Group 12

DBMS Final Report

National Chengchi University

Database Management Systems

109-2 (2021 Spring)

Group Info

Group 12

Leader:

107306004	顏新又

Members:

106306026	李秉澤
108306054	周謙皓
108208048	Sebastian Murcia
109300007	Charlotte Blessing

Table of Contents

Group Info 1
Table of Contents 2
Introduction 3
Requirement Analysis 4
    Economic Feasibility Analysis 4
    PERT Chart 4
Logical Design of the Business Transaction 5
    Conceptual Schema Design 5
    Relational Data Model Schema 6
    Constraints, Functionality, and Interaction with Database 7
Implementation Plan 11
    General Outline 11
    Data Flow Diagram (DFD) 11
Appendices 13
    Reference 13
    Special Thanks To 13

Introduction

The system is designed and developed for small & medium patisseries and confectionery shops. Online shopping is an unstoppable trend and has become one of the most popular online activities. Countless local pastry shops, impacted by the online stores, are faced with loss of customers and are forced to change their business model. Even with unaffectedly good sales, baking companies might have tried to sell their goods online but do not have sufficient knowledge or resources for it. Our team develops a database management system along with friendly interfaces in order to help the bakers that wanted to start their own online store and the bakery owners that planned on transforming their business virtual.

With our system, customers (the online bakeries) are able to offer their customers more options regarding delivery methods, coupons, and discounts. Their brand is exposed to more potential customers because of the unpretentious aspects of cybershopping, containing but not limited to privacy when shopping, more convenient shopping, no-pressure shopping, etc. Moreover, consumers can check on products availability in front of their computers instead of spending time driving to the shop location.

In addition to the natural advantages of online shopping, our DBMS as well assist the dessert business in managing the ingredients and inventory. Sweets and their ingredients usually have a short expiration date. Our system shows the status and usage of ingredients and process and sales of the products, which can reduce the waste of resources. Also, we have other functions such as customer management, recipe details, coupon management for the business' operation.

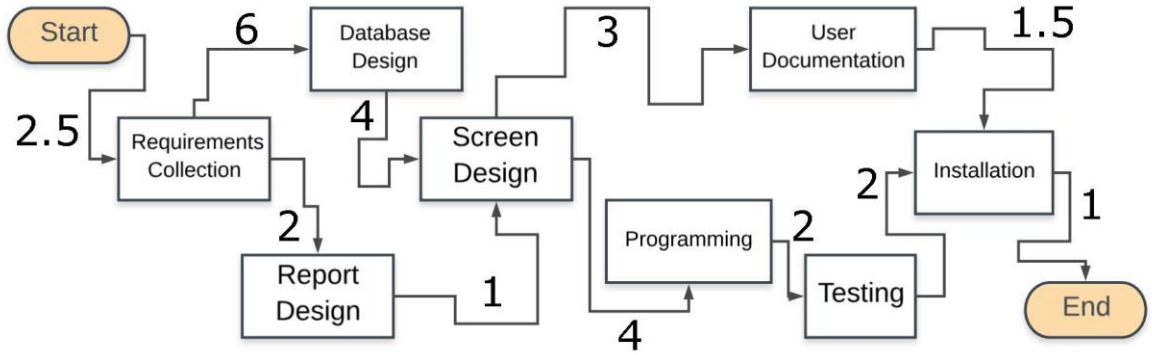
Requirement Analysis

Economic Feasibility Analysis

Expense:
  Labor cost: (Project Duration: 5 months)
        Front-end programmer (1): 1 * 40,000 * 5 = 200,000
        Database programmer (1): 1 * 40,000 * 5 = 200,000
        Testing programmer (1): 1 * 36,000 * 5 = 180,000
        Project manager (1): 1 * 50,000 * 5 = 250,000
        Marketing (1): 1 * 36,000 * 5 = 180,000
 Total: 1,010,000
Maintenance (5): 5 * 32,000 * 12 = 1,920,000 / year
Advertising: 1,000,000 (if CPC is 10, then it will create 100,000 clicks)
Total Expense on 1st year: 3,930,000
Revenue:
  Subscribe: 200 (assuming within 100,000 clicks, 0.2% becomes our clients) * 1,000 * 12 = 2,400,000 / year
  Other service: 2,000 * 200 (annual maintenance) = 400,000
  Total Revenue in 1st year: 2,800,000
Total Profit in 1st year: -1,130,000

(The development team will abort and the subscribers will increase; accordingly, the increase of profit in the following years can be expected.)

