

This assignment is the next step of the assignment on use case and data analysis.

Task 1. Draw a class diagram for the Store Management System containing major classes.

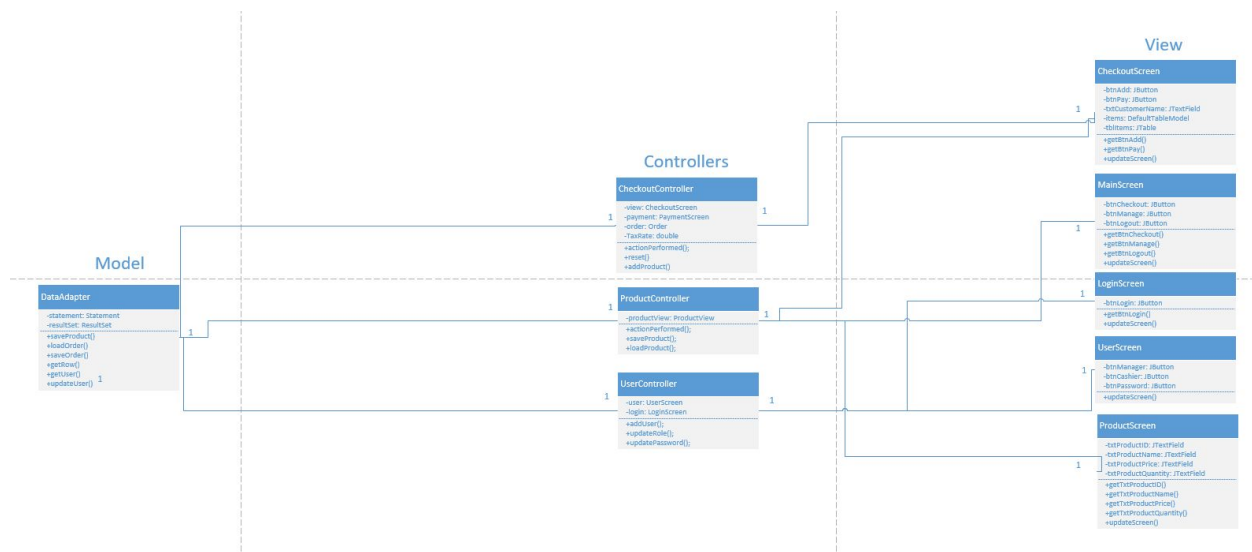
The classes should be designed strictly with MVC pattern:

- A Model class only contains data
- A View class only contains GUI components for displaying
- Application/business logic is implemented only in Controller classes.

