# COMP3013 DBMS Project Report (Group 2)

Members:

Ni Fangbo 2330026124

Miao Yan 2330026122

Li Longji 2330026079

Lin Tingheng 2330026093

Lin Hongjia 2330026091

## 1. Background information of the project

Delivery services are significant in today's world. An efficient management system improves both customer and business satisfaction, so we are working on a delivery platform to make it easier for customers, deliveryman and managers to interact on the delivery information.

Customers need to choose products, place orders and track orders. Deliveryman need to pending order information to take the order and get delivery fee after completing the order. Managers need to be able to manage products, process orders, get user feedback.

Therefore, our aim is to implement a delivery platform which has functions to satisfy the need of customer, deliveryman and manager.

## 2. List of front-end functions

(1) Customer:

Recording and modifying homepage information

Looking through the products which are currently available

Adding products to the cart

Placing orders for the products in the cart

Searching for order records

Making comments to the products in the completed order

(2) Deliveryman:

Recording and modifying the homepage information

Looking through the pending orders

Check the information of delivering orders

Complete the delivering orders and get the delivery fee

Cancel the delivering orders and return the fee back to customer

(3) Manager:

Recording and modifying the homepage information

Searching for the detailed information of products, orders, users, and comments

Updating information of products, orders, users, and comments

Deleting user accounts or comments

Adding products

Adding users

3. Assumptions for our problem

(1) Registration assumption:

New user can only register as a customer. Deliveryman account can only be added by the manager. The manager account is unique and cannot be changed.

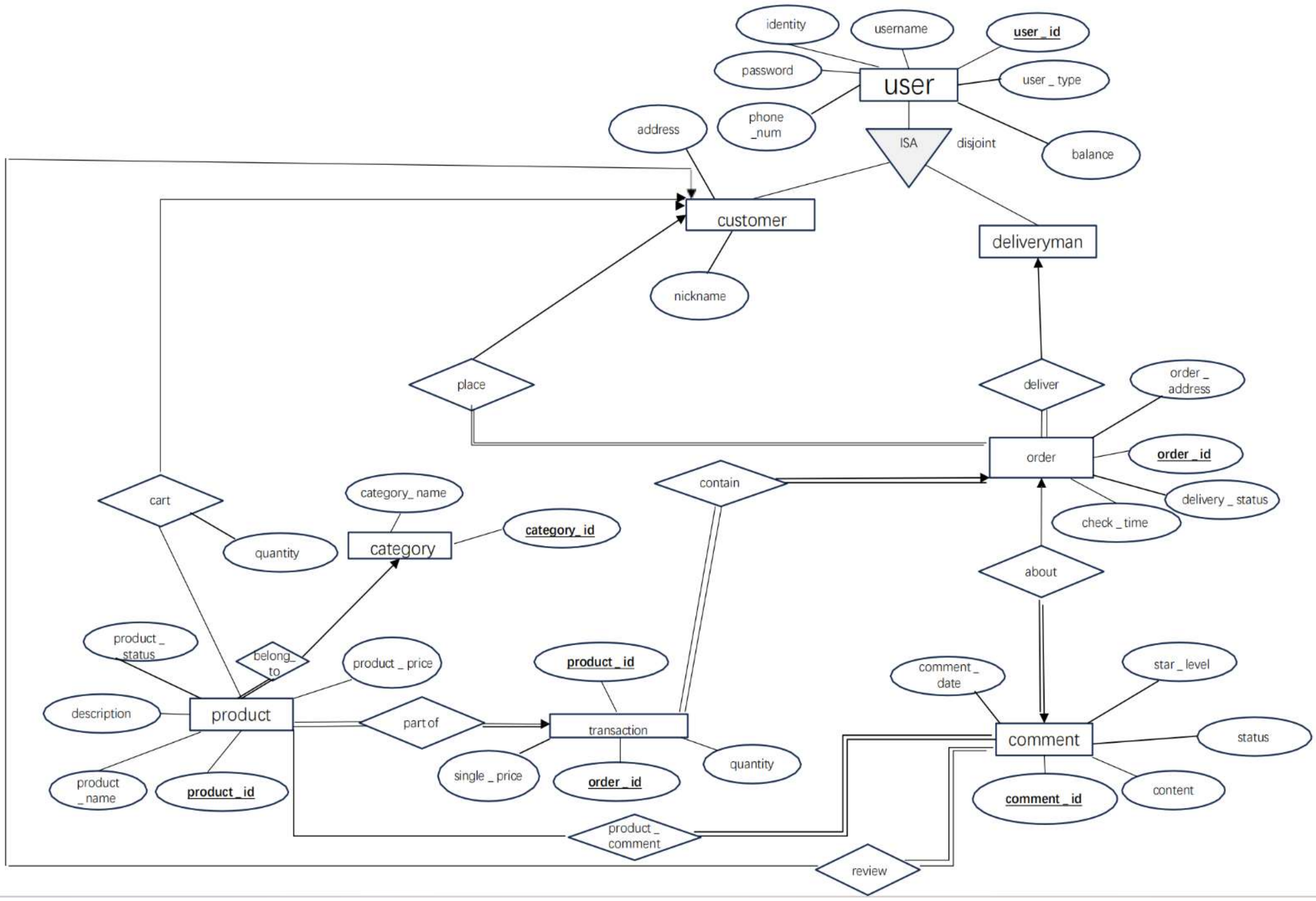
(2) Product assumption:

