Scalable and Distributed Computing: Management system for Trung Nguyen Coffee Company

Pham Hoang Minh

Phan Quoc Khoi

1. Project requirement

Distributed system has become a trend in the industry. Most of the major corporation such as Google and Facebook use distributed server in order to handle massive number of requests around the world. In this project, we are required to design a system for a corporation named Trung Nguyen. The main products that the company provides are coffee and foods. After a sequence of discussion, a project requirement is provided in [1] and the output system should be 3-tier.

2. Methodology

Before going any further to the design details, we want to provide some assumptions:

(i) All activities related to buying and selling a product will take place offline. This assumption is reasonable since if it happens online, the system should be capable of choosing branch nearest to the customer location, which is inaccessible within the budget of implementation.

(ii) All discount or promotional programs focus solely on reducing prices of one or more products during a certain amount of time. The aim of the assumption is to simplify the system design so that both employees in business and IT can understand.

(iii) If a product has a discount applied by general manager (global discount), a product will not receive any discount from local manager (local discount) until the global discount expires. If a product already has a local discount, a general manager can overwrite that local discount into a global discount (and the local discount will no longer be valid).

2.1. Use case diagram

Before implementation, we have to model the real system. The initial step that needs to be taken is to identify people participating in the system and their incentives. From [1], we find out that there are four users participating in the system: general managers, local managers, sales staffs, and customers. [1] also clearly states the incentives of each user. A customer can only see the product information (and cannot buy products due to our assumption). A sales staffs can only record transactions and provide bills. A local manager can generate a report on the shop's sales and apply local discount. A general manager can get sales report from any branch and can apply global discount. Local and global discount mechanisms should follow the assumptions. All of mentioned information can be presented by the Use Case Diagram (See Figure 1). Details on each use case is mentioned in Appendix A.