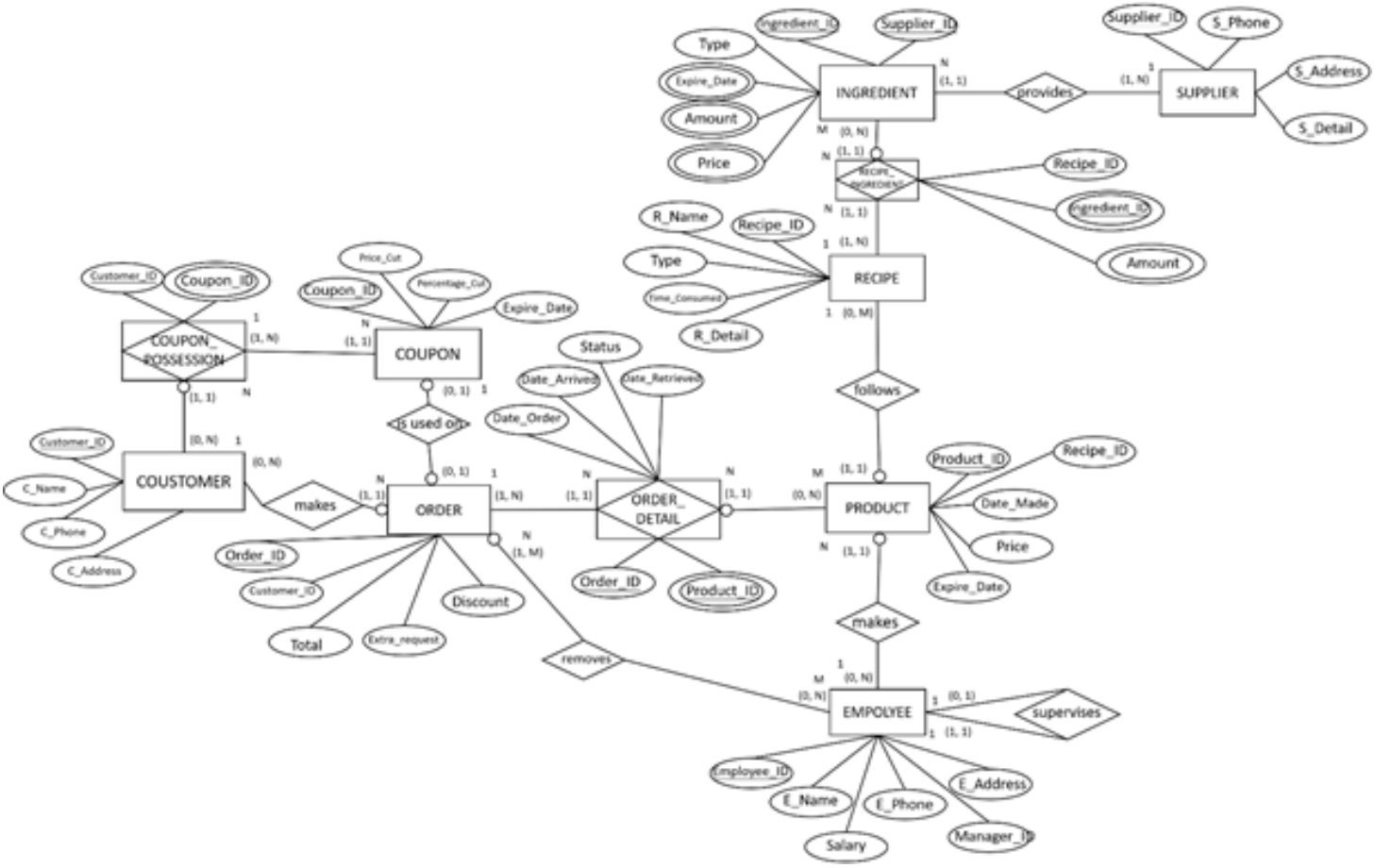
PERT Chart