Products are only distinguished by id, and there can exists products with same name. Each product has status as available or unavailable, and it cannot be deleted. After the product price is changed, the price of the same product in the old order will not be changed.

(3) Account balance assumption:

For safety, customer or deliveryman has no permission to modify the balance. Balance can only be modified by the manager. To make the process easier, each new user will initially have balance 100$.

4. ER-diagram



Entity sets:

(1) **User** has id, username, password, identity, phone, and balance. A user can be specialized into **customer** and **deliveryman**. A customer has nickname and address.

(2) **Category** has id and name.

(3) **Product** has id, name, price, description and status.

(4) **Order** has id, address, check time and delivery status.

(5) **Transaction** has product id, order id, single price and quantity.

(6) **Comment** has id, content, date, star level and status.

Relationship sets:

(1) One customer can place many orders. (place)

(2) One deliveryman can deliver many orders. (deliver)

(3) One customer can put many products into cart. (cart)

(4) One customer can make many comments. (review)

(5) One comment is about one order. (about)

(6) Many products belong to one category. (belong to)

(7) Many products are included in transaction. (part of)

(8) One comment corresponds to many products and one product correspond to many comments. (product-comment)

(9) Every order contains a transaction. (contain)

## 5. Schema and functional dependencies

(All of our schemas satisfy **BCNF**. Primary key are described by underline.)

user = (user-id, user-type, username, password)

F = {user-id→username, user-type, password}

User-id is super-key, which satisfies BCNF.

customer = (customer-id, nickname, phone, address, identity, balance)

// customer-id is the foreign key to user(user-id)

F = {customer-id→nickname, phone, address, identity, balance}

Customer-id is super-key, which satisfies BCNF.

deliveryman = (deliveryman-id, phone, identity, balance)

// deliveryman-id is the foreign key to user(user-id)

F = {deliveryman-id→phone, identity, balance}

Deliveryman-id is super-key, which satisfies BCNF.

order = (order-id, customer-id, order-time, order-address, deliveryman-id, check-time, delivery-status, complete-time)

// customer-id is the foreign key to customer(customer-id)

F = {order-id→customer-id, order-time, order-address, deliveryman-id, check-time, delivery-status, complete-time}

Order-id is super-key, which satisfies BCNF.

category = (category-id, category-name)

F = {category-id→category-name}

Category-id is super-key, which satisfies BCNF.

product = (product-id, product-name, product-price, description, image, status)

F = {product-id→product-name, product-price, description, image, status}

Product-id is super-key, which satisfies BCNF.

comment = (comment-id, order-id, comment-date, content, star-level)

// order-id is the foreign key to order(order-id)

F = {comment-id→order-id, comment-date, content, star-level}

Comment-id is super-key, which satisfies BCNF.

product-category = (product-id, category-id)

// product-id is the foreign key to product(product-id), category-id is the foreign key to category(category-id)

F = {product-id→category-id}

Product-id is super-key, which satisfies BCNF.

transaction = (order-id, product-id, single-price, quantity)

// order-id is the foreign key to order(order-id), product-id is the foreign key to product(product-id)

F = {order-id, product-id→single-price, quantity}

{order-id, product-id} is super-key, which satisfies BCNF.

customer-comment = (comment-id, customer-id)

// comment-id is the foreign key to comment(comment-id), customer-id is the foreign key to customer(customer-id)

F = {comment-id→customer-id}

Comment-id is super-key, which satisfies BCNF.

comment-product = (comment-id, product-id)

// comment-id is the foreign key to comment(comment-id), product-id is the foreign key to product(product-id)

F = {comment-id→product-id}

Comment-id is super-key, which satisfies BCNF.

cart = (customer-id, product-id, quantity)

// customer-id is the foreign key to customer(customer-id), product-id is the foreign key to product(product-id)

F = {customer-id, product-id→quantity}

{customer-id, product-id} is super-key, which satisfies BCNF.

6. Workload of each team member:

| | |
|---|---|
| Lin Hongjia 2330026091 | design front-end features & ppt |
| Ni Fangbo 2330026124 | back-end & SQL query |
| Miao Yan 2330026122 | front-end code |
| Li Longji 2330026079 | implement database |
| Lin Tingheng 2330026093 | ER diagram design |