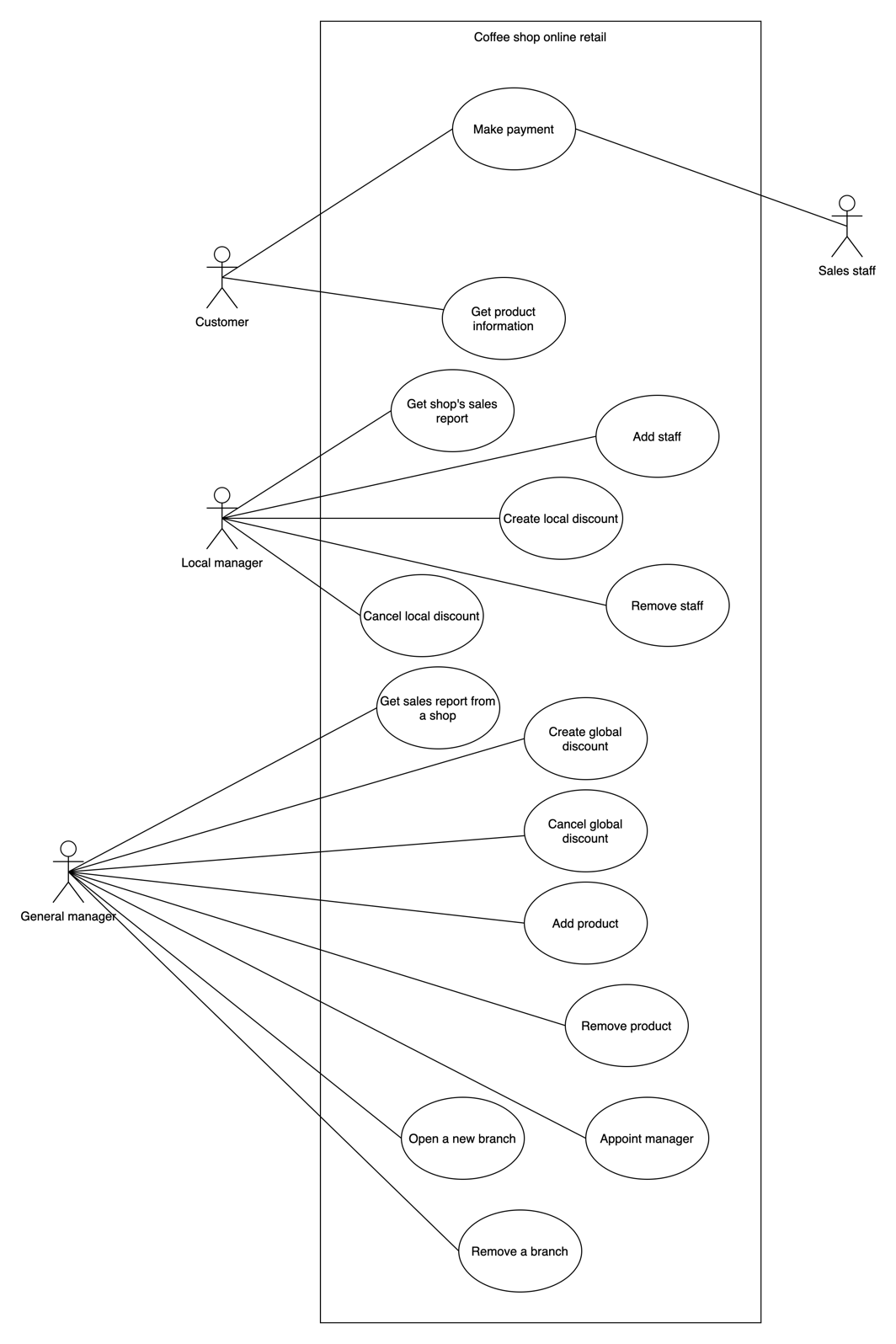
