GROCERY MANAGEMENT SYSTEM

Problem Statement: Build a Model of Grocery Management chain where enormous amounts of data are collected and that need to be stored efficiently and suitable frontend for the ease of use.

Objective and approach:
The grocery store industry is a multi-billion dollar industry. It touches everyone in society since we all eat. Through tough competition a grocery chain must run efficiently and reduce cost overruns. This includes using updated information in its purchasing decisions, inventory control, store stocking, customer satisfaction, buying trends, and a host of other business concerns.

The business of grocery stores is the ability to provide their customers with well priced and fresh food items everyday. This seemingly simple concept is very complex when put into practice.

This project aims to make a model of Grocery Stores and the model will have multiple stores. Each store will have a record in the 'store' entity including a unique ID and address. Inventory will have a store ID component plus item information. This will allow a manager to see what items a store has, the quantity, and the current dollar value of items in inventory.

Requirements :

Entities and Constraints

- ❖ **CUSTOMER:** This entity type represents all the people that shop at the grocery store. A customer performs a checkout. The CUSTOMER entity relates to the CHECKOUT table via the BUY ITEM relationship. The Cust_ID primary key is a foreign key in the checkout table.

- ❖ **EMPLOYEES:** This entity type represents all the people that work for the grocery chain. It is a superclass because both managers and salaried people are in this entity type. It is related to the STORE entity with the WORKS FOR relationship. It has a recursive relationship with SUPERVISING as employees manage themselves. Stores are managed by employees via the MANAGER relationship. Employees can have dependents through the DEPENDENTS OF relationship.

- ❖ **DEPENDENTS:** This is a weak entity type representing anyone who is a dependent of an EMPLOYEE via the DEPENDENTS OF relationship.

- ❖ **CHECKOUT:** This entity type represents an atomic transaction of a customer purchasing items in the store. It relates to STORE via the CHECKOUT LOCATION relationship. It relates to CUSTOMERS via the BUY ITEMS relationship. It is connected to the ITEMS entity through the CHECKING OUT relationship. This relationship will become a table using the PK from CHECKOUT and ITEMS to join every item on each individual checkout transaction. CHECKOUT needs a ‘total cost’ entity as it is calculated at the time of purchase with those specific item prices. If we try to derive this number later any price change would also change the total cost.

- ❖ **ITEMS:** This entity type represents the individual store items someone would purchase like milk, cheese, meat, etc. ITEMS is related to CHECKOUT as described above.

- **STORE:** This entity type represents the actual brick and mortar buildings where food is placed and customers go to and buy. It has two relationships with EMPLOYEES. One is WORKS FOR and that relationship describes which EMPLOYEES are at which STORE. The second relationship is MANAGES FOR. This connects which manager supervises which store. The store_id primary key is useful as foreign key in multiple tables.

- **FEEDBACK:** This weak entity represents feedback given by the customer and has attributes rating ,feedback_id,cust_id,item_id. ‘feedback_id’ acts as the primary key in this table and foreign keys are ‘cust_id’ and ‘item_id’. This entity is related to the customer by “PROVIDES” relation.

- **CANCELLATION:**Cancellation entity is related to check out entity by “UNDERGOES’ relationship.This is the entity related to the details of the item that are cancelled at the time of checkout. Canellation_id is the primary key and item_id is the foreign key.

- **SUPPLIER:**Suppliers supply items to the store by ‘SUPPLIES’ relationship. N number of suppliers supply to M number of stores. ‘supplier_id’ is the Primary key.

