Large Scale and Multi-Structured Databases
University of Pisa
Academic Year 2022-2023

Application: E-Commerce System

Workflow Analysis

Eng. José Corcuera, Prof. Pietro Ducange

Assignment #2

Given the Requirements and the UML diagrams provided below, carry out the following activities:

i) Complete the data modeling stage providing the design of the database: consider relational database for most of the entities and Key-Value database for the cart. To this aim, define tables (towards 3rd normal form) and the structure of the keys for Key-Value database.
ii) Implement the main JAVA classes (fields and methods) which describe the entities and their relationships as described in the UML Analysis Class Diagrams. To this aim, additional fields, methods and even classes may be defined in the code.
iii) Implement the JAVA classes for the interaction with the databases.
iv) Design and implement a simple User Interface for testing the application, namely for allowing users to exploit the functionalities offered by the application.
v) Populate the database with synthetic data randomly generated. Suppose to consider
   a. 1 company with 1 manager,
   b. 10 brands
   c. 100 products for each brand
   d. 1000 users
   e. Randomly, generate from 1 to 20 orders for each user. Consider that each order should contain from 1 to 10 products randomly selected.

For this exercise, non-functional requirements may be relaxed.

a- Functional requirements:

Functional requirements - Unregistered user:

1. The system must allow unregistered users to browse products (a simplified view of each product will be available for the user).
2. The system must allow unregistered users to display the information of a selected product. This information must include the type, name, description, images, price, other attributes defined by each type and reviews.
3. The system must allow an unregistered user to become a customer by registering his/her full name, address, phone number, email address and password.
4. The system must allow an unregistered user to login as a registered user.

Functional requirements - Customer:

1. The system must allow a customer to change his/her account information including their password.

2. The system must allow a customer to browse products (a simplified view of each product will be available for the user).
3. The system must allow a customer to find products (by name) (a simplified view of each product will be available for the user).
4. The system must allow a customer to display the information of a selected product. This information must include the type, name, description, images, price, other attributes defined by each type and reviews.
5. The system must allow a customer to add products to a cart while he/she is viewing a product.
6. The system must allow a customer to see the content of the cart.
7. The system must allow a customer to update his/her cart (to modify product quantity or to delete a product)
8. The system must allow a customer to start the checkout process.
9. The system must ask for a customer to insert an address and the billing method during the checkout process.
10. The system must allow a customer to cancel an in progress checkout process.
11. The system must allow a registered user to display the result of the checkout process.
12. The system must update the quantity of product in stock after each successful checkout process and clear the cart.
13. The system must allow a customer to see the progress of his/her order.
14. The system must allow a customer to see all his/her orders.

Functional requirements - Company manager:

