-tier is often used to refer to websites, commonly electronic commerce websites, which are built using three tiers:

1. A front-end web server serving static content, and potentially some cached dynamic content. In web-based application, front end is the content rendered by the browser. The content may be static or generated dynamically.
2. A middle dynamic content processing and generation level application server (e.g., Symfony, Spring, ASP.NET, Django, Rails).
3. A back-end database or data store, comprising both data sets and the database management system software that manages and provides access to the data.

**Traceability**

The end-to-end traceability of data flows through *n*-tier systems is a challenging task which becomes more important when systems increase in complexity. The Application Response Measurement defines concepts and APIs for measuring performance and correlating transactions between tiers. Generally, the term "tiers" is used to describe physical distribution of components of a system on separate servers, computers, or networks (processing nodes). A three-tier architecture then will have three processing nodes. The term "layers" refer to a logical grouping of components which may or may not be physically located on one processing node.

3-Tier Architecture provides the following benefits.

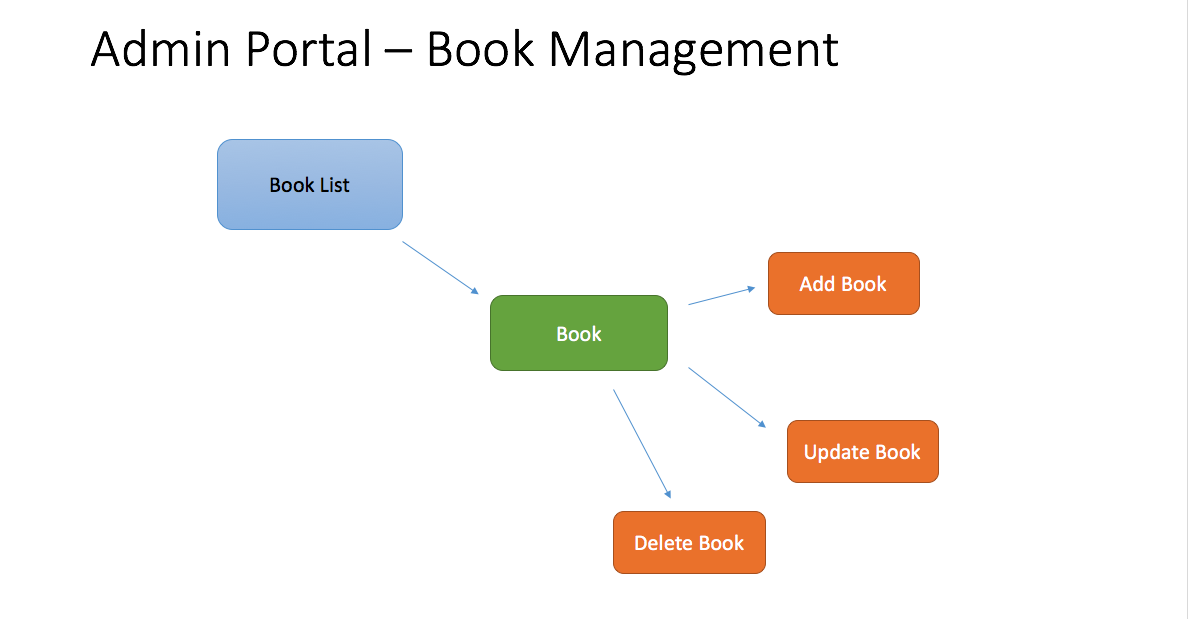
* **Scalability**—Each tier can scale horizontally. For example, you can load-balance the Presentation tier among 3 servers to satisfy more Web requests without adding servers to the Application and Data tiers.
* **Performance**—Because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers. If needed, you can load-balance any tier.
* **Availability**—If the Application tier server is down and caching is sufficient, the Presentation tier can process Web requests using the cache.

No Architecture design is perfect and hence it has its limitations also

* You must manually push templates, assets, private assets, uploaded images, and uploaded files from the Application tier to the Presentation tier. You may use a tool like Robocopy to do this.
* You create content (HTML, assets, PageBuilder pages, and so on) only in the Application tier. You can only view content from Presentation tier.
* Because 3-Tier Architecture uses WCF, it requires:
  + the 8.5 or higher Framework API and databinding. You cannot use API calls outside of the Framework API.
  + templated server controls.
  + *widget*s created by the Framework API.
* Because the business logic executes on the Application tier, its Web site directory must be identical to the Presentation tier's.

