Georgia Express - DBMS Term Project

Project Documentation

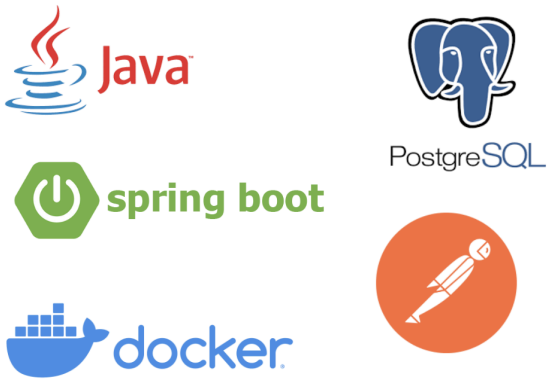
Team Members: Michael Robinson, Bandhan Patel, Dias Mashikov, Santos Ochoa, Het Pathak

Introduction

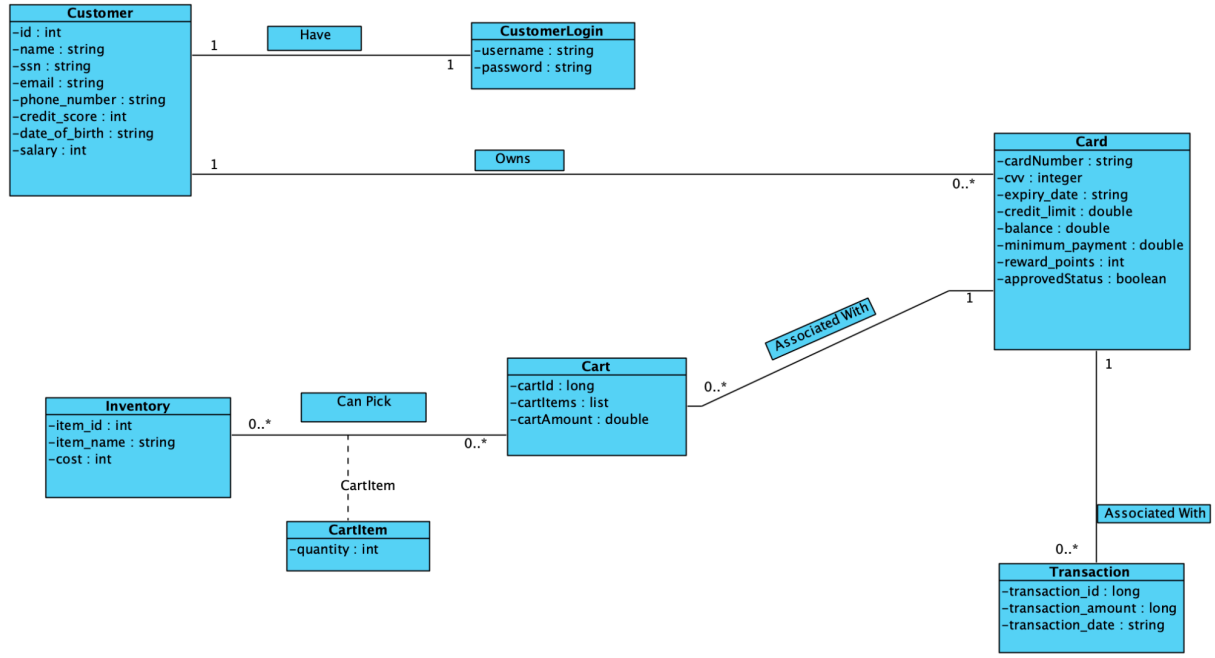
Georgia Express is an application that allows you to apply for a credit card and utilize it in its online marketplace.