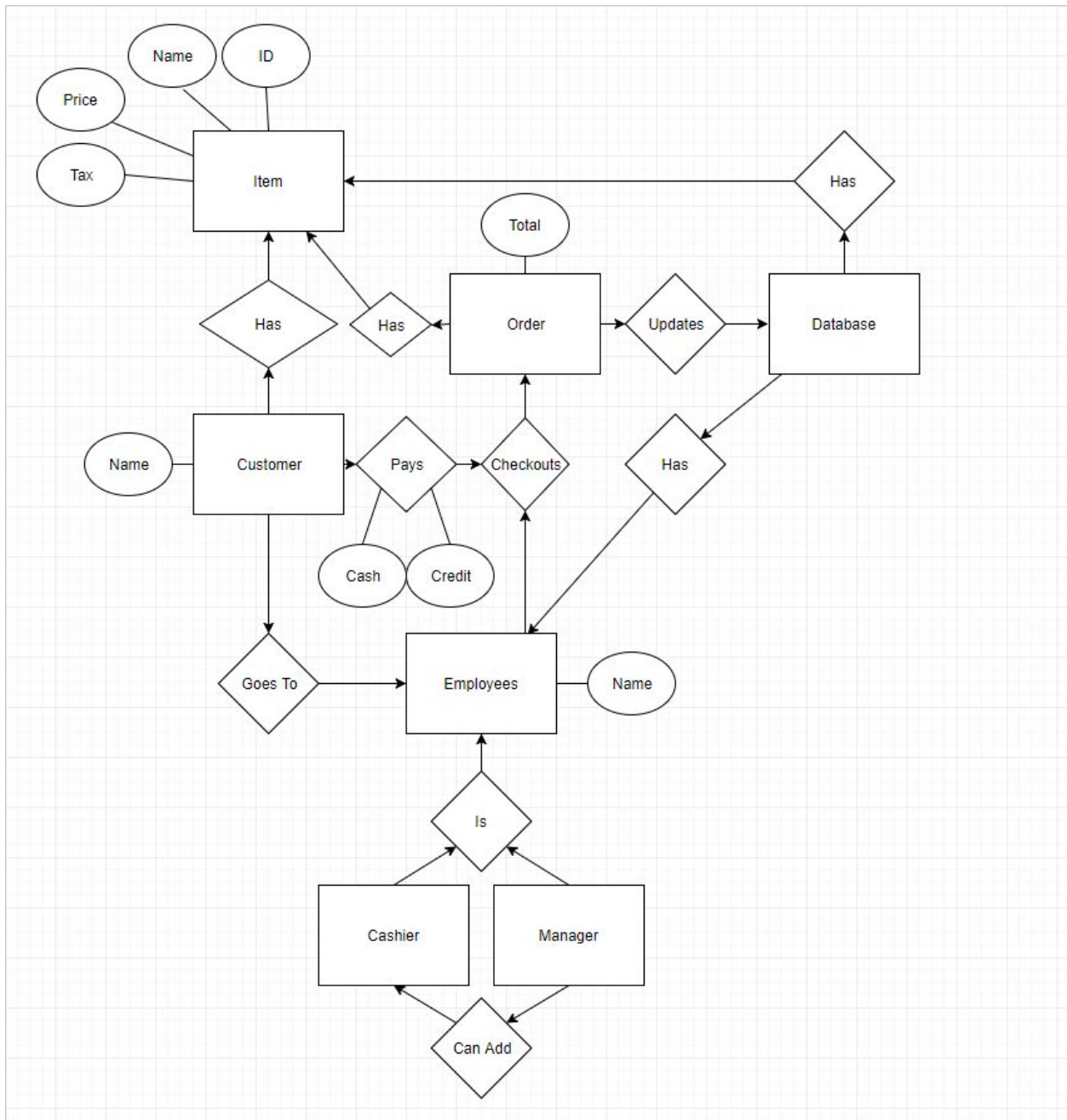
Task 2. Draw an entity-relationship diagram for the system.

Task 3. Write SQL code to implement the ER diagram in Task 2 in a database and insert the data collected from Iteration 1.

## 1. Task #1



## 2. Task #2



### 3. Task #3

- a. Creating Database
  - i. CREATE DATABASE Store;
  - ii. CREATE TABLE Inventory(Product\_ID int, Product\_Name text, Price decimal, Tax decimal, PRIMARY KEY(PRODUCT\_ID));
  - iii. CREATE TABLE Orders(Order\_ID int, Total decimal, Customer\_name text, Payment\_type text, PRIMARY KEY(Order\_ID));
  - iv. CREATE TABLE Users(Name text, Role Text);
- b. Inventory
  - i. INSERT INTO Inventory VALUES (101, "Bunch of Bananas", 3.58, .08, 122);
  - ii. INSERT INTO Inventory VALUES (102, "Dane's Bread", 2.95, .08, 45);
  - iii. INSERT INTO Inventory VALUES (112, "Eggs", 2.62, .08, 176);
  - iv. INSERT INTO Inventory VALUES (132, "Milk", 3.10, .08, 39);
  - v. INSERT INTO Inventory VALUES (195, "Pop Tarts", 2.27, .08, 54);
- c. Orders
  - i. INSERT INTO Orders VALUES (12312, 49.93, "Katie", "Cash");
  - ii. INSERT INTO Orders VALUES (14145, 27.95, "Jack", "Cash");
  - iii. INSERT INTO Orders VALUES (56754, 13.62, "Bill", "Credit");
  - iv. INSERT INTO Orders VALUES (23633, 67.10, "John", "Debit");
  - v. INSERT INTO Orders VALUES (23465, 12.27, "Smith", "Cash");