

COE 528 Final Project Report

Use Case Description:

A use case is an abstraction that describes all the following scenarios based on the functionality of the situation. In this case, the use case diagram describes the relationships of the users and the Book Store application.

Name: Book Store

Participating Actors:

The two actors participating in this use case would be the customer and the owner.

Entry condition:

- The use case starts when the owner logs onto the bookstore app in order to add or remove customers and/or books
- The use case can also start when the customer logs onto the bookstore app in order to purchase a book

Flow of events:

- Once the customer logs in, the second use case would be to buy a book. Then the use case after that would be to either redeem your points or to buy it with regular money
- Once the owner logs in, they can either add/remove customers or books. They can add a customer with their username and password and add books with their name and price. Once this is fulfilled the final list will be displayed

Exit conditions:

- The use case terminates once a customer purchases a book, the transaction goes through and they log off They will see a display indicating how many points they have and what their status is (either Gold or Silver).
- The use case will also terminate once the owner adds or removes a book or customer and the final list of it will be displayed and they log off.

Exceptions:

One exception can occur when the customer or the owner inputs the wrong password or username. In this case an error will appear to reenter them.

Special Requests:

Only the indicated customers that have been added by the owner can be logged onto the bookstore app. Similarly, only the books that have been added by the owner can only be bought by the customers.

Rationale behind using the State Design Pattern:

The objective of a State Diagram is to allow an object to change its behaviour and show those changes in its class when the internal state itself is altered. In this project, the state diagram had to be implemented in order to establish the different statuses of the customers, which is altered based on their transactions. The state of the object (the status) is given a condition (either silver or gold) based on the values of their instance variable(s). In this case, the instance variable would be 'int points' as the number of points the customer has indicates their status. This project follows a structure where the behaviour of the classes is caused by external stimuli.