Tech stack

- Java
- Spring Boot
- Docker
- PostgreSQL
- Postman



UML Diagram



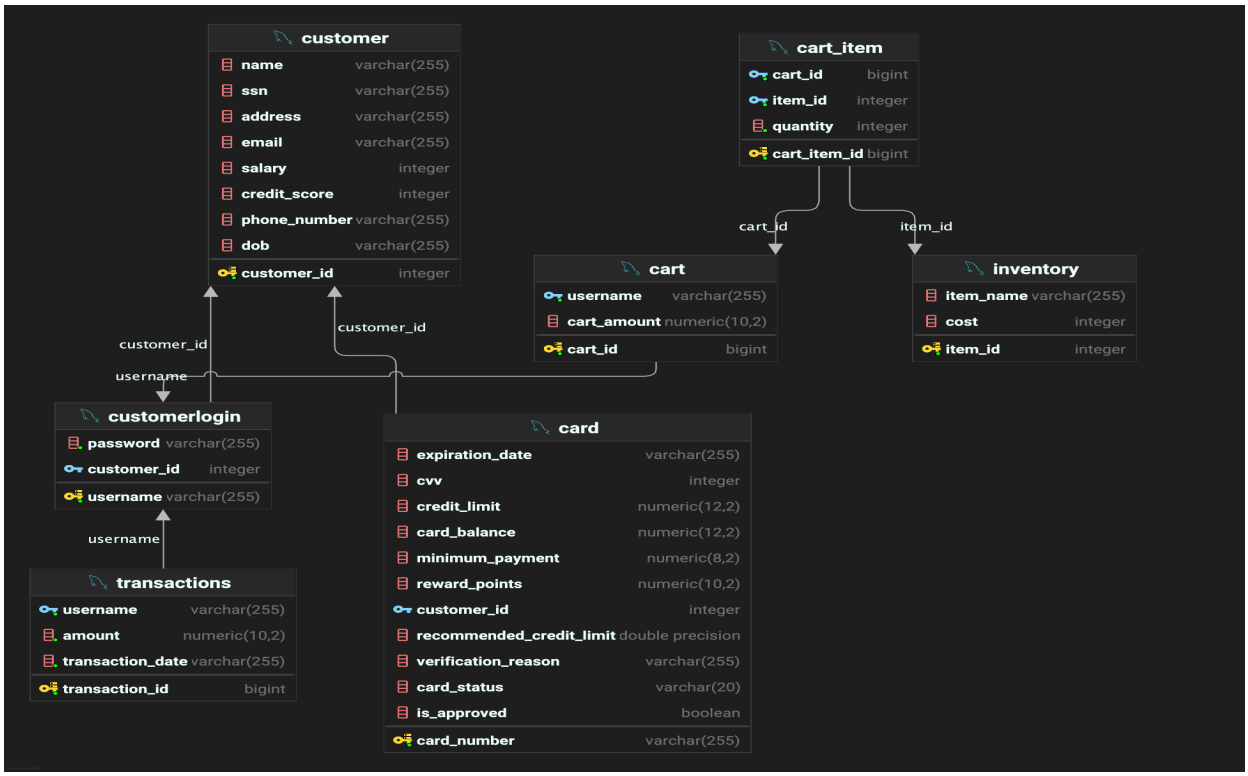
Explanation:

Classes:
1. Customer: Customer information
2. CustomerLogin: Customer login information
3. Card: Credit card information
4. Cart: A virtual shopping cart information
5. Inventory: List of items and their information in the storefront
6. Transaction: Information about a storefront transaction

Relationships:
1. A Customer can have only a single CustomerLogin (1 to 1)
2. A Customer can own multiple Cards (1 to many)
3. Multiple carts can be associated with a single credit card (many to 1)
4. A card can have multiple transactions associated to it (1 to many)
5. Multiple carts can pick multiple items from the inventory (many to many)
6. With the Cart-Inventory many to many relationships we can build a associated class called CartItem that will connect the two classes

E-R Diagram from UML:

After analyzing the UML diagram we generate the following E-R Diagram:



Explanation:

We define the following the tables to represent the given classes:

1. customer: For Customer class with PK: customer_id
2. card: For Card class with PK: card_number
3. customerlogin: For CustomerLogin class with PK: username
4. inventory: For Inventory class with PK: item_id
5. transactions: For Transaction class with PK: transaction_id
6. cart: For Cart class with PK : cart_id
7. cartItem : For the associative class CartItem with PK: cart_item_id

Note: CartItem should have combined primary key of cart_id,item_id but we added the extra element cart_item_id to get the PK as it will work seamlessly with the code

Foreign Key relationships:
1. customer_id in customerlogin references PK(customer_id) in customer
2. customer_id in card references PK(customer_id) in customer
3. cart_id in cartitem references PK(cart_id) in cart
4. item_id in cartitem references PK(item_id) in inventory
5. username in transactions references PK(username) in customerlogin
6. username in cart references PK(username) in customerlogin

Schema Information

Schema:

R(customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment,reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date)

Functional dependencies:

1. customer_id –> name,ssn,address,email,salary,credit_score,phone_number,dob

2. username –> customer_id, password

3. card_number –> customer_id, expiration_date,cvv,credit_limit,card_balance,minimum_payment,reward_points,