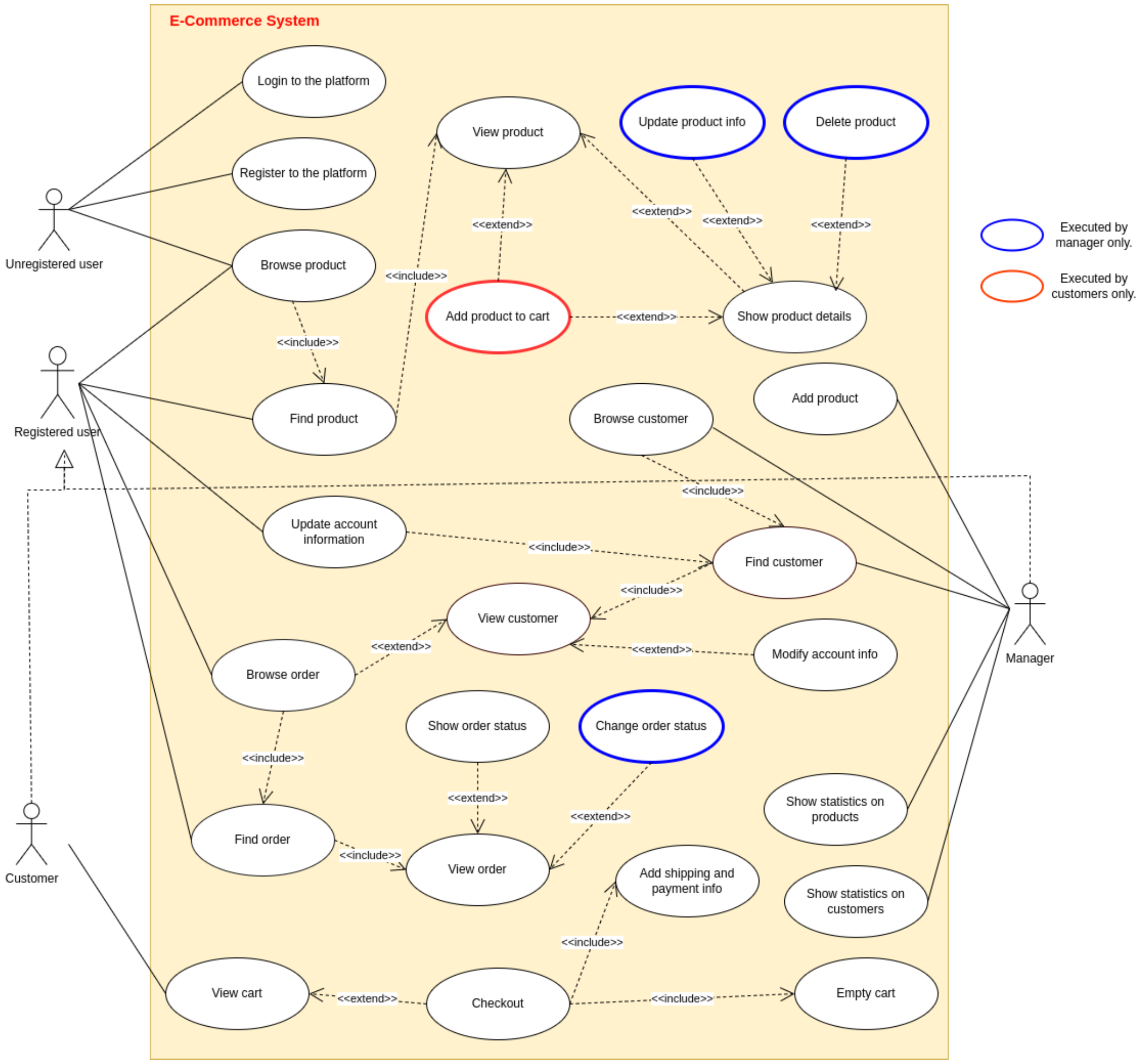
1. The system must allow a manager of a company to change his/her account information including his/her password.
2. The system must allow a manager of a company to browse all available products.
3. The system must allow a manager to display the information of a selected product. This information must include the type, name, description, images, price, other attributes defined by each type and reviews.
4. The system must allow a manager of a company to add new products by specifying the type, name, description, images, price and other attributes defined by each type.
5. The system must allow a manager of a company to update product information.
6. The system must allow a manager of a company to remove products.
7. The system must allow a manager to browse customers.
8. The system must allow a manager to browse orders of a selected customer (a simplified view of each order will be available for the user).
9. The system must allow a manager to find orders (by number, creation date range) (a simplified view of each order will be available for the user, this includes: customer name, products, quantities and shipping address).
10. The system must allow a manager of a company to change the status of an order (Paid->In Preparation->Shipped->Delivered).
11. The system must offer a manager of a company analytics about products namely:
    a. To show the most sold products by category (in a time interval)
    b. To show the less sold products by category (in a time interval)
12. The system must offer a manager of a company analytics about customers namely:

   a. To show the most active users (in terms of number of delivered orders in a time interval)
   b. To show users prone to spend more than others (i.e. rank of the users in terms of money spent in a time interval).