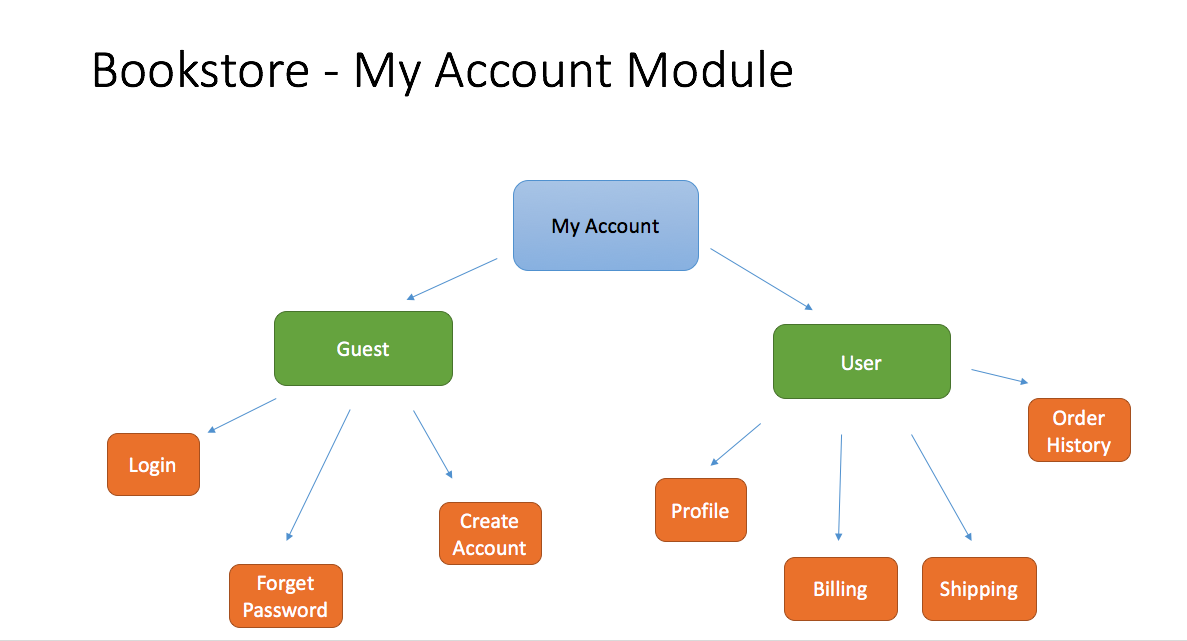
**KEY MODULES AND THEIR USE CASES**

***Admin -- Book Management Module***

****

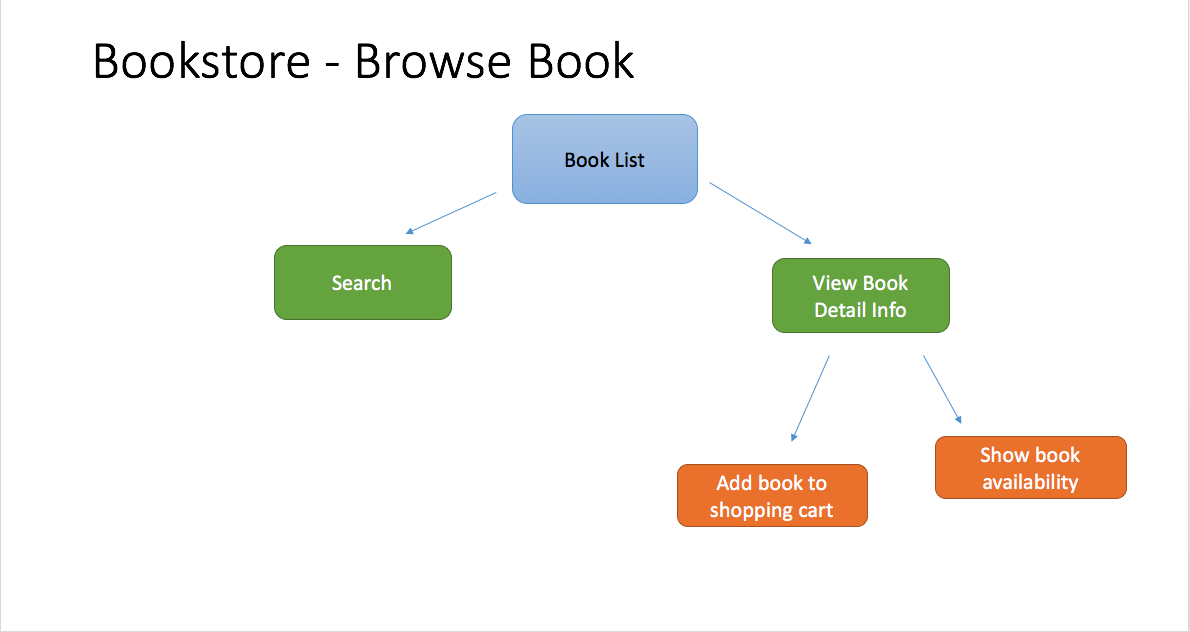
The Admin Module is a portal for library associates and is going to be simple which list all the books available in collection. There will be an authentication to verify if the admin portal user is an authorized user. Through the admin portal, user can add a new book to the database, edit details of an existing book and delete particular book from the database etc. In the implementation point also it is fairly simple. The most challenging part will be handling the session management and performing group operations. Below are the details of the different modules in the store-front.

***Store-Front -- My Account Module***

****

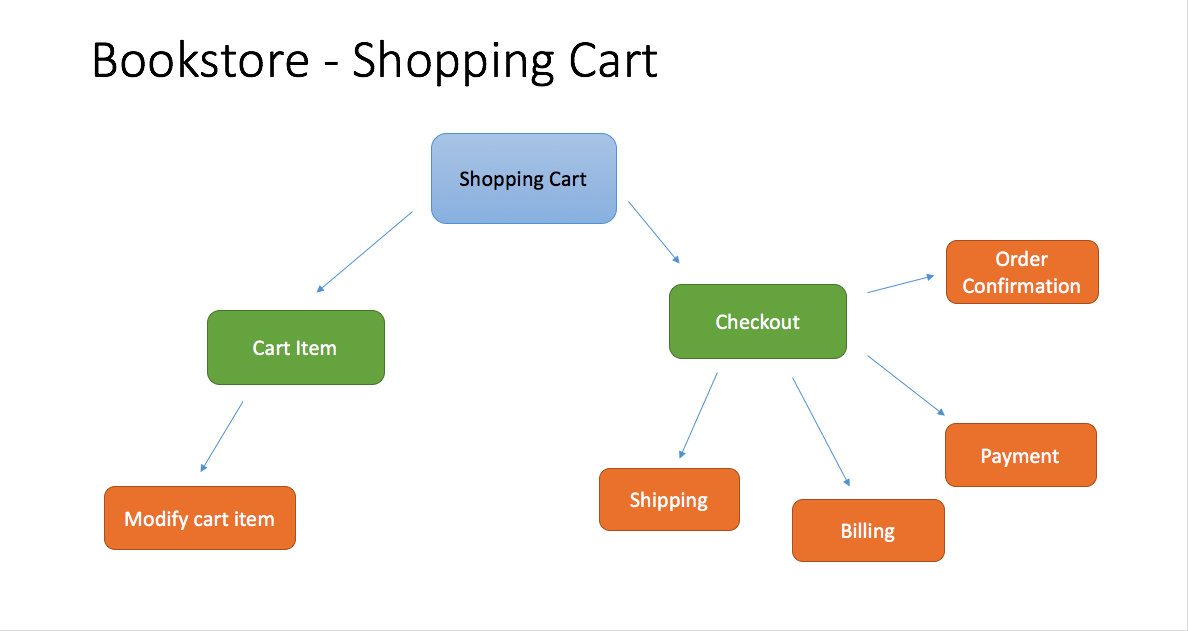
The Store Front is divided into 3 functional modules namely My Account, Browse book and Shopping Cart starting with the my account module. Students of Amrita University can register themselves in used book website’s store-front and create an account. They can browse through the books as a guest user as well. But to make a purchase they have to create an account. Some of the other use cases for the guest user can creating an account, logging in to his already registered account and Resetting his password with the forgot password option. Once the a user is logged in to his account, he will have options to update his personal info in the profile section of the website. The user will also have options to save a default billing address, shipping address which will be preloaded when he places an order for the next time. This feature saves time and adds as a convenience feature for the user. A history of all the orders placed by the user is past is maintained in the system and the user will have the option to view this history data through his profile.

***Store-Front – Browse Book Module***

****

Browsing book details is a common option available for Guest users as well as registered users. The is a Book Title pattern search implemented to make things easy. The result of the search as well as normal listing of books will give a snapshot about the book’s info like Title, Author and Price. If the user wants to see more info about the book, he/she can click on the book and it will display the detailed book info. User can add the book to his cart at this point and continue the shopping or do a checkout.