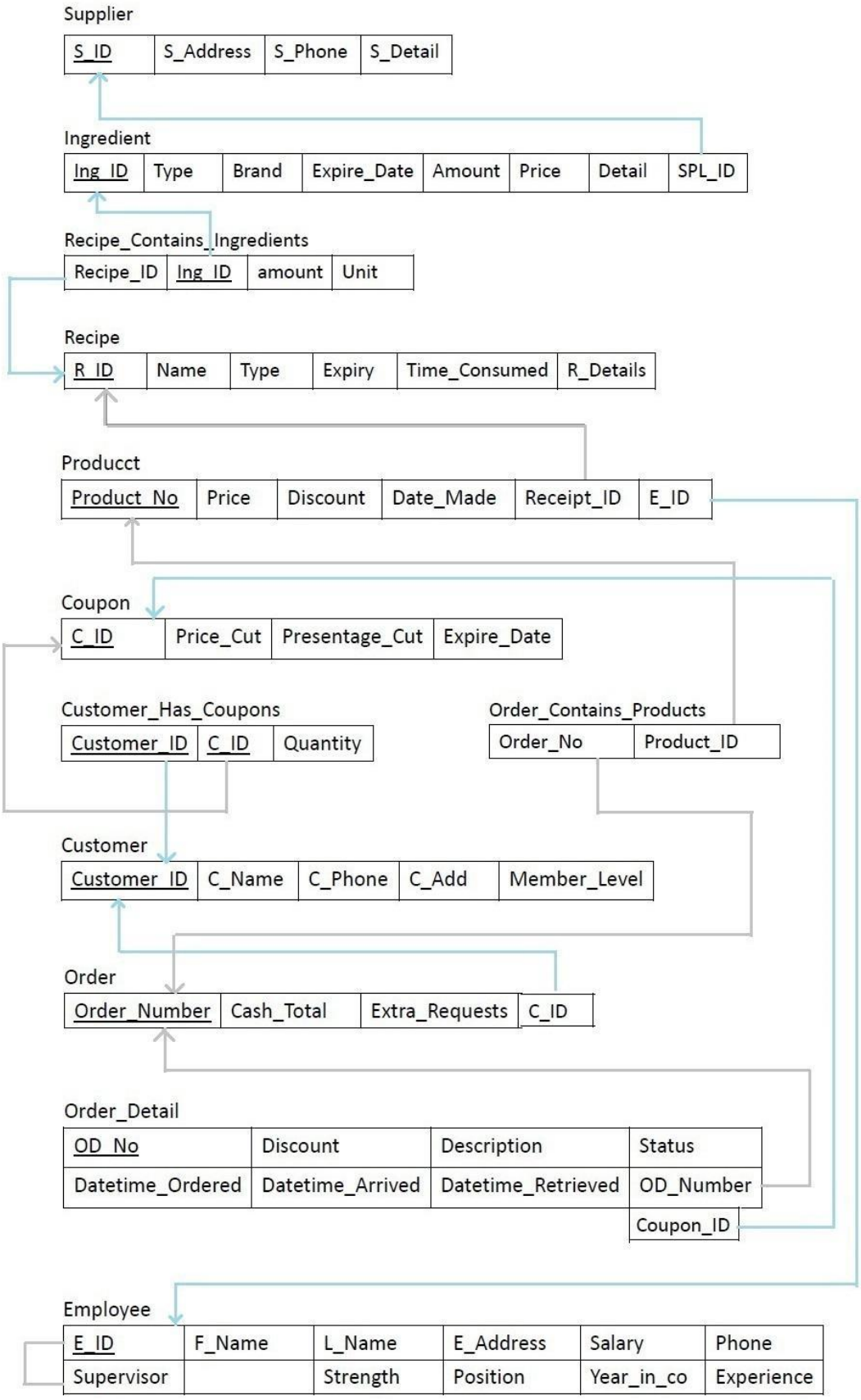
(Time period: in weeks)

