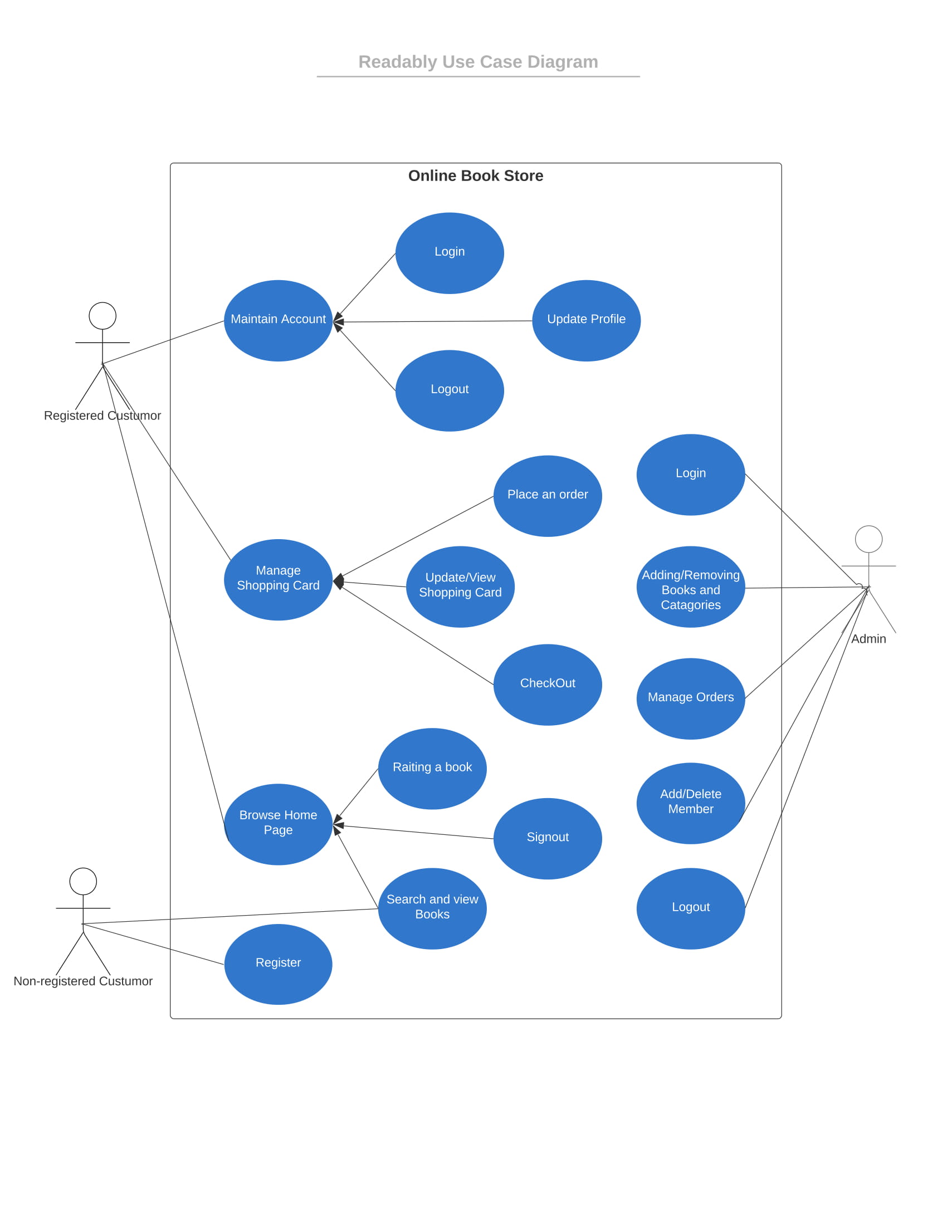
Readably

**System-Wide Requirements Specification**

# **Introduction**

A System-Wide Requirements Specification (SRS) is a document which sets out what the client expects and what is expected of the software system which is being developed. It is a mutual agreement and insurance policy between client and developer and is a vital part of the Software Development Lifecycle.

