

DATA STRUCTURES AND OBJECT ORIENTED PROGRAMMING

# E-commerce product review system

Edric Tran

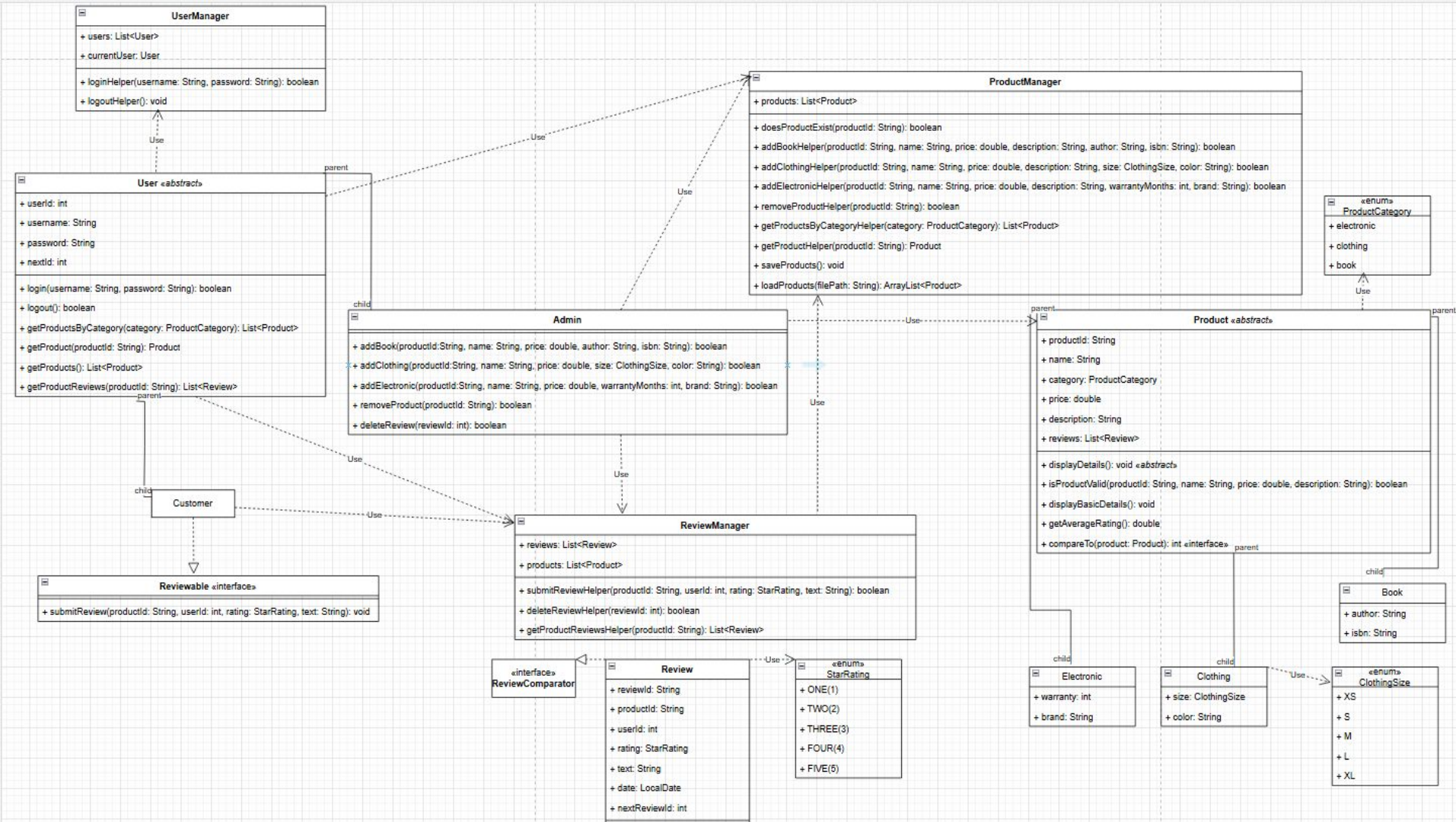
# Outline

---

- |    |                                |                       |
|----|--------------------------------|-----------------------|
| 01 | <b>Project description</b>     | Slide 03              |
| 02 | <b>UML class diagram</b>       | Slide 04              |
| 03 | <b>Program features</b>        | Slide 05              |
| 04 | <b>Program implementations</b> | Slide 06              |
| 05 | <b>Input and output code</b>   | Slide 07              |
| 06 | <b>Challenges</b>              | Slide 08              |
| 07 | <b>Learning outcomes</b>       | Slide 09 <sub>2</sub> |

### Description

This project simulates an online product review platform(backend) where users can browse different products. Products can belong to various categories (e.g., electronics, books, clothing), and customers can submit reviews with ratings. Admin users can manage the product list and moderate user-submitted reviews



# Program features

## User (superclass)

- Login
- Logout
- Search for a list of products of a specific category
- Search for a specific product
- View all the available products
- View all the reviews of a product

## Admin (subclass)

- Add products
- Remove products
- Delete reviews

## Customer (subclass)

- Submit reviews

# Program implementations

## Hierarchies & interface

Parent class #1: User  
Subclasses: Admin & Customer

Parent class #2: Product  
Subclasses: Book, Clothing & Electronic

Interface: Reviewable  
Abstract method: submitReview()

## Runtime-polymorphism & TextIO

Runtime-polymorphism method:  
Product.displayBasicDetails()

TextIO I  
loadProducts()

TextIO O  
saveProducts()

## Comparable, comparator & unit testing

Comparable  
Compare products by their name  
alphabetically

Comparator  
Compares review by:  
-date (newest to oldest)  
-rating (highest to lowest)  
-default (length of text, longest to  
shortest length)

Unit testing for every non-void  
method (general case)

# Input and output code

## Input

```
Admin admin = new Admin( username: "Mark123", password: "12345*");
admin.addBook( productId: "B1", name: "Harry Potter", price: 30, description: "Once upon a time...",
author: "J. K. Rowling", isbn: "978-0-306-40615-7");
```

```
Customer customer = new Customer( username: "Edric321", password: "54321");
customer.submitReview( productId: "B1", userId: 2, Review.StarRating.FIVE, text: "SO GOOD");
```

```
customer.submitReview( productId: "B1", userId: 2, Review.StarRating.FIVE, text: "WAY TOO GOOD");
```

```
admin.deleteReview( reviewId: 1);
```

```
admin.removeProduct( productId: "B1");
```

```
customer.submitReview( productId: "B1", userId: 2, Review.StarRating.FIVE, text: "WAY TOO GOOD");
```

```
admin.removeProduct( productId: "B1");
```

## Output

Product has been successfully added

Review submitted

Customer already reviewed this product

Review removed

Product has been successfully removed

Couldn't find product with id: B1

Product id not found

# Challenges



## Challenge #1

For the deliverable one, it was difficult to come up with the methods for the project from scratch.

## Challenge #2

At first, I had a lot of trouble using Git Bash and all the commands that comes with it.

## Challenge #3

While writing the code, it was challenging to try to connect the classes together and all its methods.



# Learning outcomes



## Gain #1

While doing this project, I learned how to use GitHub and Git Bash properly.

## Gain #2

I learned how to do the TextIO I (this is the first time I got to apply the input TextIO part in a code).

## Gain #3

In the end, I learned how to code a coding project without a given prompt and all the required methods (unlike semester 1 project where all the instructions for the methods were given).