Logical Design of the Business Transaction

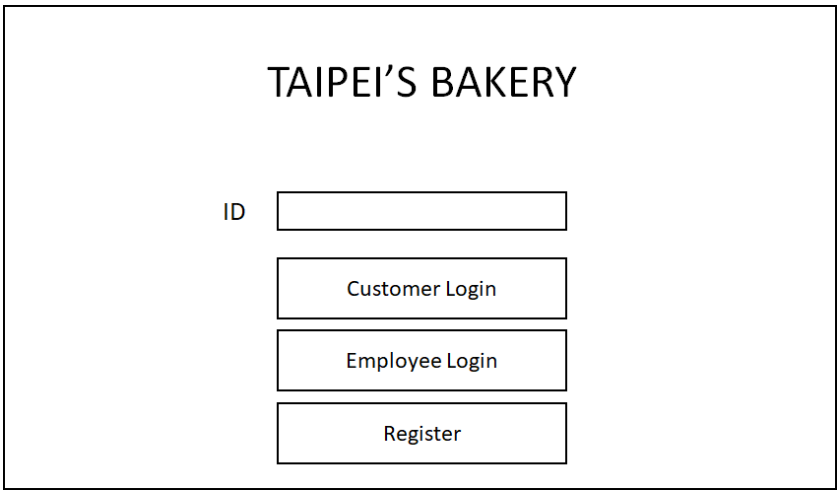
Conceptual Schema Design



Relational Data Model Schema



Constraints, Functionality, and Interaction with Database

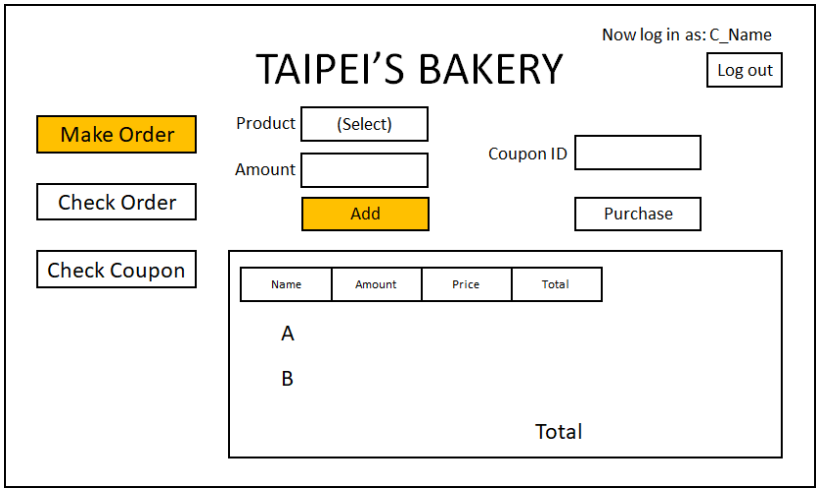


The password system was not implemented since we are convinced that passwords tend to be problematic if they are simply stored in a database before being encrypted. Thus, for the current version, customers are allowed to log-in with their ID, which may evoke some problems; however, not until an encryption method is implemented, passwords would be stored in a database.

- Customer login: Read **CUSTOMER**, check if input value exists.
- Employee login: Do not need to input value.