recommended_credit_limit, verification_reason, card_status,is_approved

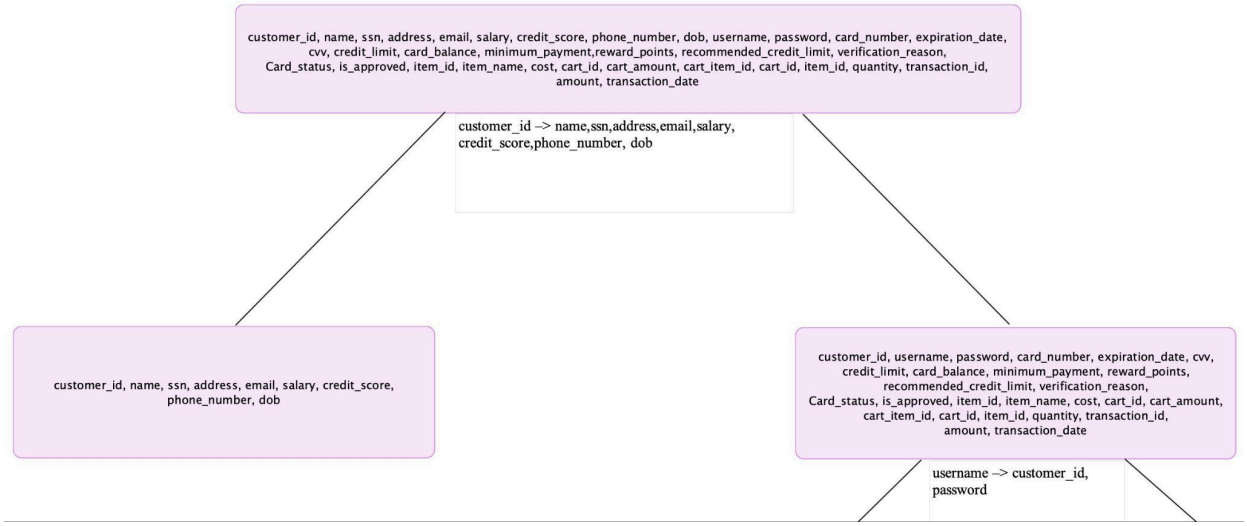
4. item_id –> item_name, cost

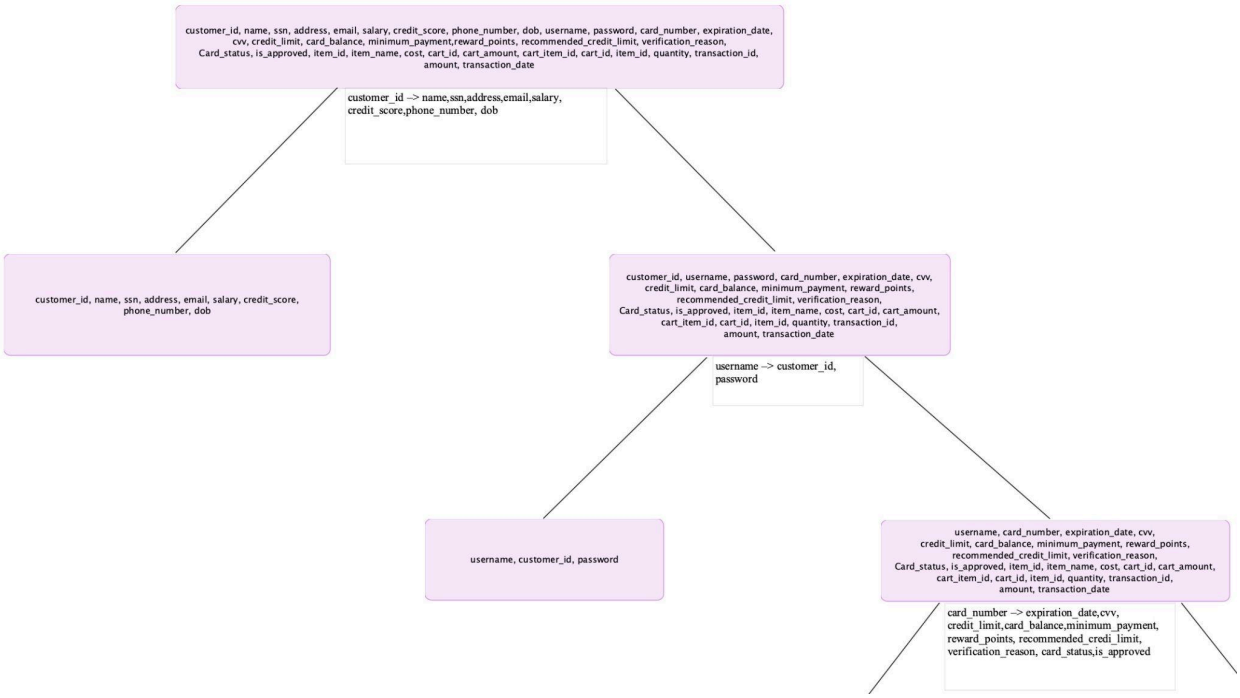
5. cart_id –> username, cart_amount