The SRS is a technical document which provides a framework for the software development process. It provides an overview of the software or application including what it should do and what its parameters are, how it will interact with its environment and the end users, and its hardware and software requirements



# **System-Wide Functional Requirements**

**2.1 Auditing**

* The data of new registered users are stored and this data can be controlled by admin. All transactions made by these users are stored and checked daily.

**2.2 Authentication**

* Database authentication is the verification process of the user trying to log in to a database. During the registration process, the user is expected to prefer an unique username. Users have limited authority in the system
* The authenticity of the user name and password match is checked for each user login.

**2.3 Reporting**

* When an unexpected event occurs, they are reported to the admin. Admin sees reports on monthly book sales and user information.

# **System Qualities**

## **Usability**

* Reably is designed to provide users with ease of use. For this, the following items are taken into account:
* Simplicity: There should be no unnecessary details in the page layouts. Thus, the user can perform the desired operation more easily.
* The books are categorized so that users can find books that suit their tastes more easily.
* All functions should not be visible in the application. Some functions are hidden in pop-up menus.
* The user can directly return to the home page from each page.
* The current page has the option to switch to all other pages.

## **Reliability**

* If users try to perform operations without permission, the system will give a warning.
* A backup system will be created for the database. This system allows data to be backed up. Thus, data loss is prevented in case of any problem.
* Tests and backups are made frequently so that if a problem is encountered, immediate action is taken.

## **Performance**

It is planned that the user can perform many operations such as book purchase registration, profile editing, payment processing in a maximum of 3 or 4 seconds. If this period is exceeded, we plan to perform complexity control.

## **Supportability**

* Upon completion of the necessary arrangements, upgrade versions come to ensure the development and sustainability of the system.
* Users can contact the support team via email. Incoming mail will be reported at regular intervals and reported to the development team.
* System requirements will be completed by the development team in line with the complaints received from users.
* Documentation will be created to inform users.

# **System Interfaces**

## **User Interfaces**

In this project, as a user interface, we used a template that is most suitable for our own wishes and that the user can use in the most comfortable way.

### *Look & Feel*

We chose this user interface so that users can reach their requests in the most comfortable way. Thus, we prevented complexity and difficulties. In this interface, the user can easily perform his requests. The user performs his operations on a site that looks good on his eyes.

### *Layout and Navigation Requirements*

The user encounters different interfaces in each process. There are other interfaces on the main page, while there are other interfaces on the registration page. Admin's and customers' home pages are different. In errors, different windows open.

### *Consistency*

In the design of our product, windows and tabs are grouped and named according to their content. Unexpected situations are avoided by enabling the user to access predictable windows in the operations he wants to do. Placing similar operation buttons in the same places makes it easier for users to guess and adapt. This makes it easier for the user to learn and love the product.

### *User Personalization & Customization Requirements*

The system interface will give users different opinions for the admin, registered and non-registered users. When the user enters the system, they will be guided according to their actions. But when the admin enters the system, it will be directed to the places where it can perform different operations because its transaction limit is wider and different. In short, creating different users will change the content displayed.

## **Interfaces to External Systems or Devices**

### *Software Interfaces*

The back end of the project is developed with java. Maven project is created and it starts to develop our project. MySQL is used in the database section. Spring Boot is used as frameworks. Hibernate and Spring Data JPA are used to connect to the database. Thus, the project provides connection with the database. The security side of our project is created with Spring Security. RestFul is used for state transfers in the HTTP protocol in the project. Template is used on the front side and necessary improvements are made on the template using Thymeleaf, HTML, CSS. Also, Javascript is used if necessary. We add the plugins we will use with the help of dependencies to our project.

### *Hardware Interfaces*

There won’t be a hardware interface associated with project

### *Communications Interfaces*

### There will be no communication interface associated with the project. To improve portability, the software will be offered as a web application in the future.

# **Business Rules**

## Business rules regarding product and order

### D*iscount status in the campaign*

If a user purchases 300 TL or more at a time, the system will automatically discount 20 TL

*5.1.2 User favorites*

When the user adds a product to their favorites, they will be notified when there is a change in the status of that book

*5.1.3 Book limit*

If users try to buy more than 5 books at a time, the system will not allow it.