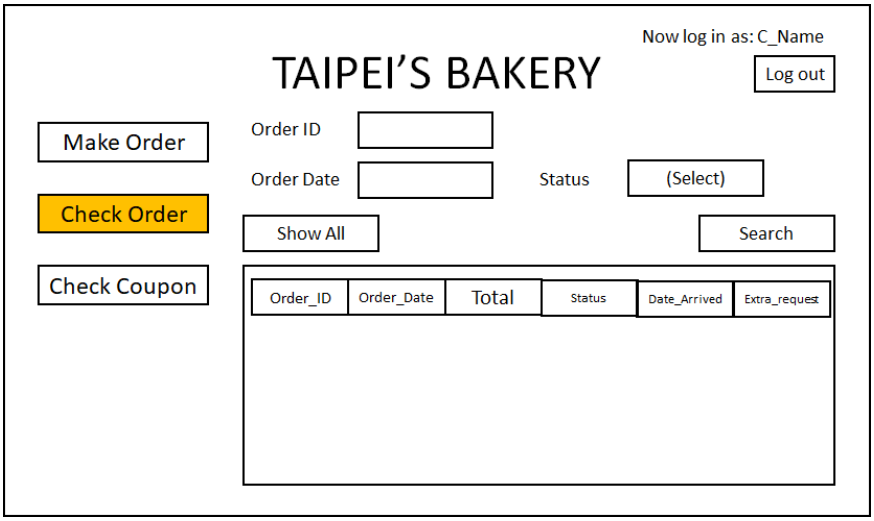
Customers can easily register with their name, phone number, and address. The system will automatically generate a distinct ID for every customer. Distinct IDs makes sure each customer is unique in the system.

- Register: Update **CUSTOMER**.



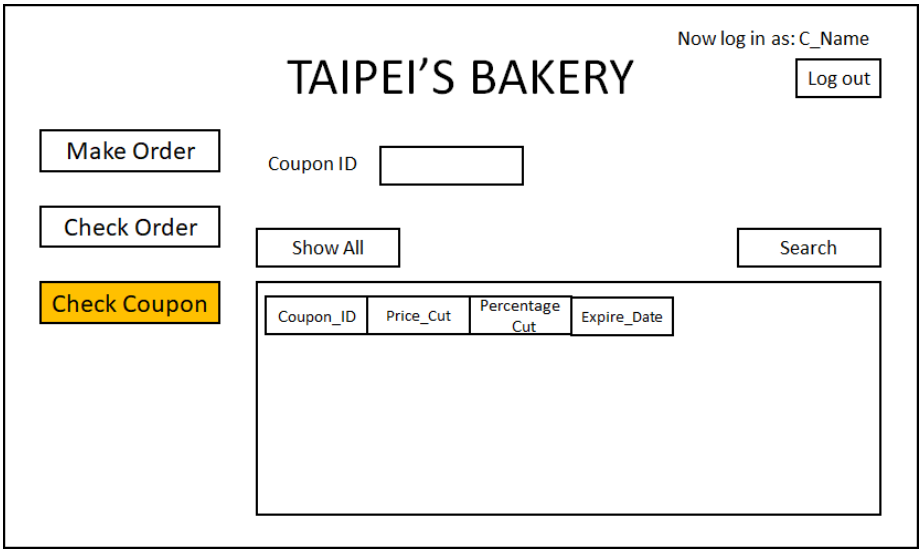
Customers are able to make an order (which is the most important function in the system) by selecting products and amounts they desire. If they have a coupon, they can also input coupon ID to get a discount on their purchase. Coupons can be checked in "Check Coupon." All products should be added to an order list before being purchased. It is required to click on "Add" before "Purchase."

- Add order: Update **ORDER**, **ORDER_DETAIL** and **ORDER_PROUDCT_LIST**.