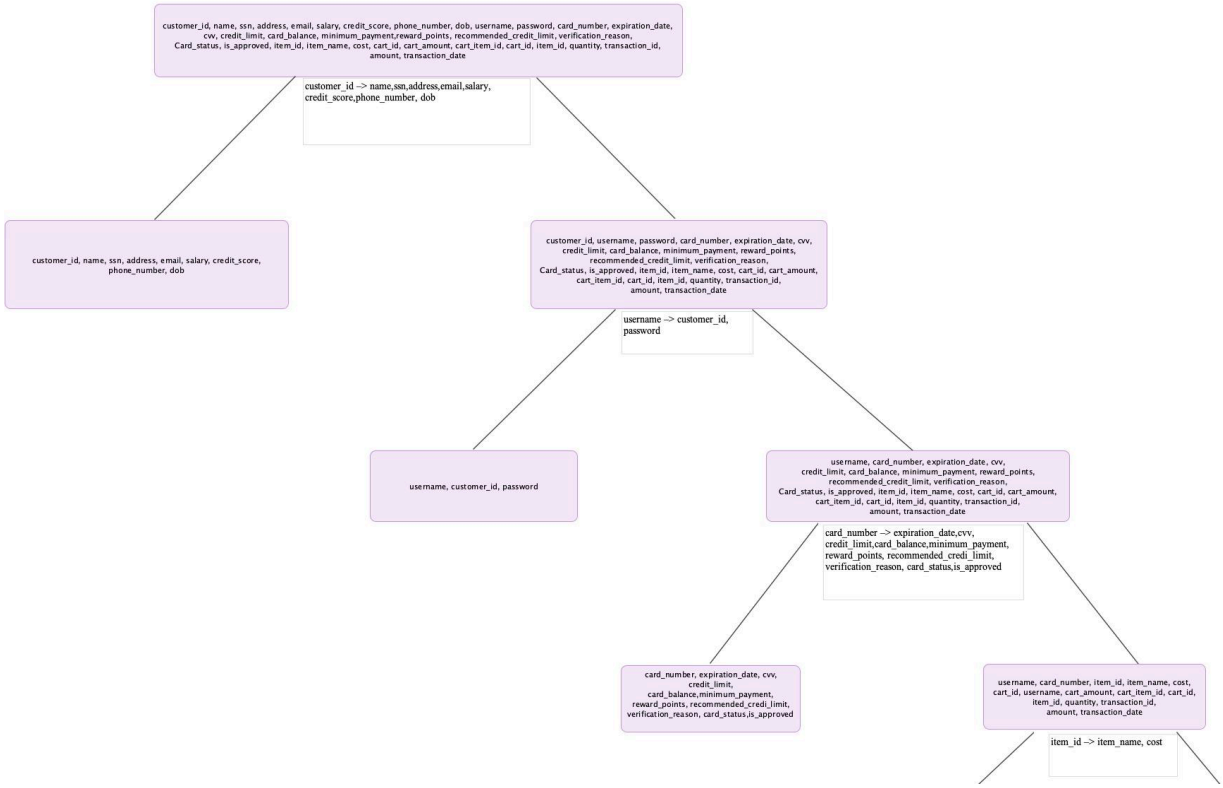
6. cart_item_id –> cart_id, item_id, quantity

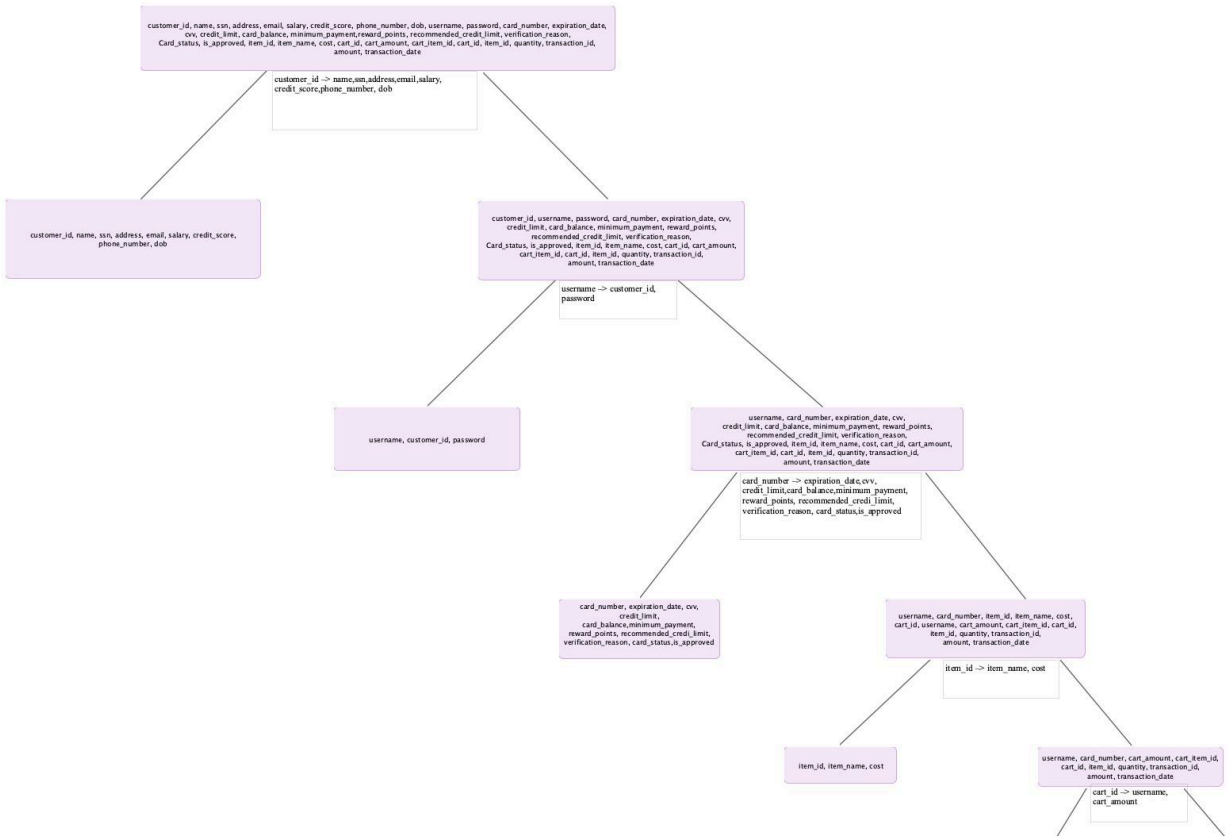
7. transaction_id –> username, amount, transaction_date