*5.1.4 Time limit*

If the user exceeds the time set during the book purchase, the system will not allow the transaction to take place

# **System Constraints**

* This application is a web application.
* Application will be developed with Java programming language.
* MySQL will be used for the database management system.
* Spring Boot will be used as the framework in this application.
* Hibernate and Spring Data JPA will be used for the database connection
* Thymeleaf, HTML, CSS will be used for front-end.

# **System Compliance**

## **Licensing Requirements**

The source code can be reached by anyone.

## **Legal, Copyright, and Other Notices**

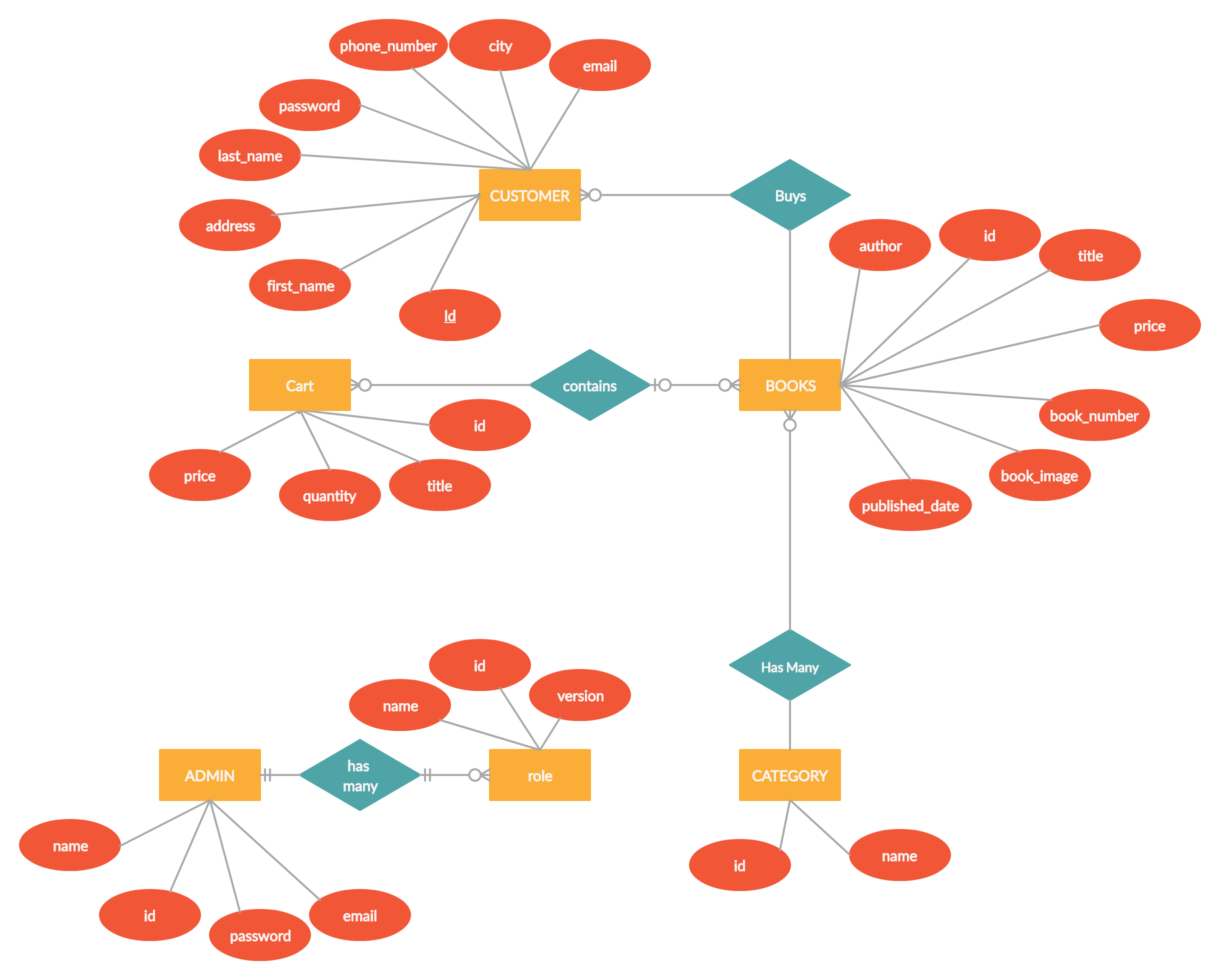
The source code can be shared, copied and modified, provided that it is not commercial.