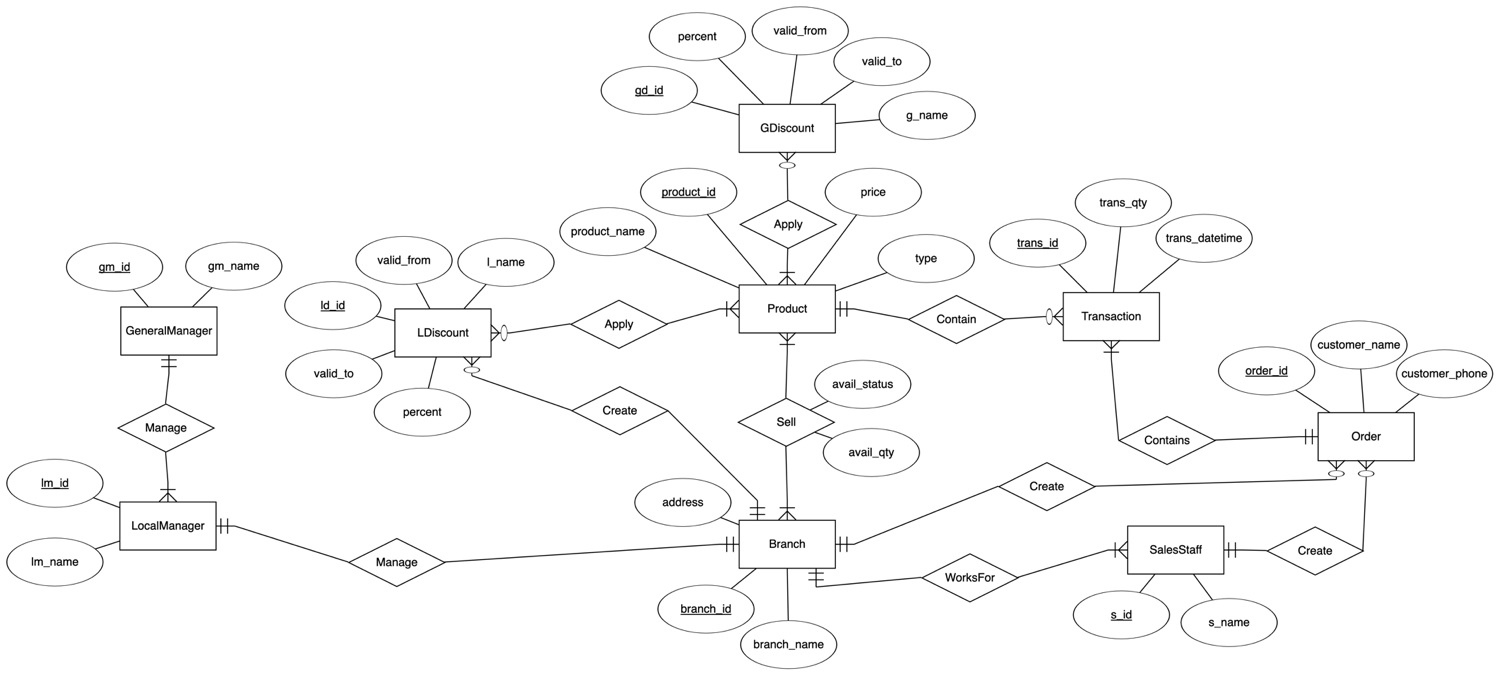