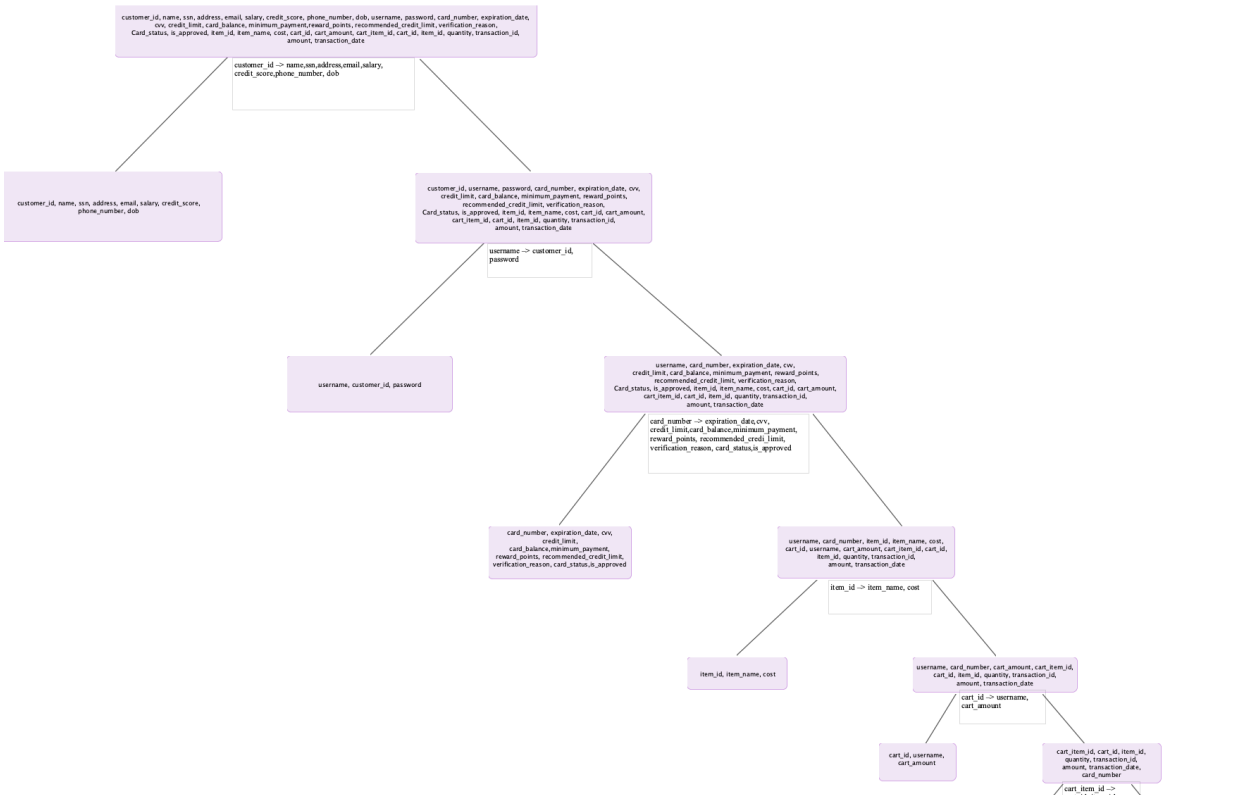
BCNF Decompostion

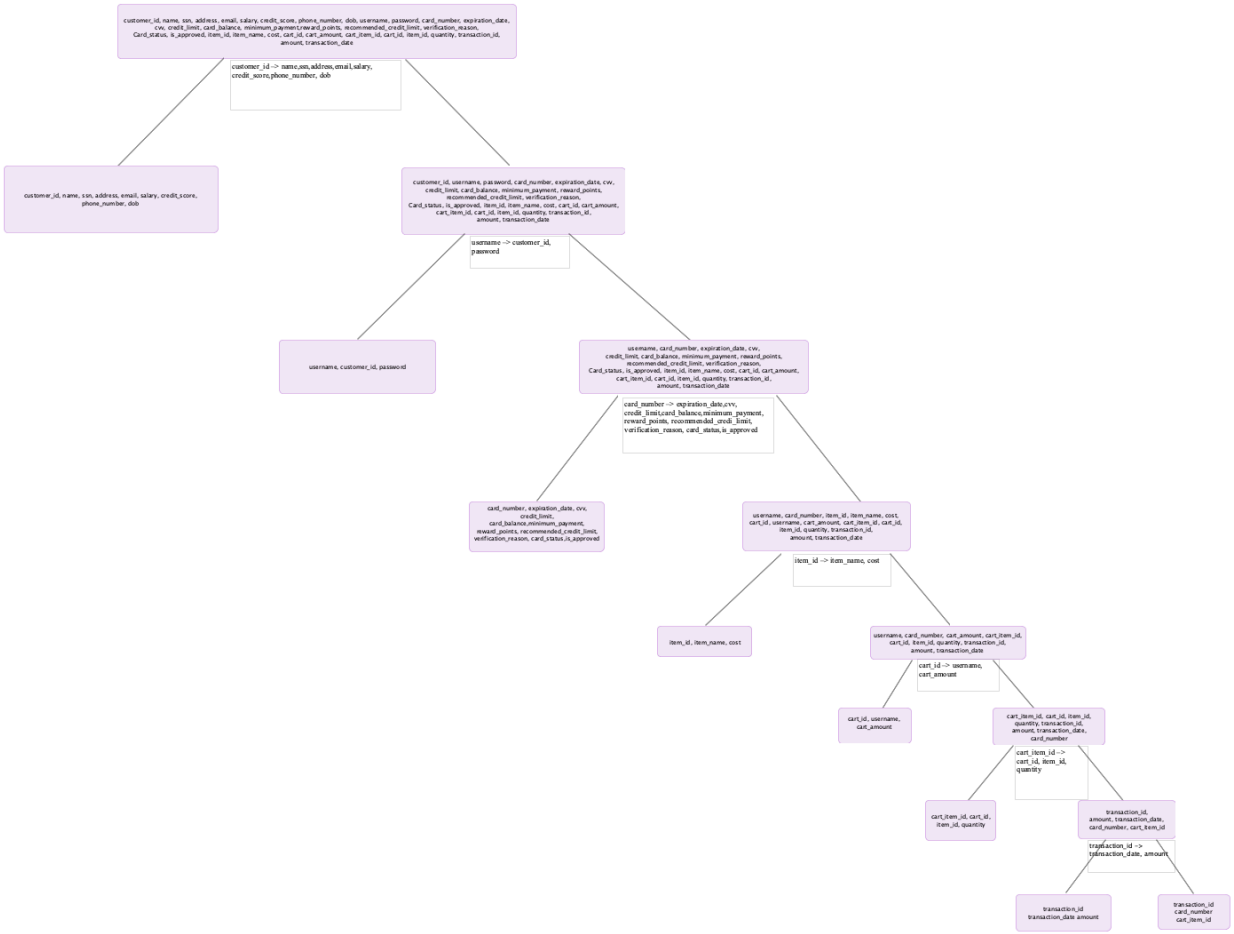

customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment,reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
customer_id –> name,ssn,address,email,salary, credit_score,phone_number, dob
customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob
customer_id, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
username –> customer_id, password


customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment,reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
customer_id –> name,ssn,address,email,salary, credit_score,phone_number, dob
customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob
customer_id, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
username –> customer_id, password
username, customer_id, password
username, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
card_number –> expiration_date,cvv, credit_limit,card_balance,minimum_payment, reward_points, recommended_credi_limit, verification_reason, card_status,is_approved


customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment,reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
customer_id –> name,ssn,address,email,salary, credit_score,phone_number, dob
customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob
customer_id, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
username –> customer_id, password
username, customer_id, password
username, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
card_number –> expiration_date,cvv, credit_limit,card_balance,minimum_payment, reward_points, recommended_credi_limit, verification_reason, card_status,is_approved
card_number, expiration_date, cvv, credit_limit, card_balance,minimum_payment, reward_points, recommended_credi_limit, verification_reason, card_status,is_approved
username, card_number, item_id, item_name, cost, cart_id, username, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
item_id –> item_name, cost


customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment,reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
customer_id –> name,ssn,address,email,salary, credit_score,phone_number, dob
customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob
customer_id, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
username –> customer_id, password
username, customer_id, password
username, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
card_number –> expiration_date,cvv, credit_limit,card_balance,minimum_payment, reward_points, recommended_credi_limit, verification_reason, card_status,is_approved
card_number, expiration_date, cvv, credit_limit, card_balance,minimum_payment, reward_points, recommended_credi_limit, verification_reason, card_status,is_approved
username, card_number, item_id, item_name, cost, cart_id, username, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
