

Design of Database

Summary of Grocery On Wheels:

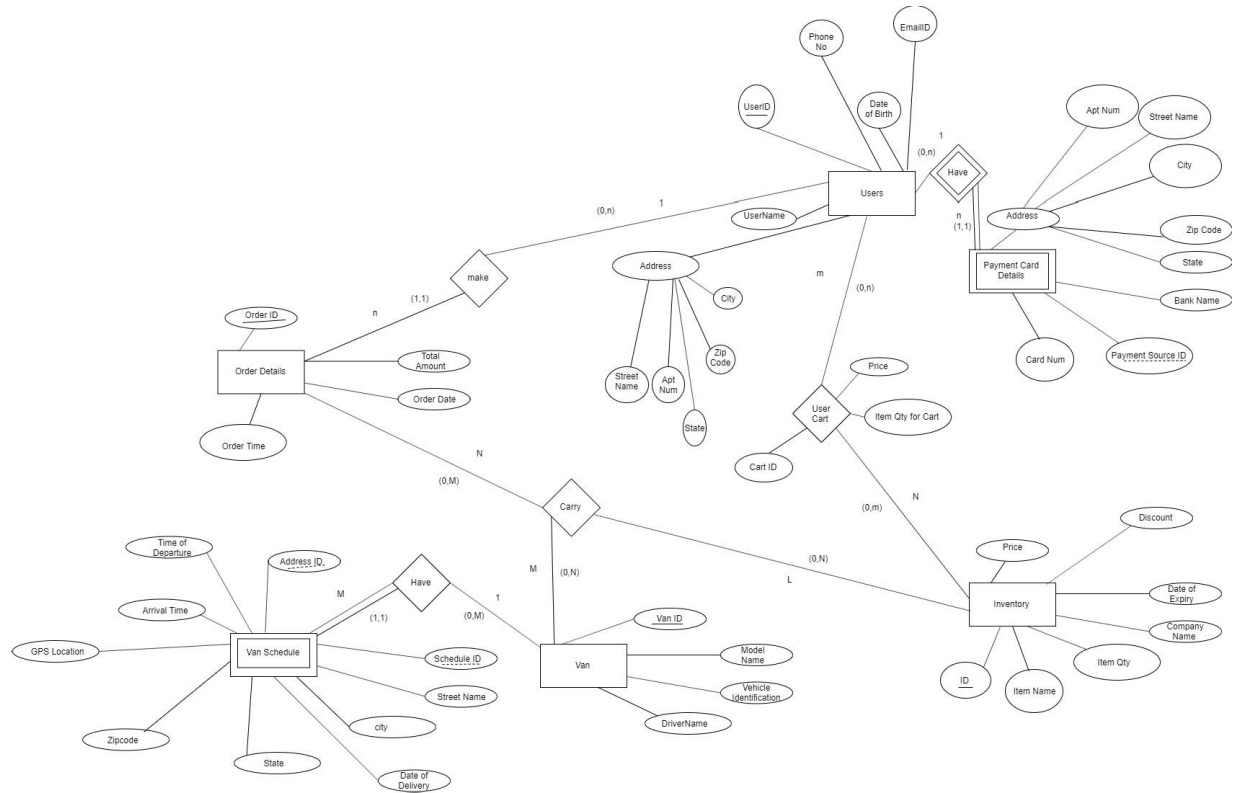
The project is to develop a terminal application using Java for '**Grocery On Wheels**'. The purpose of this application is to make grocery available at the doorsteps of customers. A grocery van makes daily visits to a fixed location (like the corner of E.Main Street and Lincoln Drive) in different neighborhoods. Once it reaches the nearest location of buyers, they can go to the van and buy their groceries and vegetables.

Instead, customers can also send the required items list to the seller so that the seller/driver of the van can keep the foodstuff ready to pick before hitting one's nearest location.

This application enables the customers to search for any product and add to the cart and pay them using the saved card details. It is developed solely to focus on food products available to common people so that people should not bother about going to the store and buying especially in unsuitable weather conditions."

We need database, because the project contains abundant of information regarding Users, Payment, Stock of Products, Orders , Van and its schedule which has to be stored in DB to keep the track of user activities, products in inventory and the visits of vans. So, DB is an important part of the system and one cannot go ahead with the project without saving the details of their software components in DB.

ER Diagram:



ER Schema:

Users - (UserID: Integer, UserName: Varchar(25), AptNo : Varchar (10), StreetName : Varchar(20),City: Varchar(25),State: Varchar(2), ZipCode: Integer,DateOfBirth : Date,PhoneNo:"String",EmailID:"String")

Relation in ER Diagram :‘HAVE’ - User ‘have’ PaymentSource

PaymentCardDetails - (UserID: Integer,PaymentSourceID: Integer, BankName :Varchar(25), CardNum: Varchar(20), City: Varchar(25),State: Varchar(2), ZipCode: Integer,AptNum:"Integer",Street Name:"String")

Relation in ER Diagram :Inventory uses user cart and users uses user cart

UserCart-(UserID: Integer, CartID: Integer, InventoryID: Integer ,ItemQtyforCart: Integer, Price: Integer)

Relation in ER Diagram : ‘HAVE’ - Van and Van Schedule related by ‘HAVE’

Van- (VanID: Integer, ModelName : Varchar(25), VehicleIdentification :Varchar(25),DriverName : Varchar(25))

VanSchedule- (VanID: Integer,ScheduleID: Integer,AddressID: Integer, StreetName : Varchar(20),City: Varchar(25),State: Varchar(2), ZipCode: Integer, GPSLocation : Varchar(20) , ArrivalTime:TIMESTAMP, Date of Delivery :DATE,TimeofDeparture: TIMESTAMP)

Relation in ER Diagram : ‘CARRY’ - Van, Order details and Inventory are related by ‘CARRY’

Carry- (VanID: Integer, OrderID: Integer,InventoryID:Integer)

Inventory - (ID: Integer, ItemName: Varchar(25),ItemQty: Integer, CompanyName: Varchar(25),Price: Integer, Date_of_Expiry : Date,Discount: Integer)

Order Details - (UserID: Integer,OrderID: Integer, Total Amount : Integer, OrderDate : Date, OrderTime : TIMESTAMP,

Relation in ER Diagram : ‘MAKE’ - User ‘make’ order

Order Details - (UserID: Integer,OrderID: Integer, Total Amount : Integer, OrderDate : Date, OrderTime : TIMESTAMP,

SQL Queries:

1. All the items that has use by date yyyy/mm/dd - 1 table
2. Products that are manufactured by a particular company and count of that items-1 table
3. List of user names who have ordered groceries more than 150 \$ - 2 tables
4. Group the members by their age and find the count of each age group. 1 table
5. Check whether an item is available or not in the inventory -1 table
6. List of Orders a van is carrying on 'Nov 1, 2019' at scheduled arrival time between 2:00 PM and 8:00 PM - 2 tables
7. Group the number of orders by each bank and find the count - 3 tables
8. Print the name of drivers who are working on 'Nov 1 2019' - 2 tables
9. List of users and email ids who have ordered 'Milk' - 4 tables
10. Count of users who have ordered ' Milk' and 'Beer' in the same order - 3 tables