Question 1:

1. **Frontend** is everything you see and can interact with using a browser. The front-end is built using a combination of technologies such as HTML, JavaScript and CSS
2. **Backend**, also called the server-side, consists of the server which provides data on request, the application that channels it, and the database which organizes the information.
3. A **web server** is a computer that stores web server software and a website's component files.
4. **Database** is an organized collection of structured information, or data (popular databases being used are MySQL, SQLite, Postgres, etc.).
5. **File system** handles the persistent storage of data files, apps, and the files associated with the operating system itself.
6. **User Story** is an informal, general explanation of a software feature written from the perspective of the end user or customer.
7. **Git** is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development.
8. **Docker** is a platform for developing, shipping, and running applications inside containers.
9. **Ruby** is an open-source, object-oriented, interpreted, and high-level programming language.
10. **Rails** is a full-stack framework which ships with all the tools needed to build amazing web apps on both the frontend and the backend.

Question 2:

1. **Application topic**: Online Chocolate Shop Management System
2. **User stories:**

- As a customer, I want to read reviews and rates of a product so that I can decide which one to buy.
  Acceptance criteria:
  + Reviews of a product should be displayed below the details of that product on its own page.
  + Average rate of a product should be displayed next to its overall information (name, price) in the product page (page that lists all products).
  + Only customers who bought the product can leave a comment and rate that product.
  + Customer can leave only 1 comment for each product in 1 order.

- As a customer, I want to see my historical orders so that I can decide what to buy easier in the future.
  Acceptance criteria:
  + Data to be displayed: product name, product price, quantity, comment, rate, order date.
  + Data should be ordered by order date.

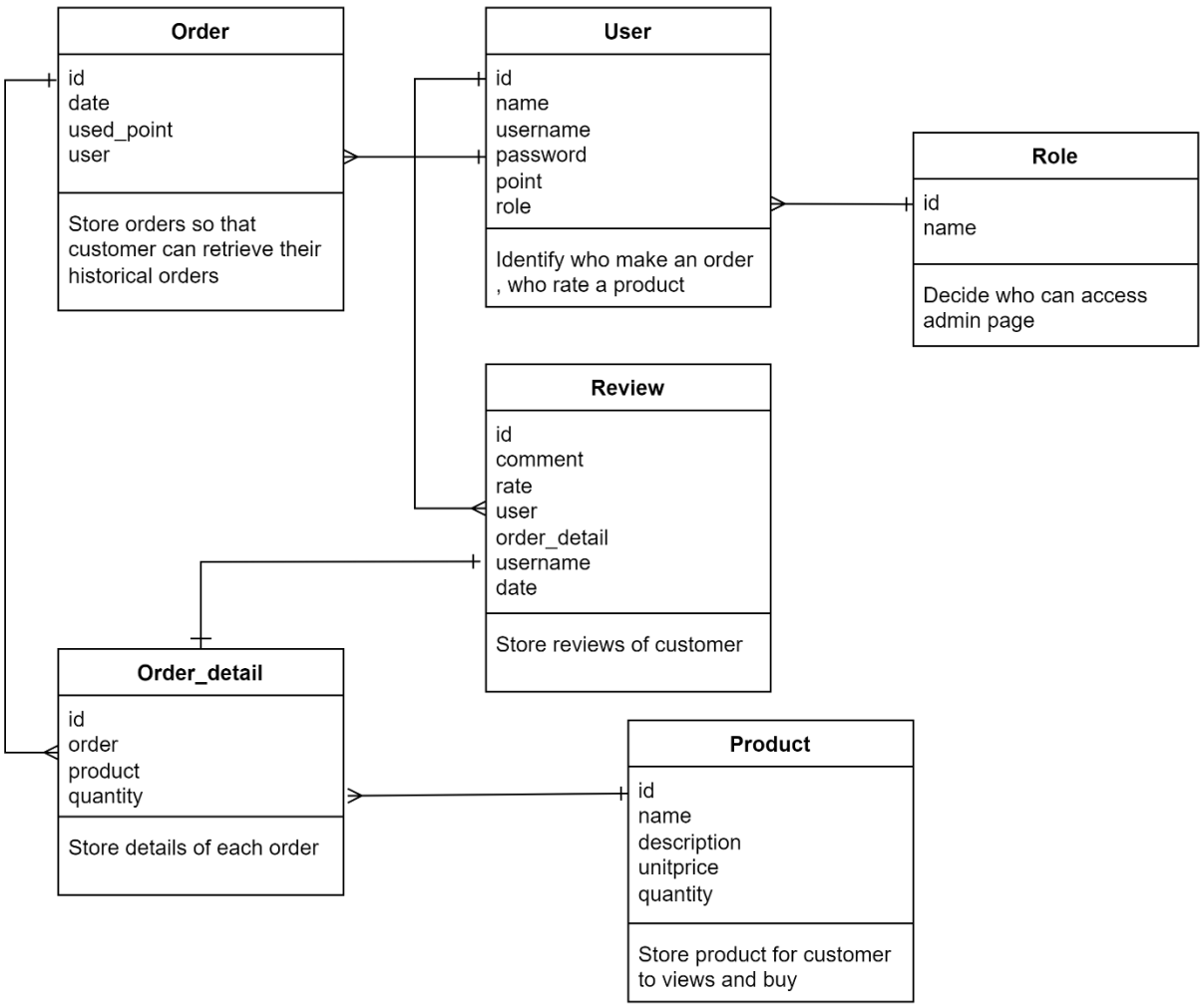
- As a customer, I want to search for products by multiple filters so that I can find the product more quickly.
  Acceptance criteria:

+ Search by product name: can search by key words (display all the product whose name contains these key words).
+ Search by product rating: display all the products whose average rating is larger or equal to the number that customers input.
+ Search by price: display all the products price is in the range that customers input.

- As the owner of my shop, I want to have an admin page so that I can add new products and update existing products without the support of the IT team.
  Acceptance criteria:
  + Data to be inserted for new products: product name, unit price, description, quantity.
  + Data can be updated for existing products: unit price, description, quantity.
  + There should be automative notice when I input unreasonable data (for example, un-unique product name, non-numeric for quantity)
  + Only the owner can access this page.

- As the owner of my shop, I want to give my customers points so that I can encourage them to buy my products again.
  Acceptance criteria:
  + Points are given to customers only when they have made a purchase (points can be calculated as 1% of the value of an order).
  + Points can be bonused if customers leave a review and rate after making a purchase.
  + Points can be converted to money to decrease the payment customers need to pay in the next order.

3. **Database Schema:**


Order
id
date
used_point
user
Store orders so that customer can retrieve their historical orders
User
id
name
username
password
point
role
Identify who make an order , who rate a product
Role
id
name
Decide who can access admin page
Review
id
comment
rate
user
order_detail
username
date
Store reviews of customer
Order_detail
id
order
product
quantity
Store details of each order
Product
id
name
description
unitprice
quantity
Store product for customer to views and buy