item_id –> item_name, cost
item_id, item_name, cost
username, card_number, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
cart_id –> username, cart_amount


customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment,reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
customer_id -> name,ssn,address,email,salary, credit_score,phone_number, dob
customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob
customer_id, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
username -> customer_id, password
username, customer_id, password
username, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
card_number -> expiration_date,cvv, credit_limit,card_balance,minimum_payment, reward_points, recommended_credit_limit, verification_reason, card_status,is_approved
card_number, expiration_date, cvv, credit_limit, card_balance,minimum_payment, reward_points, recommended_credit_limit, verification_reason, card_status,is_approved
username, card_number, item_id, item_name, cost, cart_id, username, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
item_id -> item_name, cost
item_id, item_name, cost
username, card_number, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
cart_id -> username, cart_amount
cart_id, username, cart_amount
cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date, card_number
cart_item_id ->
customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment,reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
customer_id -> name,ssn,address,email,salary, credit_score,phone_number, dob
customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob
customer_id, username, password, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
username -> customer_id, password
username, customer_id, password
username, card_number, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, Card_status, is_approved, item_id, item_name, cost, cart_id, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
card_number -> expiration_date,cvv, credit_limit,card_balance,minimum_payment, reward_points, recommended_credit_limit, verification_reason, card_status,is_approved
card_number, expiration_date, cvv, credit_limit, card_balance,minimum_payment, reward_points, recommended_credit_limit, verification_reason, card_status,is_approved
username, card_number, item_id, item_name, cost, cart_id, username, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
item_id -> item_name, cost
item_id, item_name, cost
username, card_number, cart_amount, cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date
cart_id -> username, cart_amount
cart_id, username, cart_amount
cart_item_id, cart_id, item_id, quantity, transaction_id, amount, transaction_date, card_number
cart_item_id -> cart_id, item_id, quantity
cart_item_id, cart_id, item_id, quantity
transaction_id, amount, transaction_date, card_number, cart_item_id