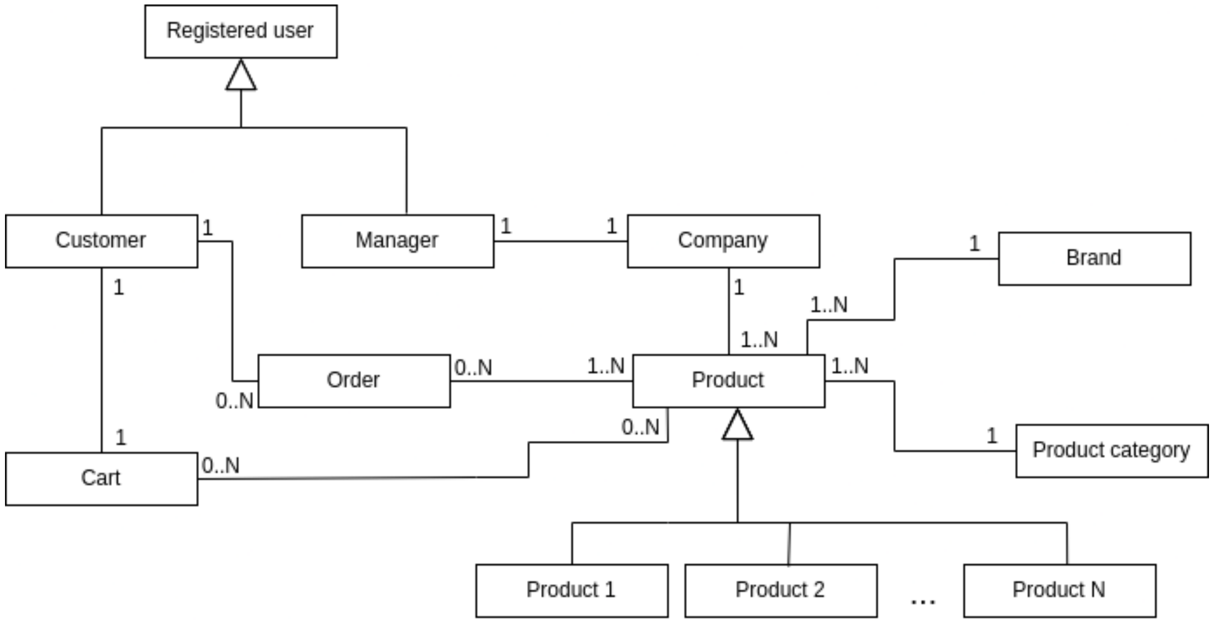
Non-functional requirements:

1. The system must be a responsive website application.
2. The system must encrypt users' passwords.
3. The system must communicate with any user by using a secure communication channel (HTTPS).
4. The system will ensure a throughput of at least 50 orders per second.
5. The system will be available 24 * 7.
6. The system must be developed by using an OOP language (i.e.: Java, Python, etc.)
7. The system will be deployed under a serverless computing system.
8. The system will be in compliance with the microservices architecture.