***Store-Front – Shopping Cart Module***

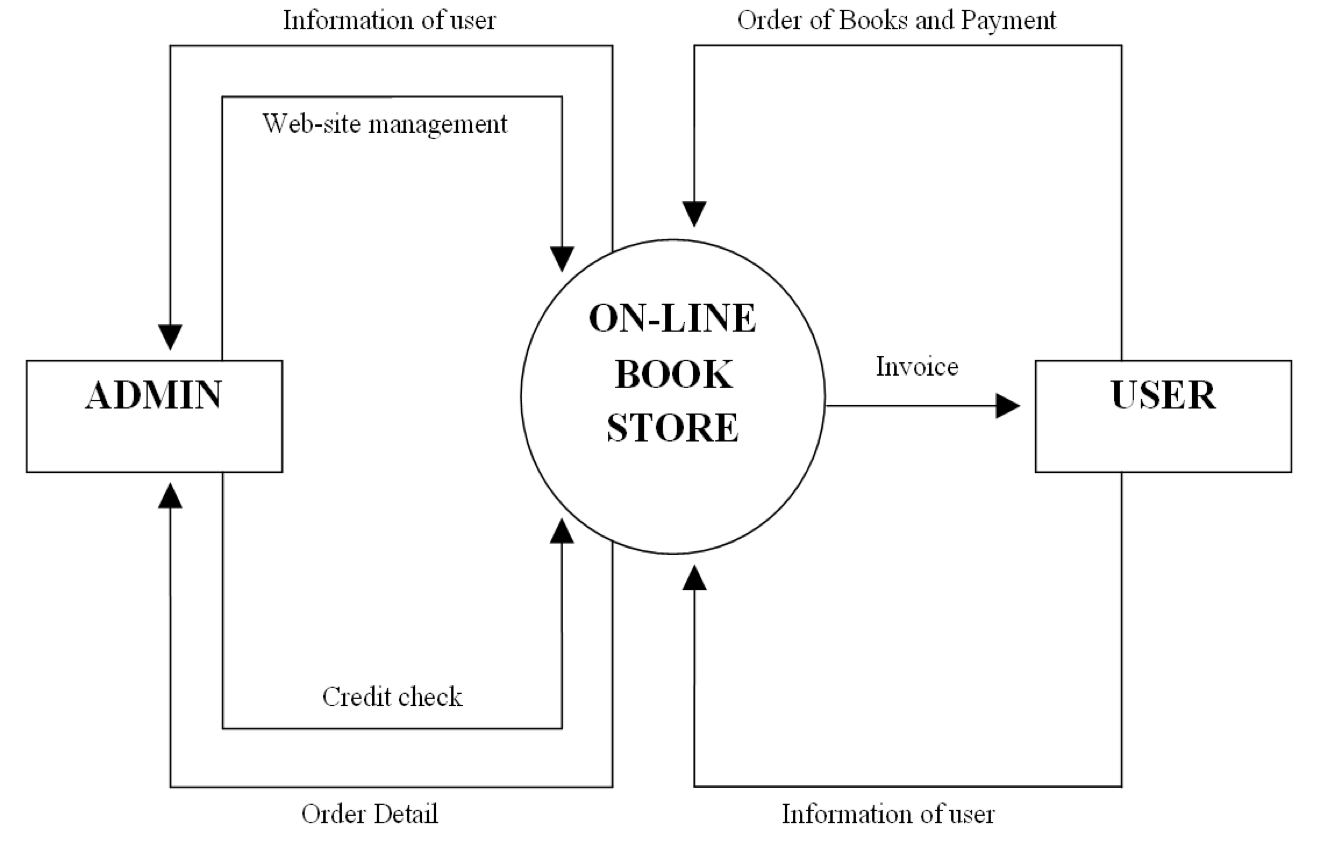
****

The module with the most functionality is the Shopping cart module. The functionality in this module starts from the point where the users adds a book to his cart. He will have the option of editing this cart item at any point of time such as removing the item from the cart, increasing the quantity of purchase etc. We will have to validate it against the available stock each time if he increases the number of items for purchase. The checkout option enable the user to take all the items in his cart for billing. If the user has already updated his personal information in his profile we need to take those and update them as the shipping, billing, payment details in this order. Of course the user can manually edit these fields as well. Once the user places the order an email should be send to the user’s email id confirming the order.

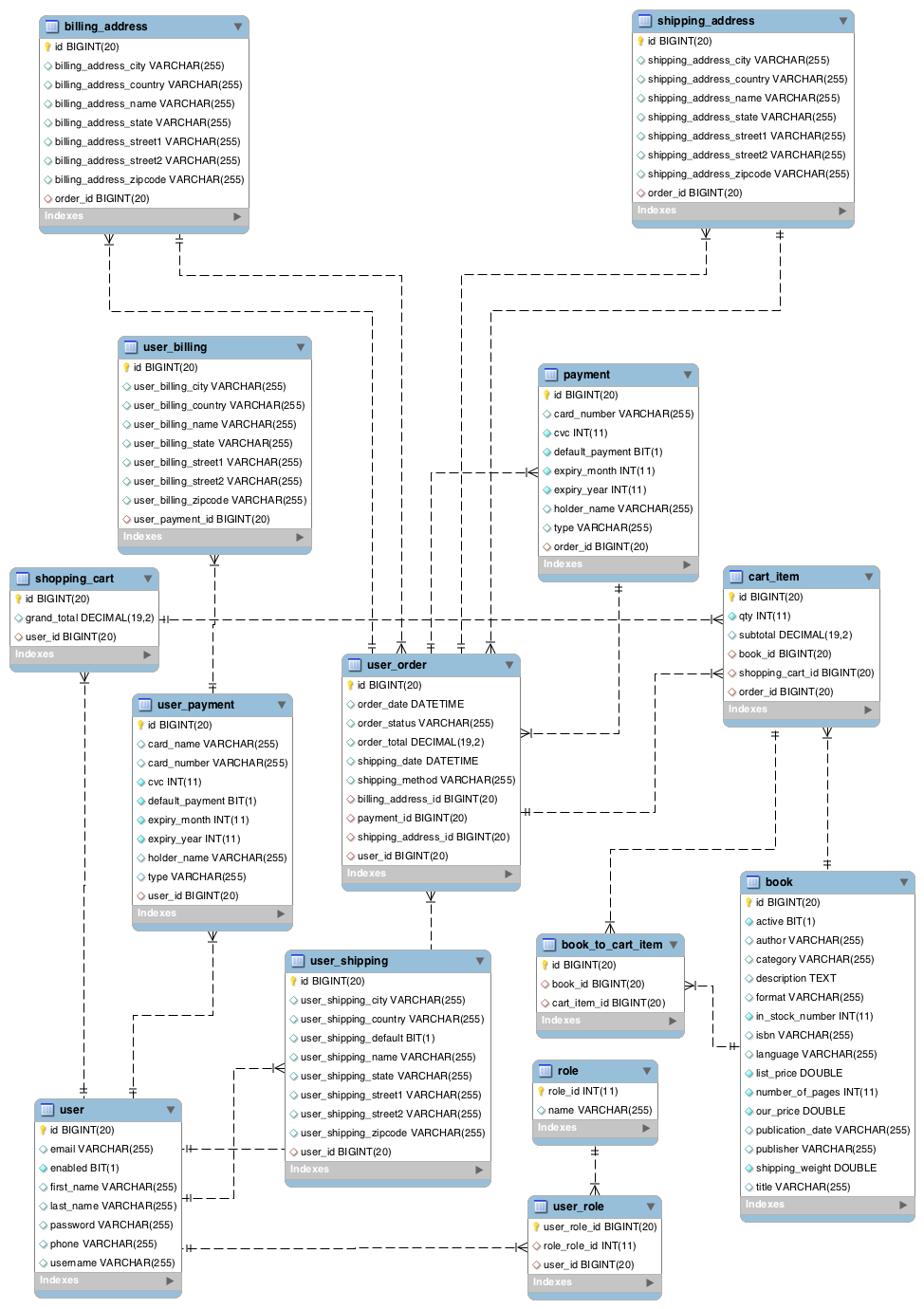
**DATA FLOW DIAGRAM**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model



As you could imagine the admin portal and the store-front will be interacting with the same database. The Admin portal will have the website management privileges and there will be only one authorized user who has access to the admin portal. If there is a need for creating another user with admin privileges that has to be done through the backend. It has been decided to keep the functionality like this because the need for having a second user with admin privileges is very less. The normal user(Customer) interacts only through the store front and that’s how he can place order for a book and do the payment. The payment done through the front will be credited to the admin’s account and the customer will get order confirmation invoice. The order placed by the user should update the available stock information in the database. Since admin portal uses the same database, admin will see the updated values when the he logs in.