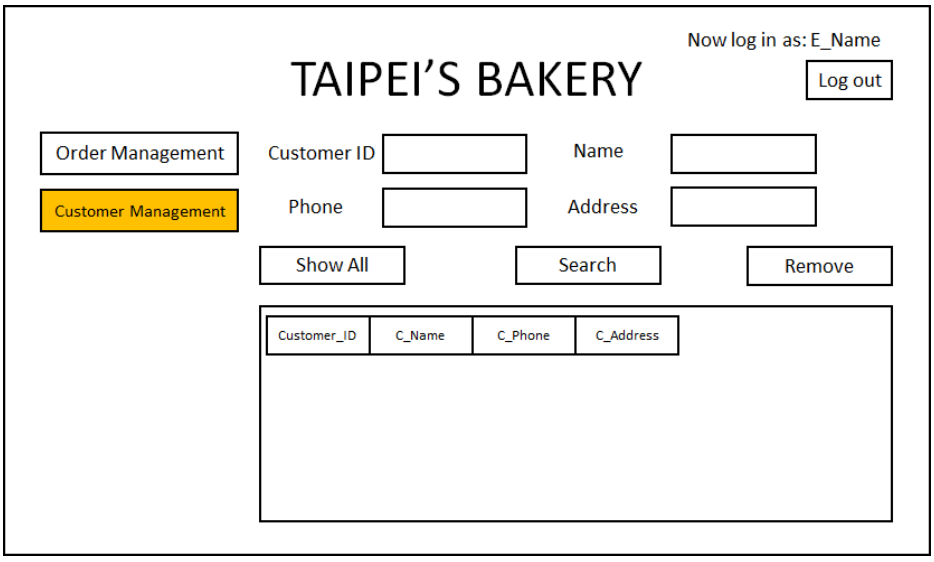
Customers can check their orders at this page. They can see all the orders they made simply by clicking on "Show ALL", or specific ones by inputting values and clicking on "Search". Filter by "Status" is not working well at the moment. All customers are encouraged to search by "Order ID" and "Order Date."

- Show all: Read **ORDER**, **ORDER_DEATIL**.
- Search: Read **ORDER**, **ORDER_DEATIL** with given input values.



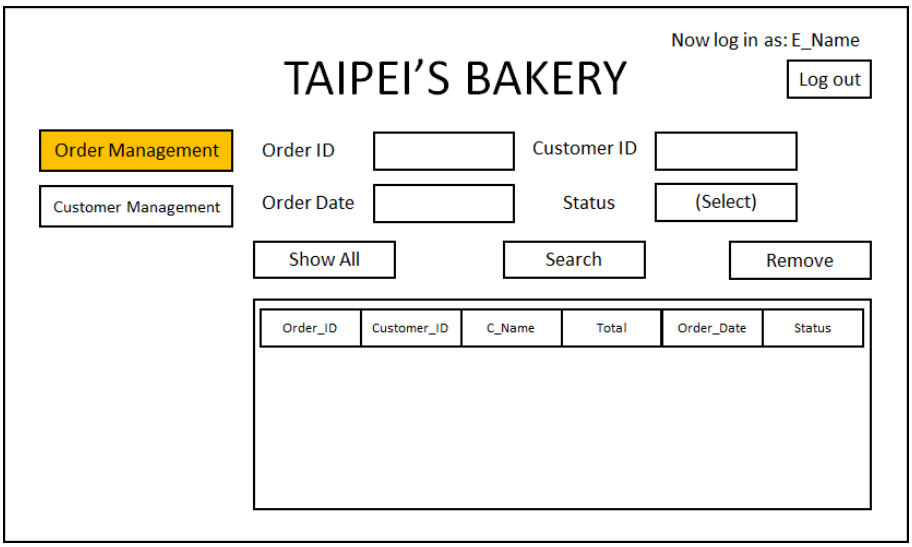
Customers can check their coupons here. After checking coupons, customers can use these coupons on their purchases.

- Show all: Read **COUPON** and **COUPON_POSSESSION**.
- Search: Read **COUPON** and **COUPON_POSSESSION** with given input values.



Employees can check and remove customers' information here. Since only the customer ID is identical, employees can remove only customers with customer ID (input any other values will cause some problems). However, employees can search any customer by inputting any values.

- Show all: Read **CUSTOMER**.
- Search: Read **CUSTOMER** with given input values.
- Remove: Update **CUSTOMER** with given input ID.