Data Requirements:

External Inputs:

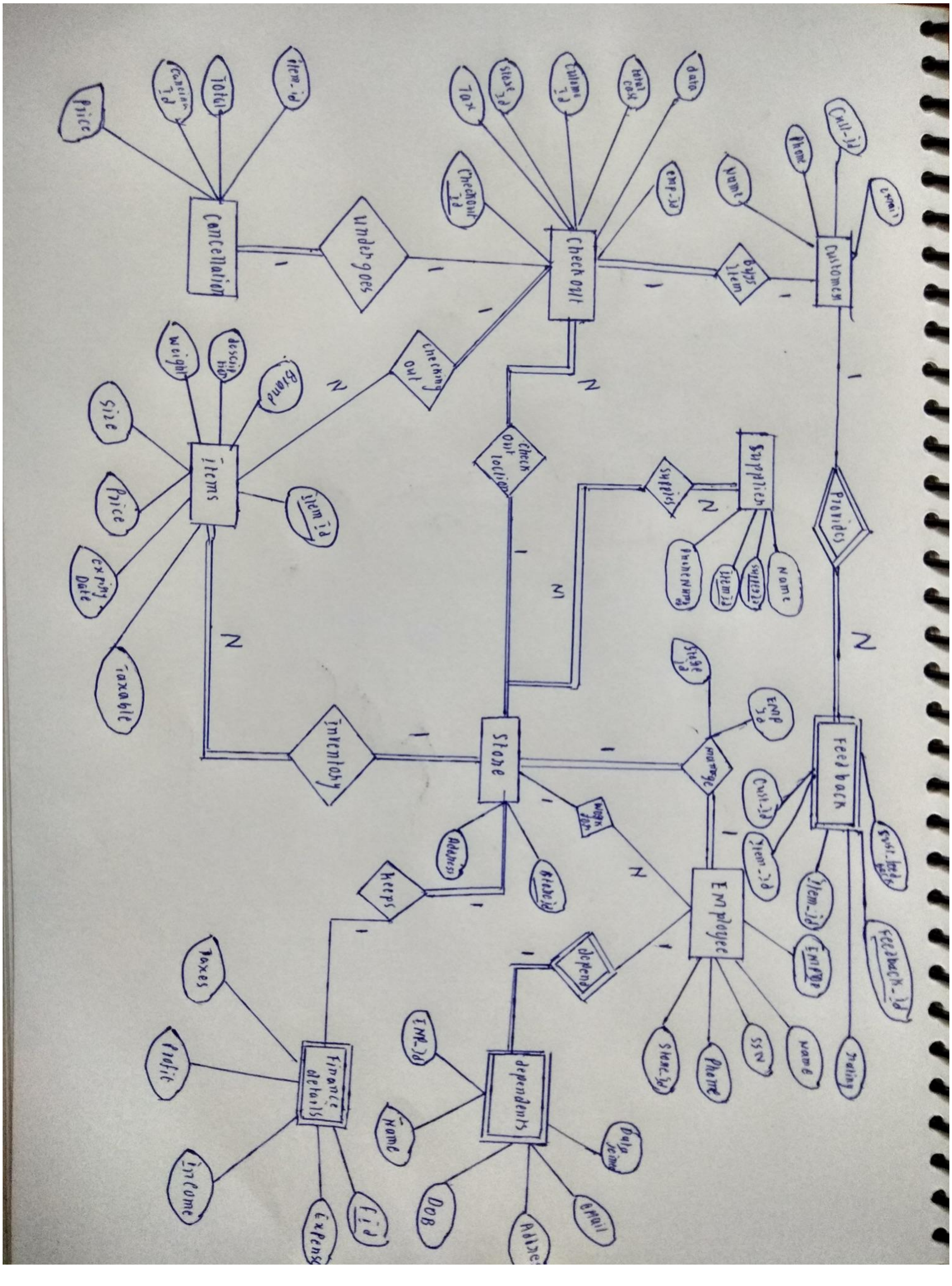
- Customer
  - Email
  - Phone
  - Name
- Supplier
  - Name
  - Phone number
- Employee
  - Name
  - Phone
    - Dependents
      - Name
      - Date Joined
      - Date of Birth
      - Address
      - Email
- Store
  - Store Address
- Feedback
  - Rating
  - Feedback

Other attributes like ids are auto generated by the system.