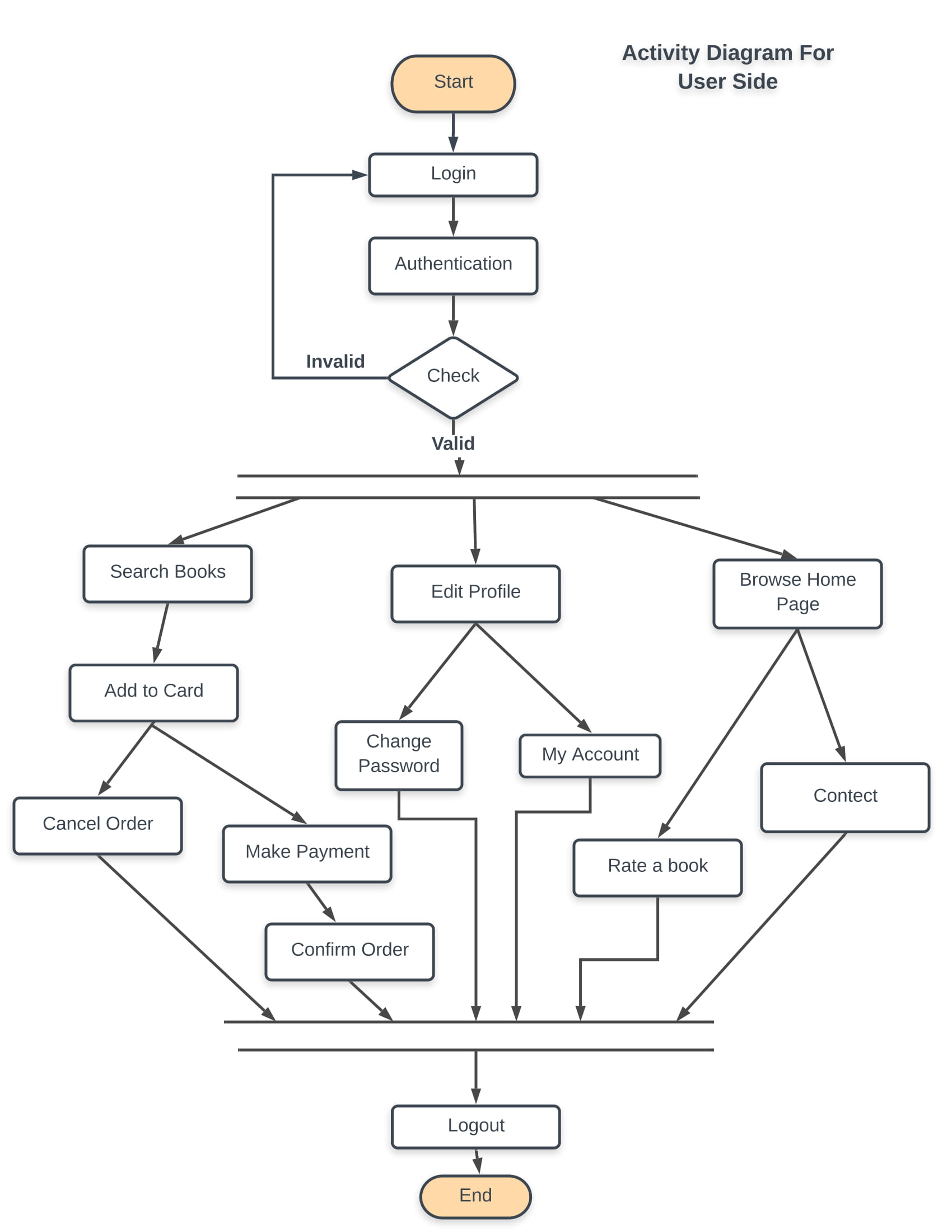
## **Applicable Standards**

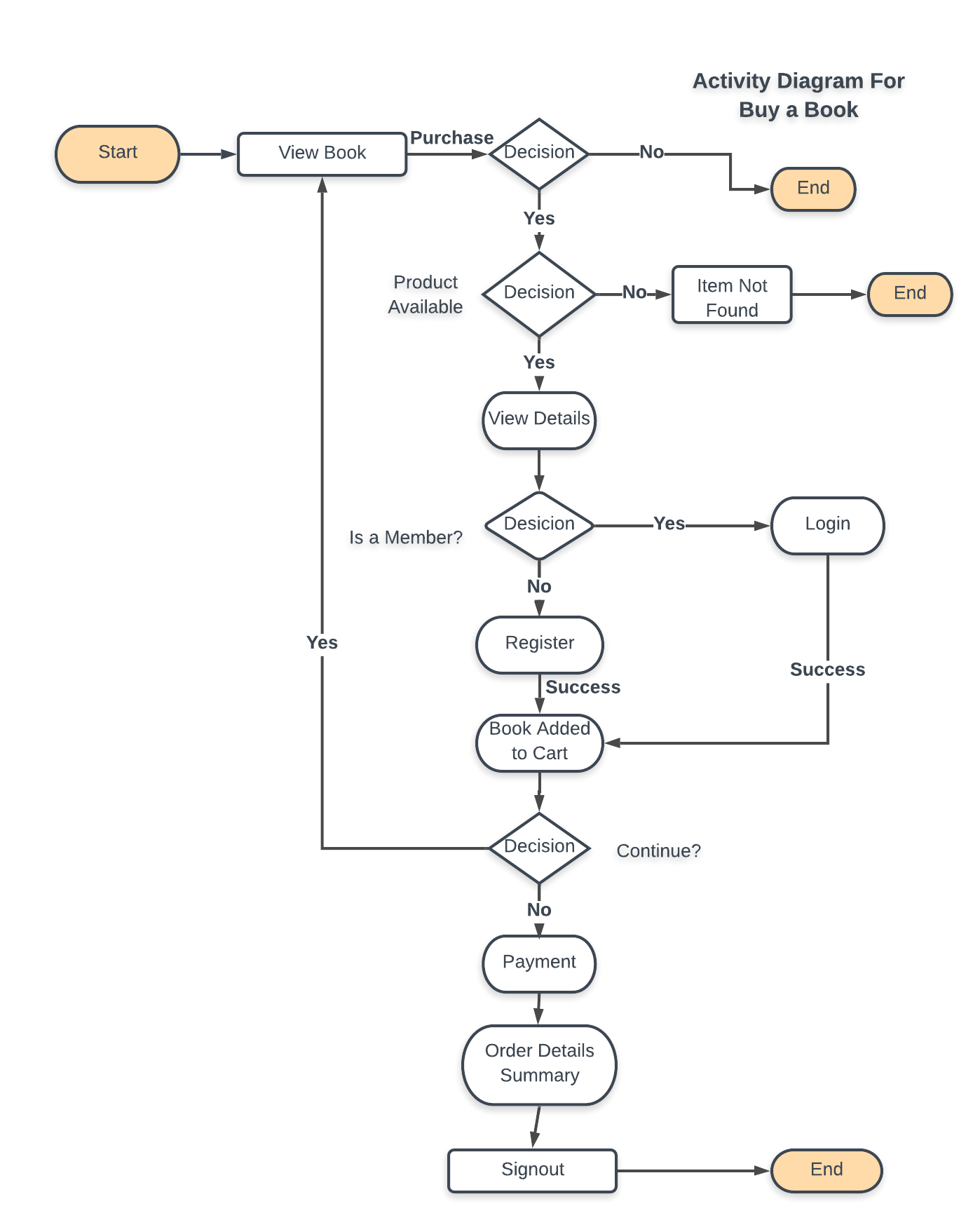