Figure 1: System use case diagram

2.2. Entity-relationship diagram (ERD):



APPENDIX

A. Use case description

|  |  |  |  |
| --- | --- | --- | --- |
| UC | Use case name | Use case info | Flows |
| 1 | Make payment | Triggers/Goals: Customer wants to pay for one or more products at a specific Trung Nguyen shop  Actors: Customer, sales staff | Main flow:  1. Customer brings products to the counter  2. Sales staff will scan for the product’s ID  3. System returns product details  4. Sales staff takes customer’s name, customer’s phone number and enter it to the form  5. Sales staff tells customer the total price  6. Customer gives money and sales staff returns change (if any)  7. Sales staff submits the form  8. System saves the details and notify the transaction is processed successfully  9. Sales staff gives customer bill  Exception:  8a. Transaction is failed to process  1. Sales staff uses paper to record the bill and give it to the customer  2. Sales staff immediately contact the local manager |
| 2 | Get product information | Triggers/Goals: Customer wants to search for Trung Nguyen product’s information  Actors: Customer | Main flow:  1. Customer accesses to Trung Nguyen’s web page  2. System returns a list of products  Exception:  2a. Server has internal error  1. System terminates |
| 3 | Get shop’s sales report | Triggers/Goals: Local shop’s manager wants to have sales report of his own shop  Actors: Local manager | Main flow:  1. Local manager chooses which kind of report to be generated  -- if the local manager chooses weekly report, do step 2 --  2. System returns the total price that the shop earns, and the product with its total sales this week.  -- if the local manager chooses monthly report, do step 3 --  3. System returns the total price that the shop earns, and the product with its total sales this month.  -- if the local manager chooses yearly report, do step 4 --  4. System returns the total price that the shop earns, and the product with its total sales this year.  Exception:  2a, 3a, 4a. System has internal error  1. System terminates |
| 4 | Create local discount | Triggers/Goals: Local shop’s manager wants to launch a local discount program for a product  Actors: Local manager  Invocation constraints:  Product ID exists  Product does not have any discount program going on  Product does not have a global discount program | Main flow:  1. Local manager enters the product ID  2. Local manager enters the name of discount program  3. Local manager enters the discount factor  4. Local manager enters the valid period of discount program  5. System saves the details, and updates the new local prices in the web pages  Exception:  4a. System has internal error  1. System terminates  4b. Product already had discount program (global discount or on-going discount)  1. System notifies setting discount failed and shows the on-going discount  Post-obligations: Reset price after valid time (stated in step 4)  Note: Contact general manager if the promotion/discount is not implemented |
| 5 | Cancel local discount | Triggers/Goals: Local shop’s manager wants to cancel discount program before expiration  Actors: Local manager  Invocation constraints: The discount program has been launched | Main flow:  1. Local manager enters the launched program ID  2. System returns the details  3. Local manager presses Cancel button  4. System notifies cancellation success  Exception:  2a. ID not found  1. System asks the manager to re-enter the program ID |
| 6 | Get sales report from a shop | Triggers/Goals: General manager wants to get sales report from a specific shop  Actors: General manager  Invocation constraints: Shop ID exists | Main flow:  1. General manager enters the shop ID to get report  2. System returns the report  Exception:  1a. shop ID not found  1. System asks user to re-enter the shop ID |
| 7 | Create global discount | Triggers/Goals: General manager wants to create a global discount for a product  Actors: General manager  Invocation constraints:  Product ID exists  Product does not have any global discount program going on | Main flow:  1. General manager enters the product ID  2. General manager enters the name of discount program  3. General manager enters the discount factor  4. General manager enters the valid period of discount program  5. System saves the details, and updates the new prices in the web pages  Exceptions:  5a. Product has an ongoing local discount  1. System automatically overwrite the local discount to the global discount  5b. Product has an ongoing global discount  1. System notifies the discount program setting failed and shows the on-going global discount  Post-obligations: Reset price after discount expired  Note: Contact developer team if discount program is not implemented |
| 8 | Cancel global discount | Triggers/Goals: General manager wants to cancel a global discount program before expiration  Actors: General manager  Invocation constraints: The global discount program has already been launched | Main flow:  1. General manager enters the launched program ID  2. System returns the details  3. General manager presses Cancel button  4. System notifies cancellation success  Exception:  2a. ID not found  1. System asks the manager to re-enter the program ID |
| 9 | Add product | Triggers/Goals: General manager wants to add new product  Actors: General manager  Invocation constraints: Product ID does not exist | Main flow:  1. General manager enters the new product ID, name, price, and description.  2. System saves the details and updates the web pages  Exception:  2a. Product ID exists  1. System asks the manager to re-enter the product ID |
| 10 | Remove product | Triggers/Goals: General manager wants to stop providing a product  Actors: General manager  Invocation constraints: Product ID exists | Main flow:  1. General manager enters the product ID to be removed  2. System saves the details and update the web pages  Exception:  2a. Product ID does not exist  1. System asks the manager to re-enter the ID |
| 11 | Add staff | Triggers/Goals: Local manager wants to add a new employee  Actors: Local manager  Invocation constraints: Staff ID does not exist | Main flow:  1. Local manager enters new employee ID and employee information (specified later in project)  2. System saves the detail to the database  Exception:  1a. Employee ID exists  1. System notifies the local manager and asks the manager to repeat the process |
| 12 | Remove staff | Triggers/Goals: Local manager wants to fire an employee  Actors: Local manager  Invocation constraints: Staff ID exists | Main flow:  1. Local manager enters the ID of employee he wants to remove  2. System updates the database  Exception:  1a. Employee ID does not exist  1. System notifies the local manager and asks local manager to repeat the process |
| 13 | Open a new branch | Triggers/Goals: General manager wants to add a new branch to the database  Actors: General manager  Invocation constraints:  Branch ID does not exist  Desired branch has been opened in reality | Main flow:  1. General manager enters the branch ID, name, and address  2. System updates the database  Exception:  1a. Branch ID exists  1. System notifies the general manager and asks general manager to redo the process |
| 14 | Appoint a new manager to a branch | Triggers/Goals: General manager wants to change the manager of a branch  Actors: General manager  Invocation constraints:  Local manager ID does not exist  Branch ID exists | Main flow:  1. General manager enters the branch ID that needs new manager  2. General manager enters the new local manager ID, as well as name, and other information  3. System updates the database  Exception:  2a. Local manager ID exists  1. System notifies the error  2b. Branch ID does not exist  1. System notifies the error |

REFERENCES

[1] Dr. Mai Hoang Bao An. “Course Project”