Schema Design In 3NF

Step 1: Find Minimal Cover

1. Convert to Singleton FDs:
   - customer_id → name
   - customer_id → ssn
   - customer_id → address
   - customer_id → email
   - customer_id → salary
   - customer_id → credit_score
   - customer_id → phone_number
   - customer_id → dob
   - username → customer_id
   - username → password

- card_number → customer_id
- card_number → expiration_date
- card_number → cvv
- card_number → credit_limit
- card_number → card_balance
- card_number → minimum_payment
- card_number → reward_points
- card_number → recommended_credit_limit
- card_number → verification_reason
- card_number → card_status
- card_number → is_approved
- item_id → item_name
- item_id → cost
- cart_id → username
- cart_id → cart_amount
- cart_item_id → cart_id
- cart_item_id → item_id
- cart_item_id → quantity
- transaction_id → username
- transaction_id → amount
- transaction_id → transaction_date

2. Remove Redundant FDs:
- No redundant FDs in this case, so FDs will remain unchanged.

3. Remove Extraneous Attributes:
- There are no extraneous attributes present as each dependency has a single attribute on the left-hand side (LHS)

Step 2: Merge FDs with same LHS

1. customer_id → dob, phone_number, credit_score, salary, email, address, ssn, name
2. username → password, customer_id
3. card_number → is_approved, card_status, verification_reason, recommended_credit_limit, reward_points, minimum_payment, card_balance, credit_limit, cvv, expiration_date, customer_id
4. item_id → cost, item_name
5. cart_id → cart_amount, username
6. cart_item_id → quantity, item_id, cart_id

7. transaction_id → transaction_date, amount, username

Step 3: Form a table for each FD

- R1 = {customer_id, dob, phone_number, credit_score, salary, email, address, ssn, name}
- R2 = {username, password, customer_id}
- R3 = {card_number, is_approved, card_status, verification_reason, recommended_credit_limit, reward_points, minimum_payment, card_balance, credit_limit, cvv, expiration_date, customer_id}
- R4 = {item_id, cost, item_name}
- R5 = {cart_id, cart_amount, username}
- R6 = {cart_item_id, quantity, item_id, cart_id}
- R7 = {transaction_id, transaction_date, amount, username}

Step 4: Remove Subset tables

No subset tables in this case.

- R1 = {customer_id, dob, phone_number, credit_score, salary, email, address, ssn, name}
- R2 = {username, password, customer_id}
- R3 = {card_number, is_approved, card_status, verification_reason, recommended_credit_limit, reward_points, minimum_payment, card_balance, credit_limit, cvv, expiration_date, customer_id}
- R4 = {item_id, cost, item_name}
- R5 = {cart_id, cart_amount, username}
- R6 = {cart_item_id, quantity, item_id, cart_id}
- R7 = {transaction_id, transaction_date, amount}

Step 5: Check for lossless condition

- Closure of any one of the candidate keys (customer_id, username, card_number, item_id, cart_id, cart_item_id, transaction_id) does not give all the attributes in the schema.
- **customer_id+** = {customer_id, name, ssn, address, email, salary, credit_score, phone_number, dob}
- **username+** = {username, customer_id, password}
- **card_number+** = {card_number, customer_id, expiration_date, cvv, credit_limit, card_balance, minimum_payment, reward_points, recommended_credit_limit, verification_reason, card_status, is_approved}
- **item_id+** = {item_id, item_name, cost}
- **cart_id+** = {cart_id, username, cart_amount}

- **cart_item_id+** = {cart_item_id, cart_id, item_id, quantity}
- **transaction_id+** = {transaction_id, username, amount, transaction_date}

Therefore, lossless condition is not satisfied.

So, a new table that contains the global key needs to be added: **R8 = {card_number, transaction_id, cart_item_id}**

Verification:

- {card_number}+ = { card_number, is_approved, card_status, verification_reason, recommended_credit_limit, reward_points, minimum_payment, card_balance, credit_limit, cvv, expiration_date, customer_id, dob, phone_number, credit_score, salary, email, address, ssn, name }
- {transaction_id}+ = { transaction_id, transaction_date, amount, username, password, customer_id, dob, phone_number, credit_score, salary, email, address, ssn, name }
- {cart_item_id}+ = { cart_item_id, quantity, item_id, cart_id, cart_amount, cost, item_name, username, password, customer_id, dob, phone_number, credit_score, salary, email, address, ssn, name }

RESULTING 3NF SCHEMA:

- **R1 = {customer_id, dob, phone_number, credit_score, salary, email, address, ssn, name}**
- **R2 = {username, password, customer_id}**
- **R3 = {card_number, is_approved, card_status, verification_reason, recommended_credit_limit, reward_points, minimum_payment, card_balance, credit_limit, cvv, expiration_date, customer_id}**
- **R4 = {item_id, cost, item_name}**
- **R5 = {cart_id, cart_amount, username}**
- **R6 = {cart_item_id, quantity, item_id, cart_id}**
- **R7 = {transaction_id, transaction_date, amount, username}**
- **R8 = {card_number, transaction_id, cart_item_id}**