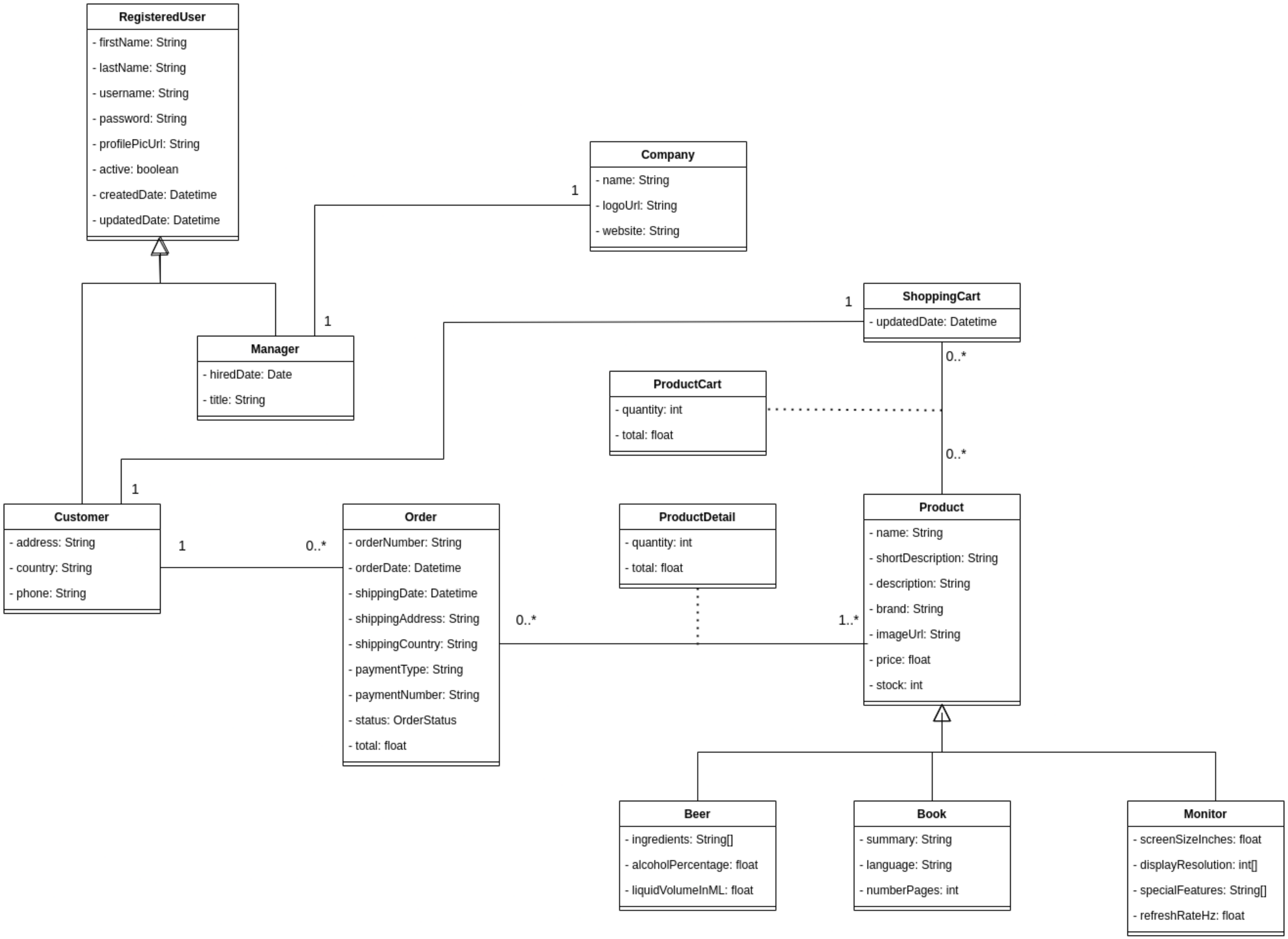
B. UML Use case diagram



C. UML Analysis Class (simplified)



D. UML Class Diagram


RegisteredUser
- firstName: String
- lastName: String
- username: String
- password: String
- profilePicUrl: String
- active: boolean
- createdDate: Datetime
- updatedDate: Datetime
Company
- name: String
- logoUrl: String
- website: String
1
1
Manager
- hiredDate: Date
- title: String
ShoppingCart
- updatedDate: Datetime
1
0..*
ProductCart
- quantity: int
- total: float
0..*
1
Customer
- address: String
- country: String
- phone: String
1
0..*
Order
- orderNumber: String
- orderDate: Datetime
- shippingDate: Datetime
- shippingAddress: String
- shippingCountry: String
- paymentType: String
- paymentNumber: String
- status: OrderStatus
- total: float
ProductDetail
- quantity: int
- total: float
0..*
1..*
Product
- name: String
- shortDescription: String
- description: String
- brand: String
- imageUrl: String
- price: float
- stock: int
Beer
- ingredients: String[]
- alcoholPercentage: float
- liquidVolumeInML: float
Book
- summary: String
- language: String
- numberPages: int
Monitor
- screenSizeInches: float
- displayResolution: int[]
- specialFeatures: String[]
- refreshRateHz: float