**ENTITY RELATIONSHIP MODEL**

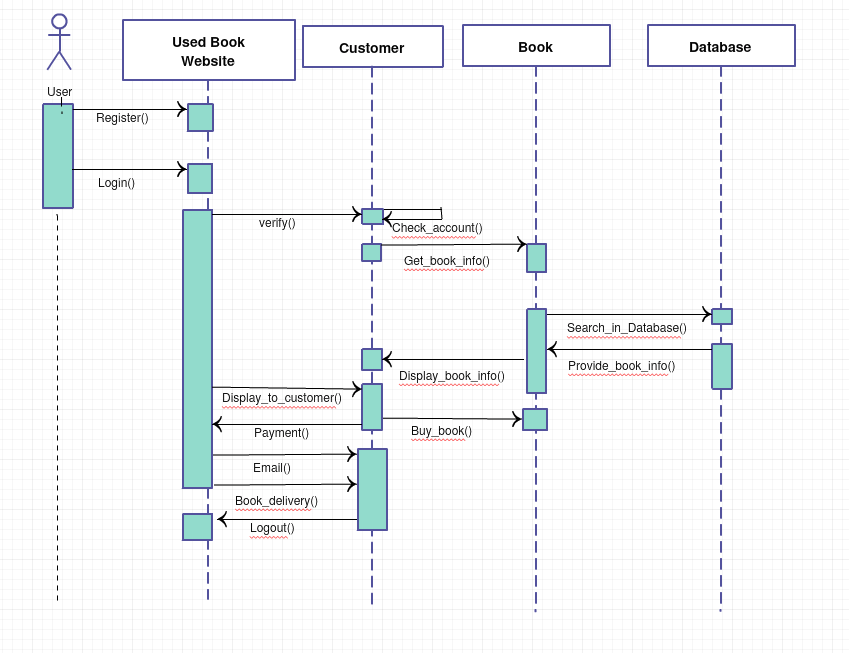
An **entity–relationship model** (**ER model** for short) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types

In software engineering, an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model, that defines a data or information structure which can be implemented in a database, typically a relational database

**SEQUENCE DIAGRAM**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

****

Since the sequence diagram for the admin portal shares login functionality with store-front. It does not make much sense to make a separate sequence diagram for admin portal. The above diagram shows a register method that enables the user to register in the store-front. Once registered he can login to the website to make purchases. Once he login, in the backend we verify his account. When we makes a search for a book we make a call to the database to fetch the books information. This information is brought back to the website and displayed to the user. Once he decides to buy a book, the cart operation managed within the website itself but when he makes a payment the details are updated in the database with a call to the backend. An email is send out to the customer as well. The main use of customer as a separate entity in the sequence diagram is to show when does the book reach the customer’s hand and when he gets confirmation about certain things in the whole process.

**TOOLS AND TECHNOLOGIES**

1. **Angular 5**

Angular is a platform that makes it easy to build applications with the web. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop

.



1. **TypeScript**

**TypeScript** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [programming language](https://en.wikipedia.org/wiki/Programming_language) developed and maintained by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is a strict syntactical [superset](https://en.wikipedia.org/wiki/Superset) of [JavaScript](https://en.wikipedia.org/wiki/JavaScript), and adds optional static typing to the language. TypeScript is designed for development of large applications and [transpile](https://en.wikipedia.org/wiki/Source-to-source_compiler) to JavaScript.[[9]](https://en.wikipedia.org/wiki/TypeScript#cite_note-9) As TypeScript is a superset of JavaScript, existing JavaScript programs are also valid TypeScript programs.

.



1. **Sublime**

**Sublime Text** is a proprietary cross-platform source code editor with a Python application programming interface(API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses



1. **Angular CLI**

The Angular CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices. It can generate components, routes, services and pipes with a simple command. The CLI will also create simple test shells for all of these. It has features to easily test your app locally while developing.



1. **JAVA 8**

**Java** is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA). Java 8 is the latest iteration of Java released by Oracle during the development of this project.



1. **Spring Technologies** (Spring Data + Spring Security + Spring Boot)

The **Spring Framework** is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform.

Spring Boot is Spring's convention-over-configuration solution for creating stand-alone, production-grade Spring-based Applications that you can "just run".

Spring's data access framework addresses common difficulties developers face when working with databases in applications. Support is provided for all popular data access frameworks in Java.

**Spring Security** is a Java/Java EE framework that provides authentication, authorization and other security features for enterprise applications.



1. **Hibernate**

**Hibernate ORM** (Hibernate in short) is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database. Hibernate handles object-relational impedance mismatch problems by replacing direct, persistent database accesses with high-level object handling functions.



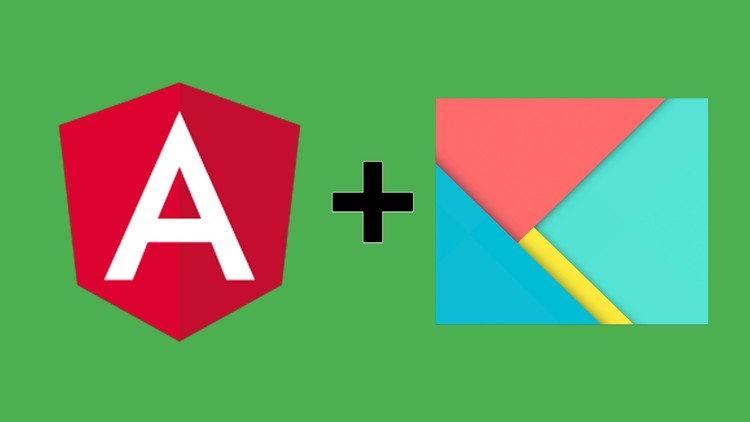
1. **MySQL**

**MySQL** is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.



1. **Angular Material**

**Material Design** (codenamed **Quantum Paper**) is a design language developed in 2014 by Google. Expanding upon the "card" motifs that debuted in Google Now, Material Design makes more liberal use of grid-based layouts, responsive animations and transitions, padding, and depth effects such as lighting and shadows. Material design integrated with Angular technology as a CSS theme is called Angular Material