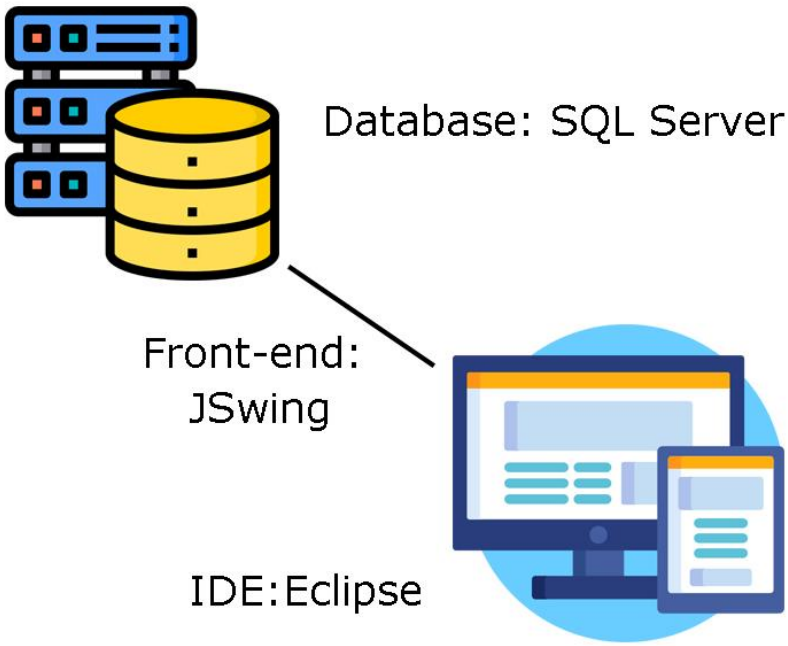


Employees can check and remove orders here. Searching with “Status” is not functioning currently. Searches can be ordered by “Order ID,” “Customer ID,” and “Order Date”. In addition to that, since only order ID is distinct, such is needed when it comes to removing orders (input any other values will cause problems).

- Show all: Read **ORDER**, **ORDER_DETAIL**.
- Search: Read **ORDER**, **ORDER_DETAIL** with given input values.
- Remove: Update **ORDER** with given input ID.