Comparison between the 3 schemas

UML (Lossy)

- Composed of 7 tables
    - Customer, Inventory, CartItem, CustomerLogin, Cart, Card, Transaction

BCNF

- Decomposes into 8 tables
    - Customer, Inventory, CartItem, CustomerLogin, Cart, Card, Transaction, CardTransactionCart

3NF

- Decomposes into 8 tables
    - Customer, Inventory, CartItem, CustomerLogin, Cart, Card, Transaction, CardTransactionCart

Final Schema and Constraints

Core Tables:

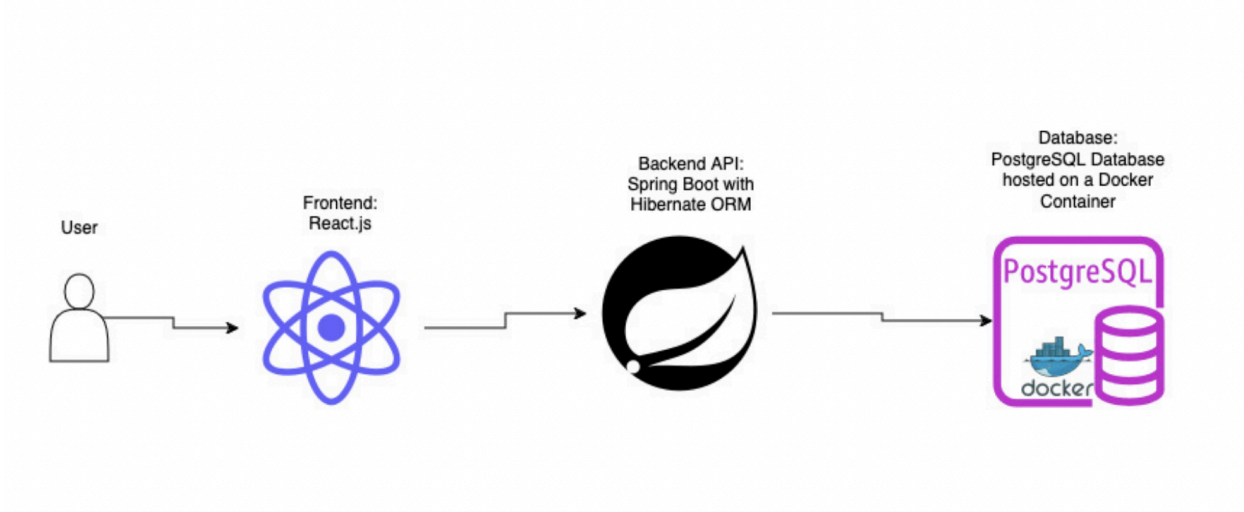
- **Customer**: Stores customer details and serves as the primary entity.
- **Card**: Tracks credit card information for each customer.
- **CustomerLogin**: Manages authentication for customers.
- **Inventory**: Stores items available for purchase.
- **Cart**: Tracks shopping carts for customers.
- **CartItem**: Links carts to individual inventory items.
- **Transactions**: Logs customer transactions.

Relationships:

- **Customer** ↔ **Card**: Each customer can own multiple cards.
- **Customer** ↔ **CustomerLogin**: Authentication is tied to the customer ID.
- **CustomerLogin** ↔ **Cart**: Each user has a unique cart.
- **Cart** ↔ **CartItem** ↔ **Inventory**: Carts contain multiple items from the inventory.
- **CustomerLogin** ↔ **Transactions**: Records user-specific purchases.

We used the ER Diagram provided above for our final schema .

Software Architecture



Explanation:

The user accesses the application through the react front-end which is being hosted on Port 3000 which calls a Java Spring Boot backend which has all the appropriate APIs on Port 8080. The Backend API makes transactions to create, read, delete and update information to a Postgres database which is being hosted on a Docker container and exposed at port 5432. Docker containers are being used to allow scaling in case the database gets overwhelmed.

Software Components

- Frontend:
  - Customer Sign up screen
  - Customer Login and Registration screen
  - Customer and Card Information Screen
  - Store Screen to show items available
  - Cart and checkout screen
- APIs:
  - Customer API - Ability to create, get, update, delete customer
  - Card API - Ability to generate credit card and make changes
  - Cart API - Ability to buy items and add to a cart
  - Transaction API - Ability to process cart transactions (Something like store checkout)
- Database

- Postgres Database with all the relevant tables
- Get updated based on how the endpoint makes transactions

Secure Logins

The secure login happens in the following way:
1. First we register the user by calling the register endpoint in the customer login API
2. The API then stores the login information to the customerlogin table. The password is hashed using the Bcrypt library which adds a salt to the hash to add extra security.
3. To log the user in, the login endpoint takes the username and password and verifies it against the database and once the passwords are matched it generates a JWT (Json Web Token) which needs to be passed as an Authorization header to all API requests and the APIs verifies the token and returns the appropriate response

End to End Execution and Database Transactions

We have attached a video demo .