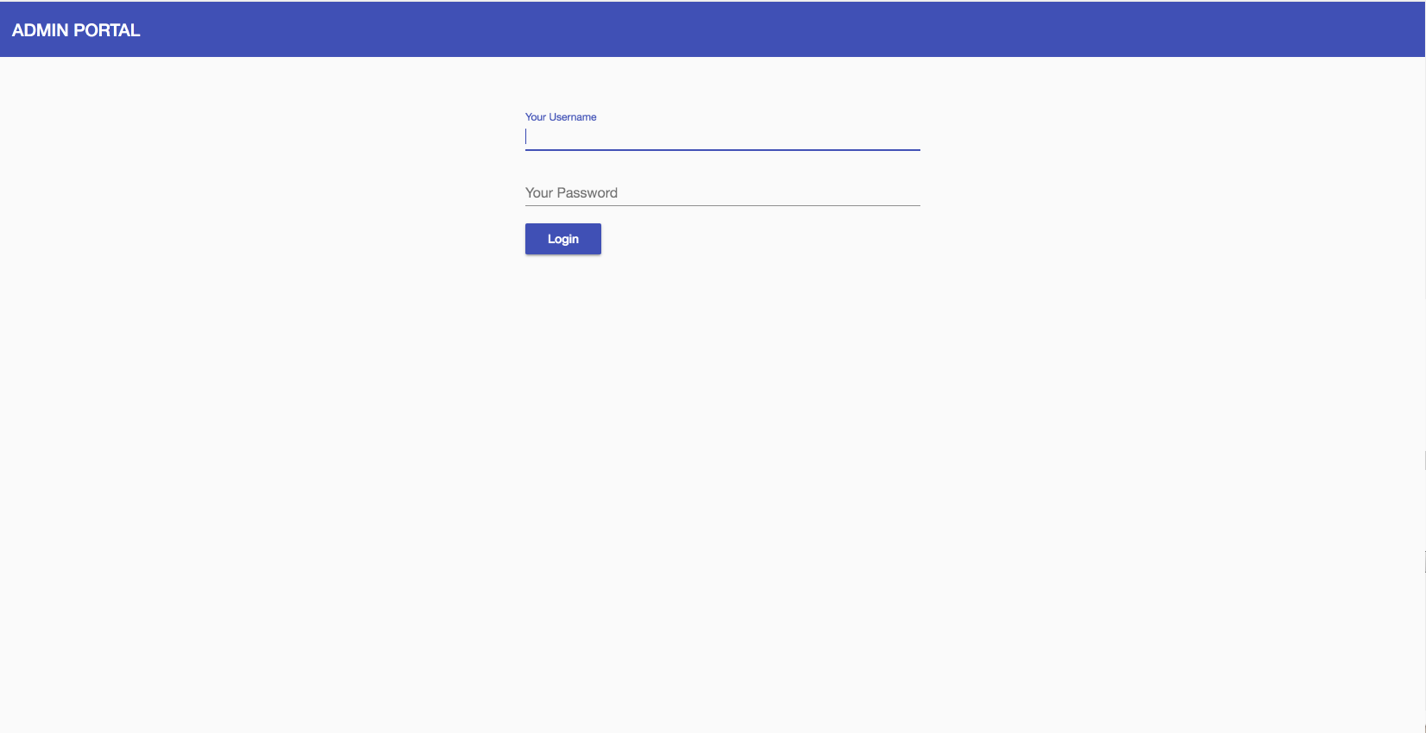
1. **Tomcat**

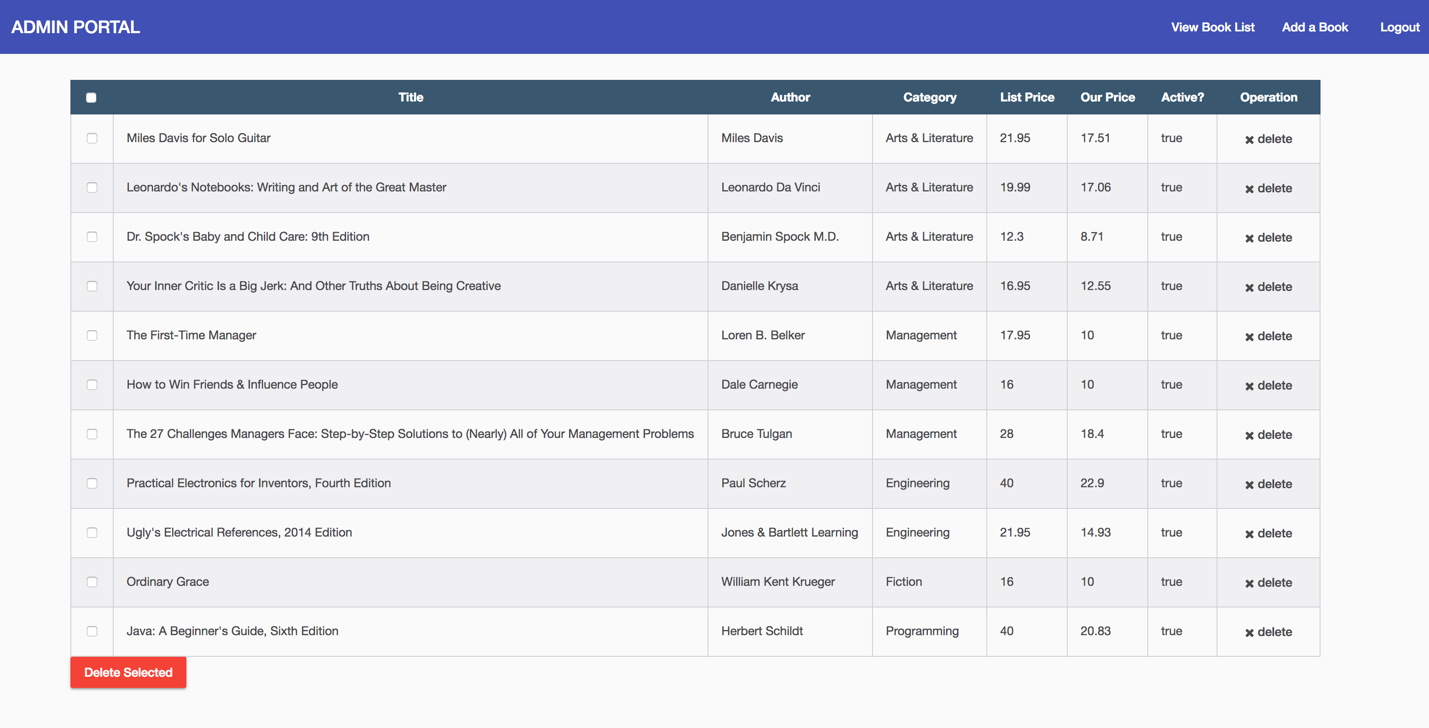
**Apache Tomcat**, often referred to as **Tomcat Server**, is an open-source Java Servlet Container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment in which Java code can run.