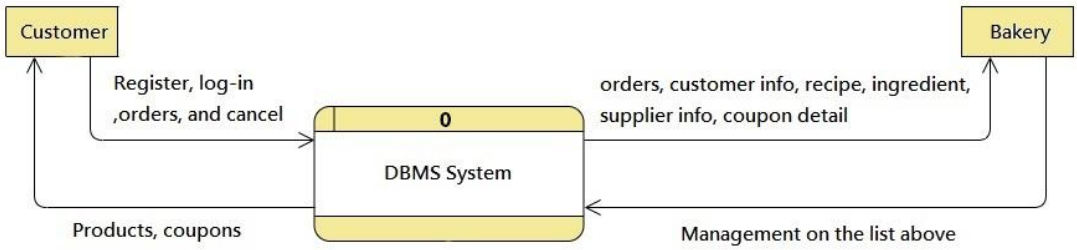
Implementation Plan

General Outline

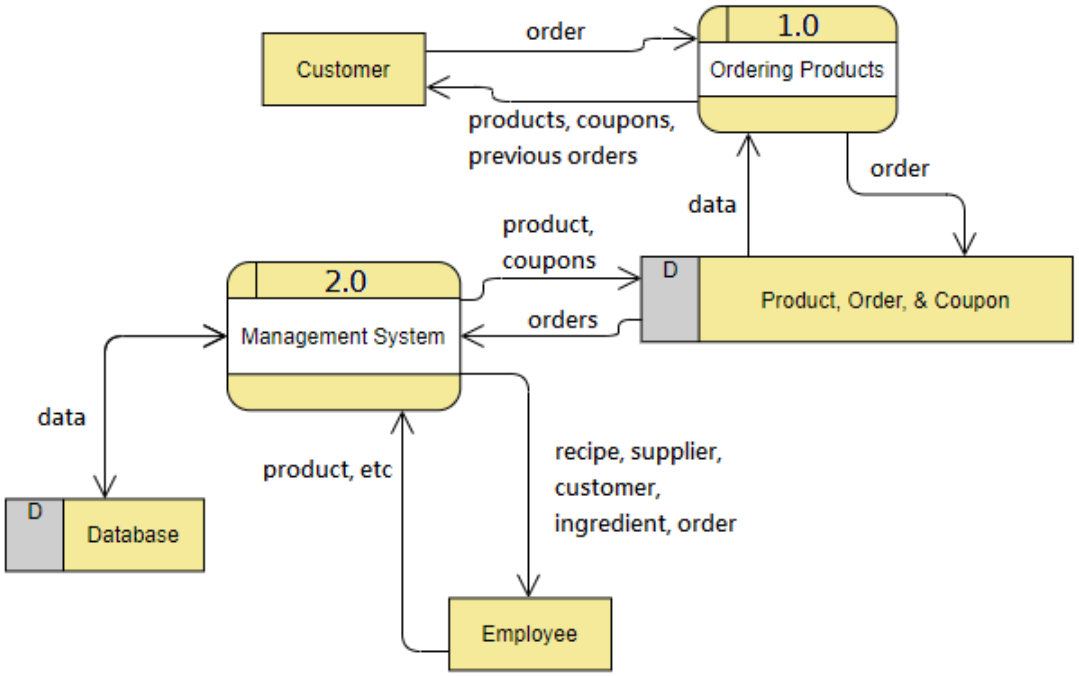


Data Flow Diagram (DFD)

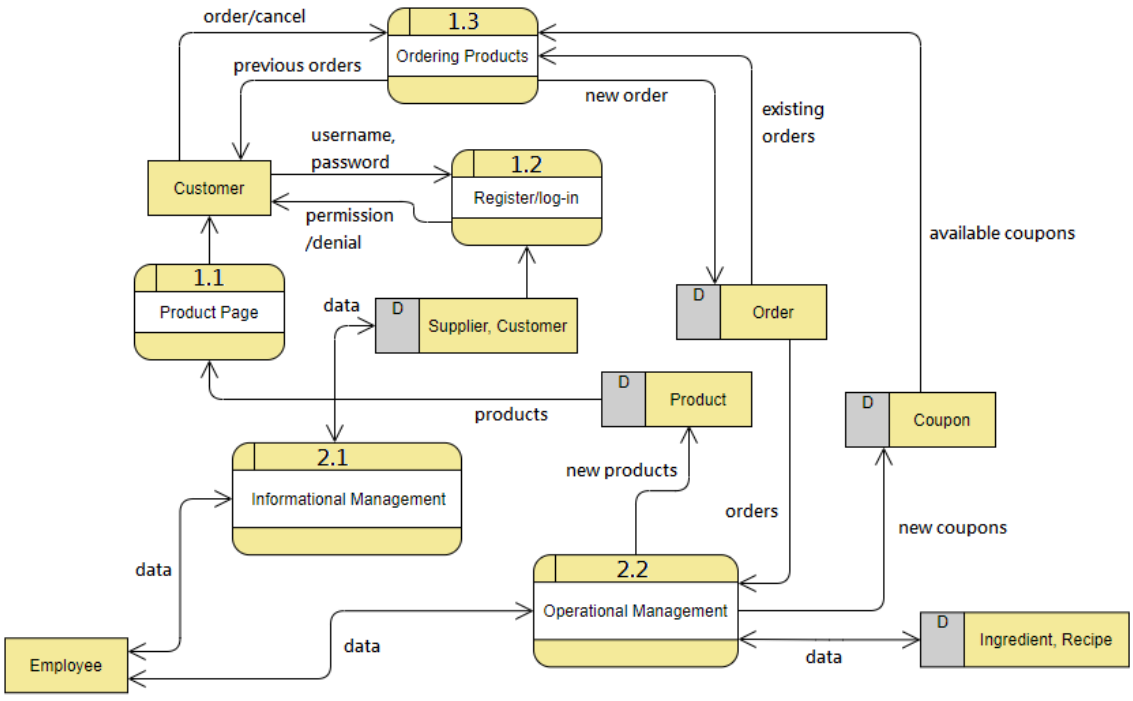
Context Diagram



Level 0



Level 1



Appendices

Reference

https://mset.nccu.edu.tw/course/view.php?id=62

Special Thanks To

Instructor: Chih-Yuan Chou (Ben)
Teacher's assistant: Bo-Yi Li, Yu Xi Tan, & Joy Huang
Every member in Group 12
And finally, everyone that participated in this class!