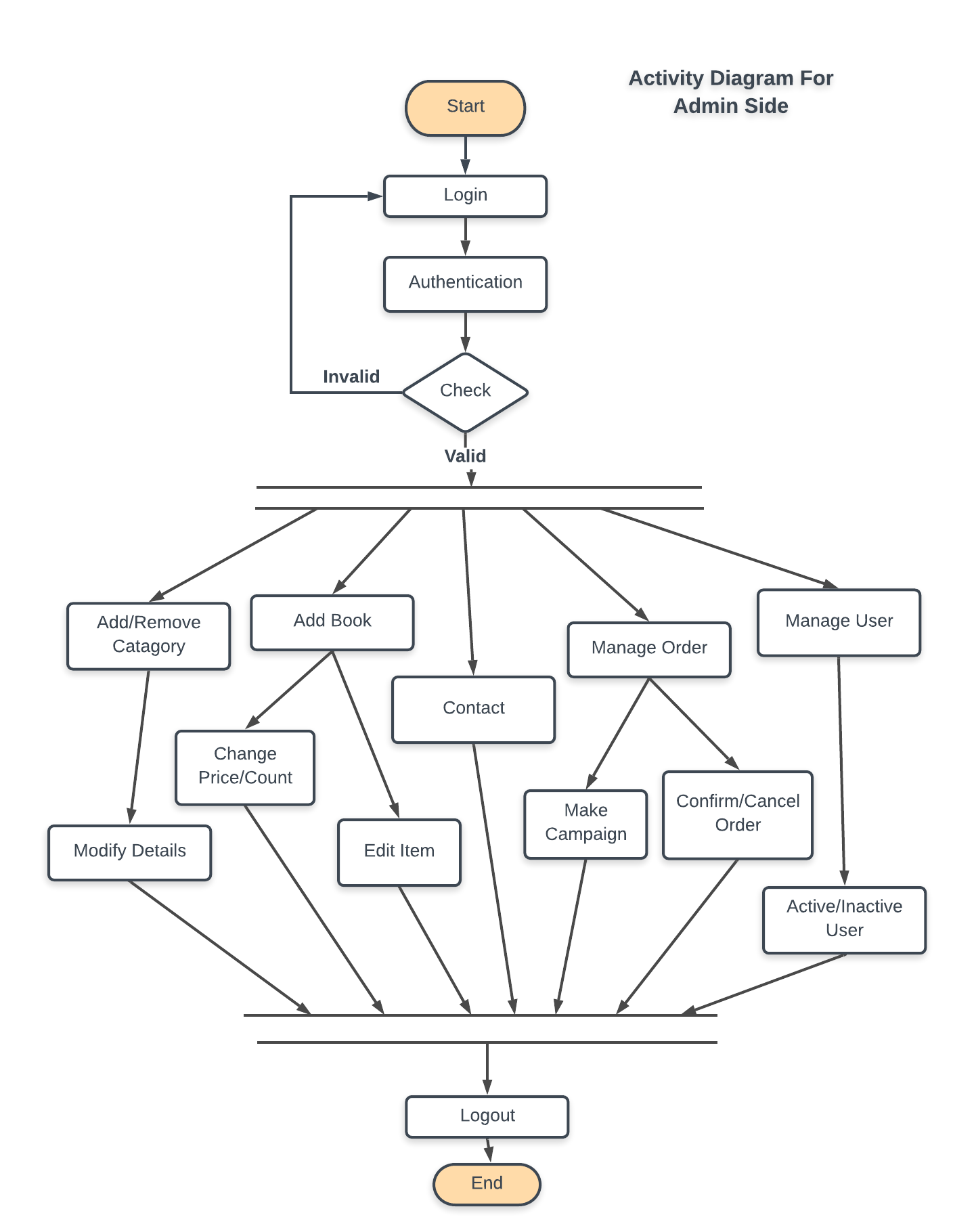
* OpenUp documents are used.
* IEEE standards have been taken into account.

# **System Documentation**

**Entity Relationship Diagram (ER Diagram)**





**