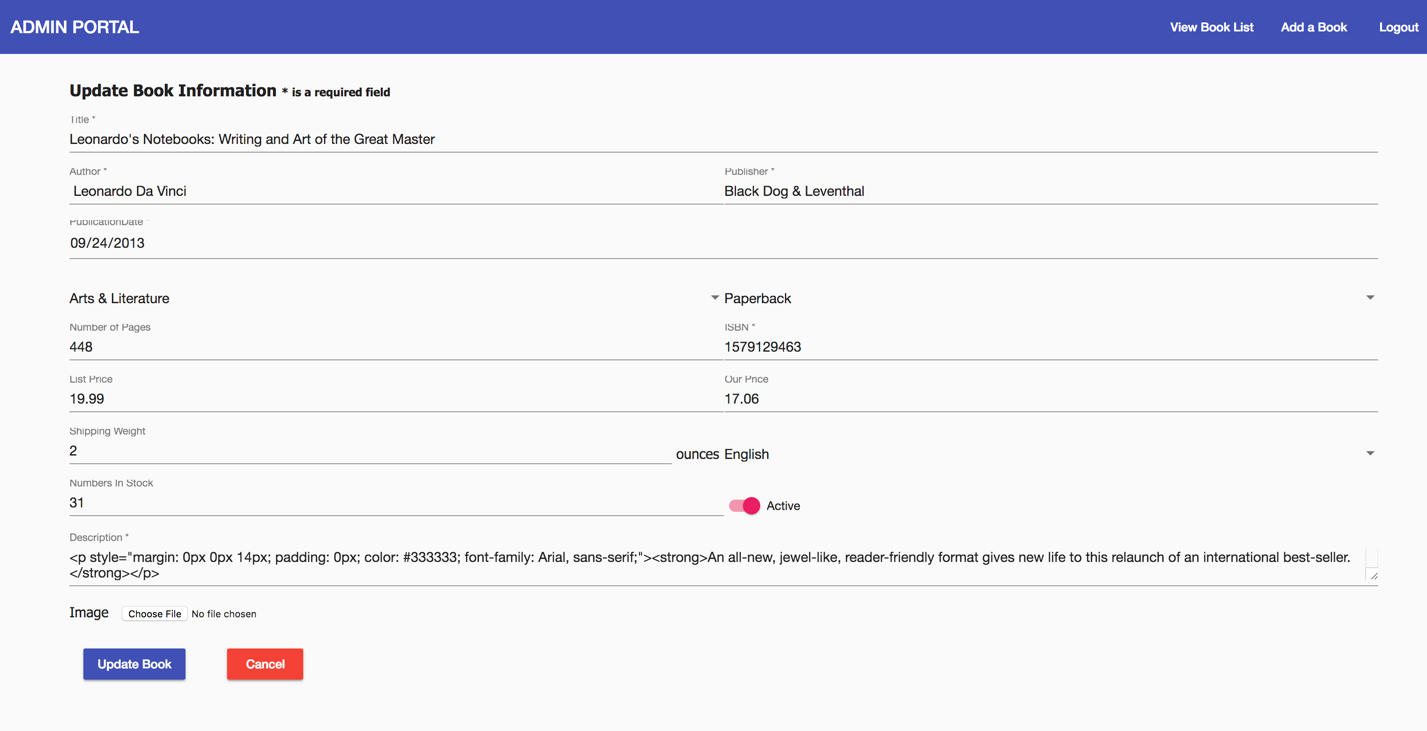


**RESULT**

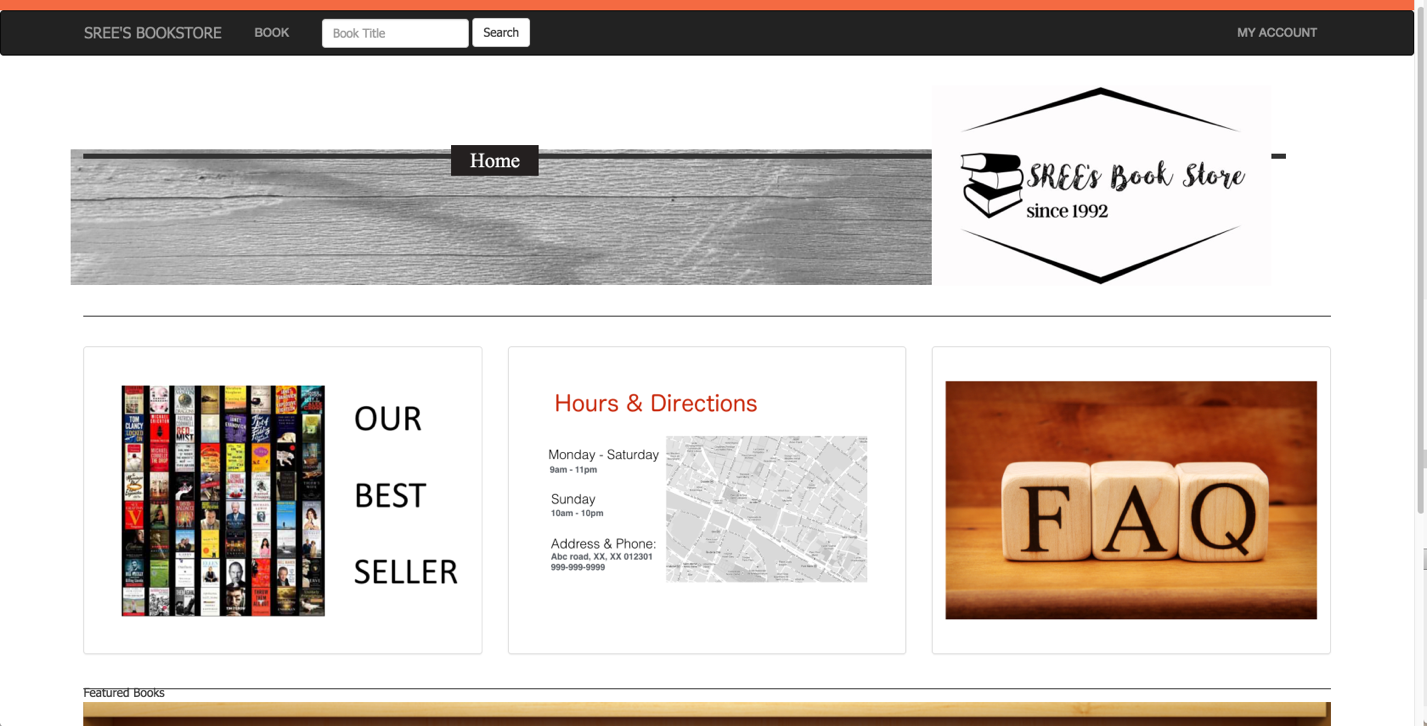
*ADMIN PORTAL*

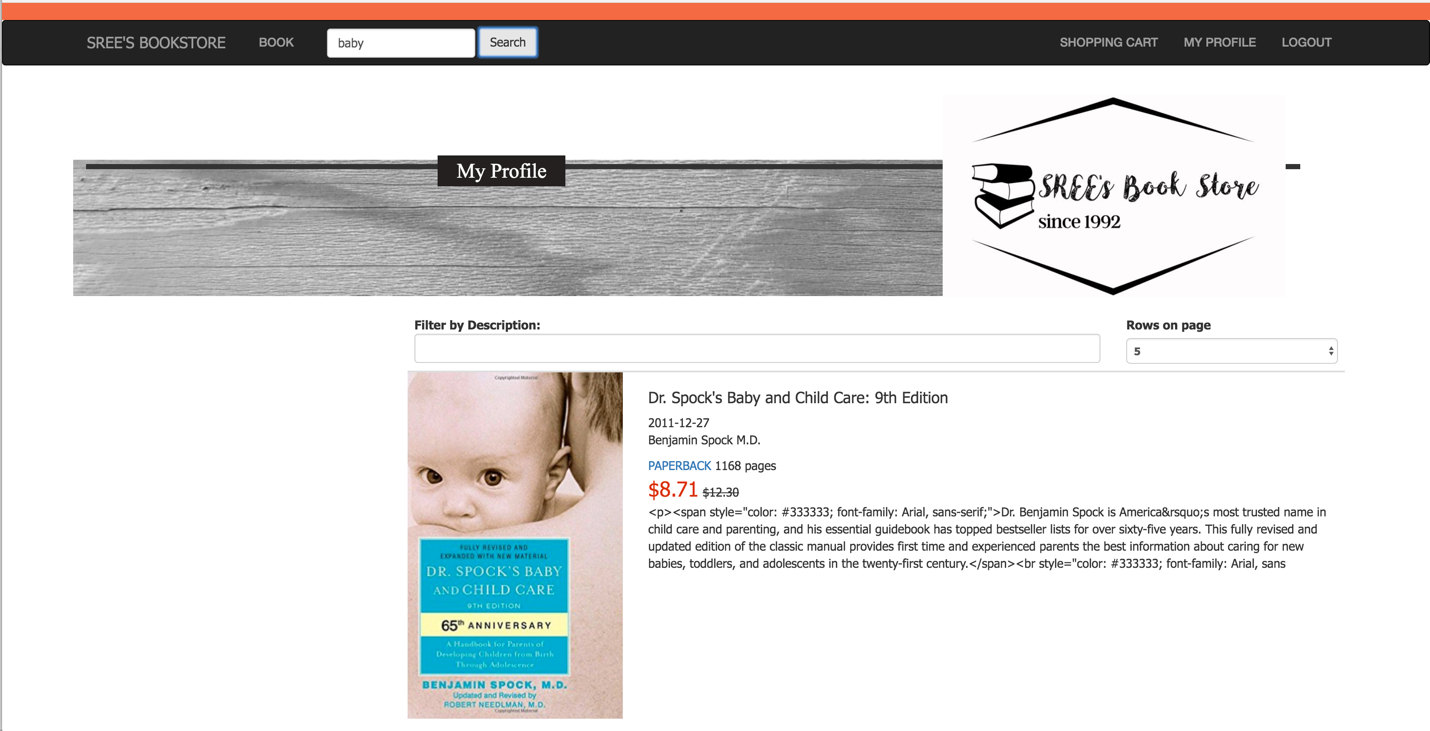
**

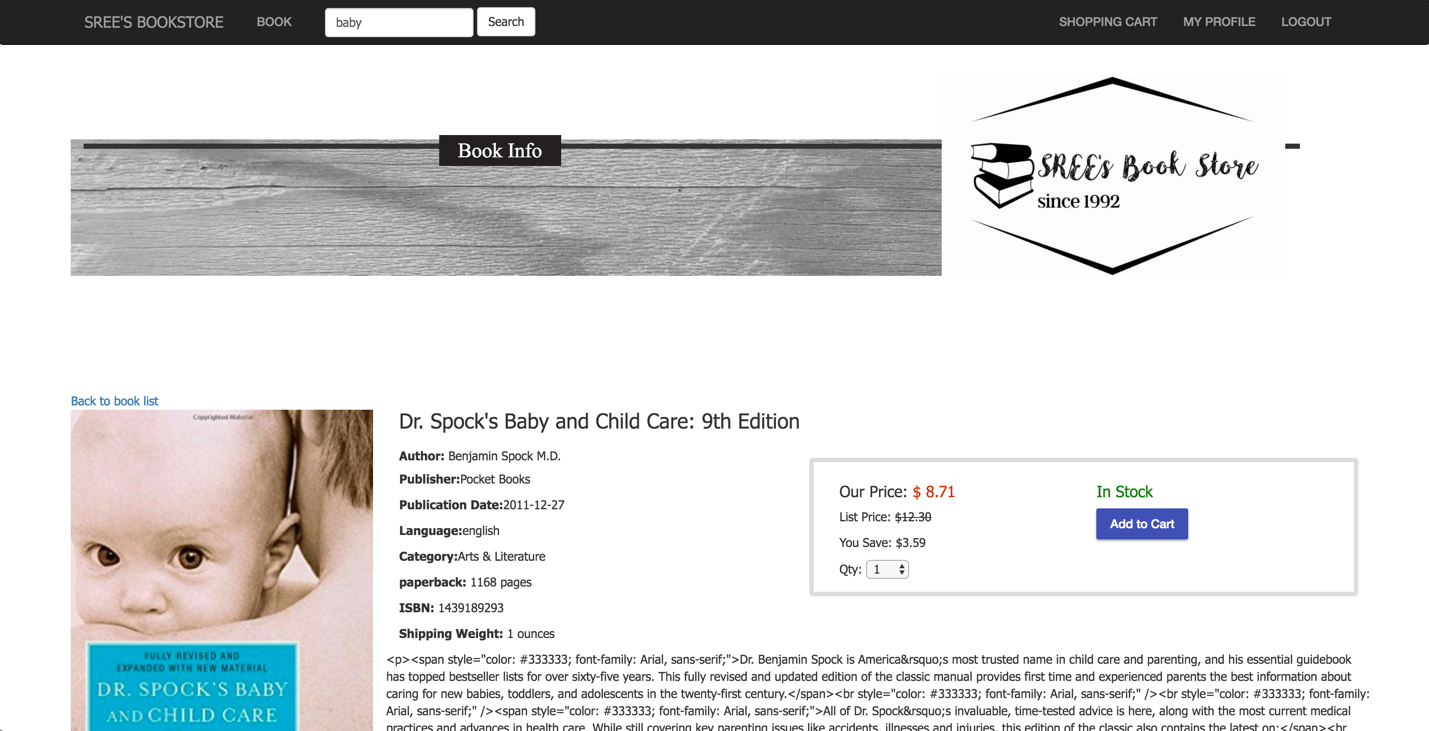
**

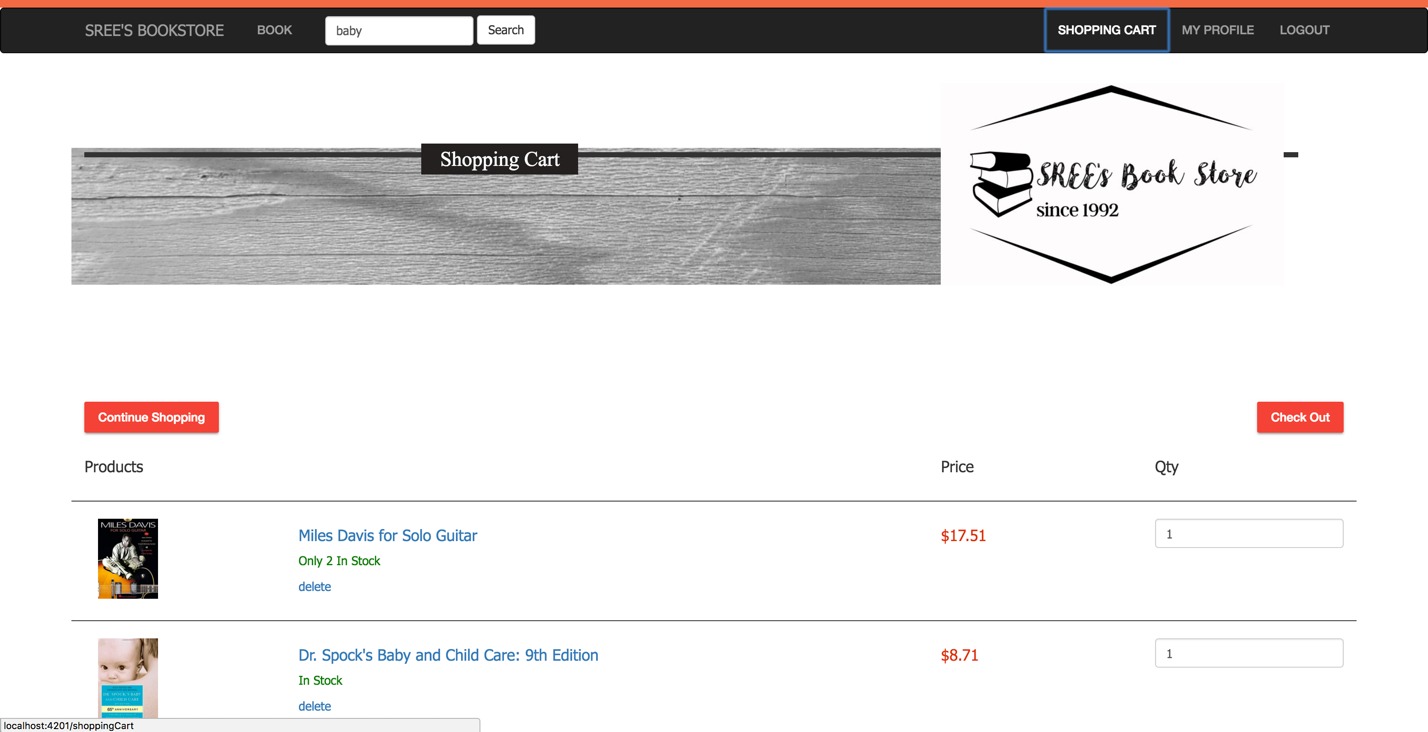
**

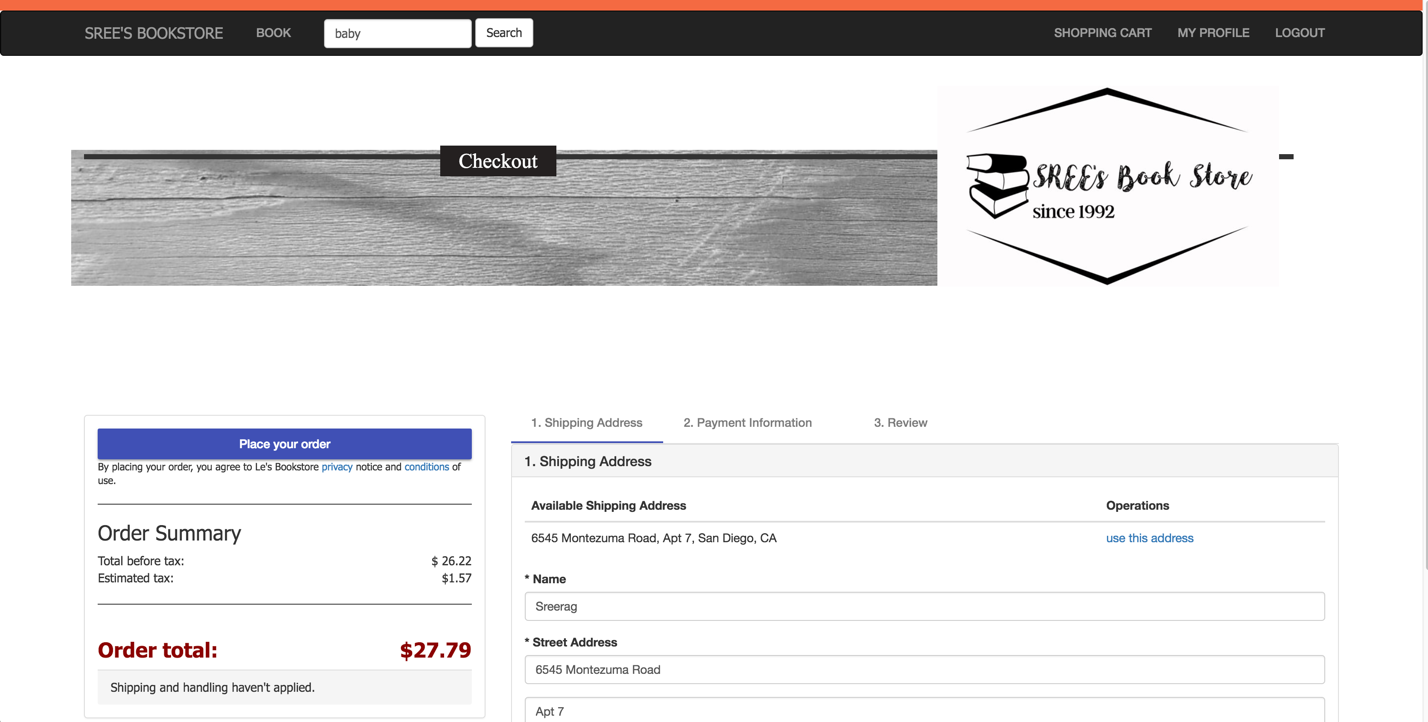
*STORE-FRONT*

**

**

**

****

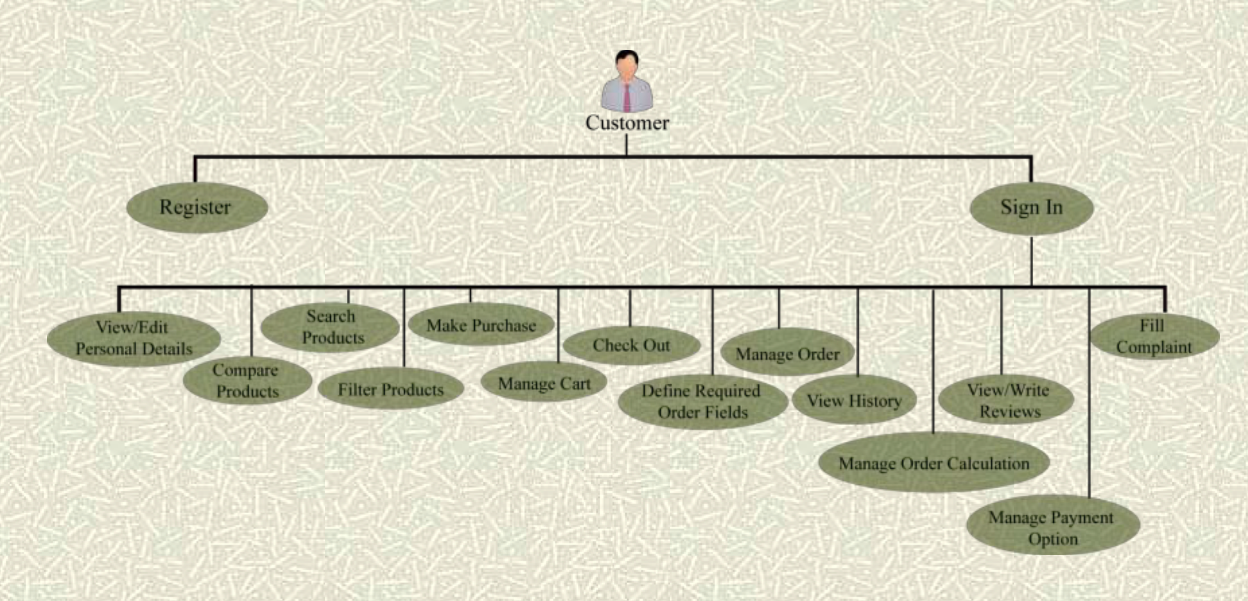
****

**Future Work**

The scope of this project is to build the first prototype, so obviously there are many ways to expand this project. Starting from the CSS, this project is to be handed over to IT team of Amrita University, they will port the theme of the project to their standard university theme. Functionalities like FAQ section and Featured book section are to be implemented by IT team.

Web scraping is another feature that is under consideration, rather than the library fixing the price for the book, when you update the book title, the condition and author information, The system will automatically check popular website and then fixes a price lesser than the actual price of the product published in these website. Since this is a feature that requires lot of effort, it’s been skipped from this prototype, The amrita university officials have also decided to implement this features only after considering the response from students for this initial version of the website.

Increasing the functionality of the store-front to the below use cases is also under discussion.

****

**References**