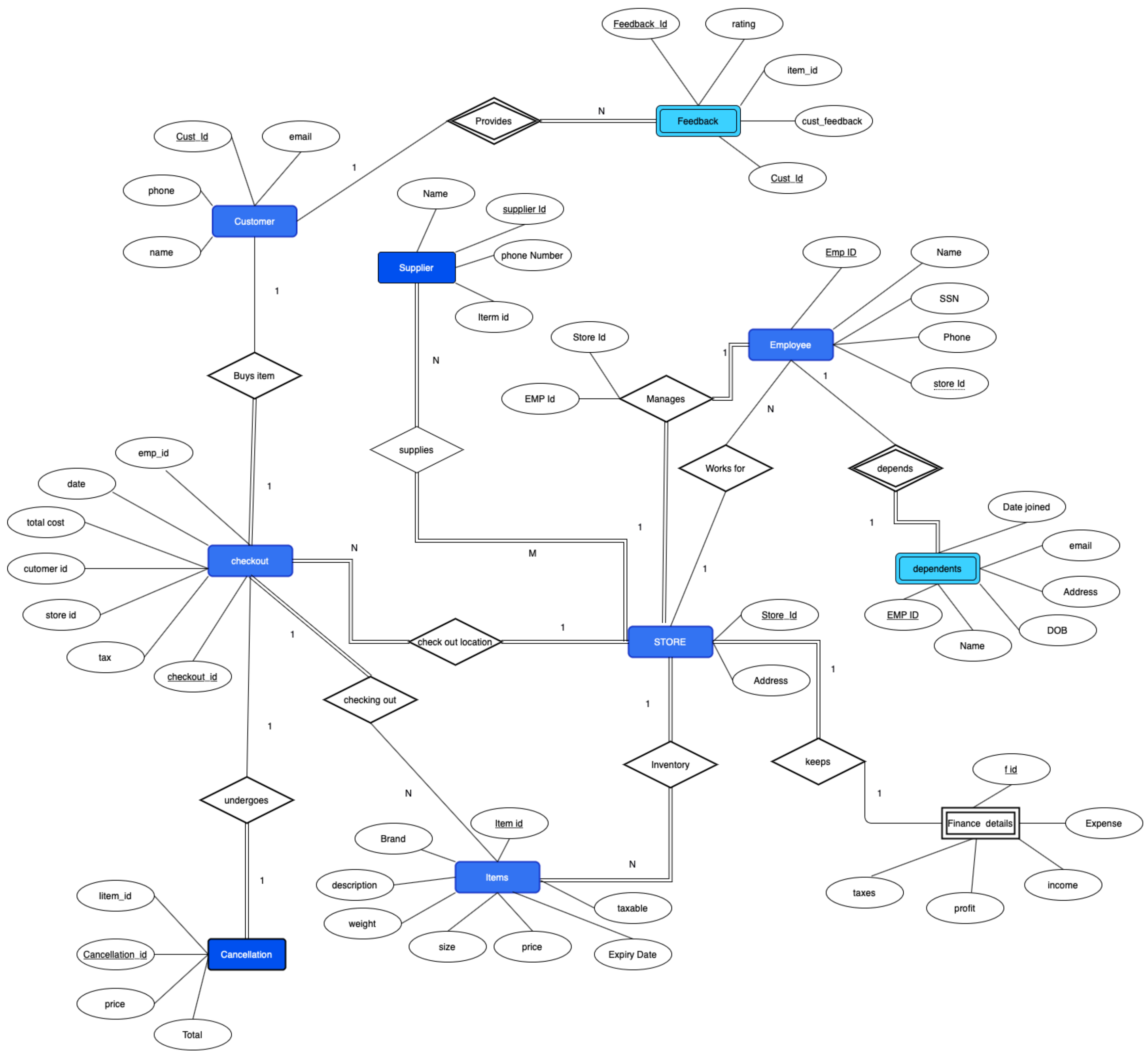
Other Specifications:

- Manager with unique employee_id must add products on the site for the customers to buy
- Manager has login account
- Customer has no login account
- Customer can choose the product of his/her choice
- item details : defines the product
- Items are identified by unique item-id
- N number of suppliers supply to M number of stores
- 1 Customer can give feedback on N number of items (optional)

ER DIAGRAM PEN SKETCH



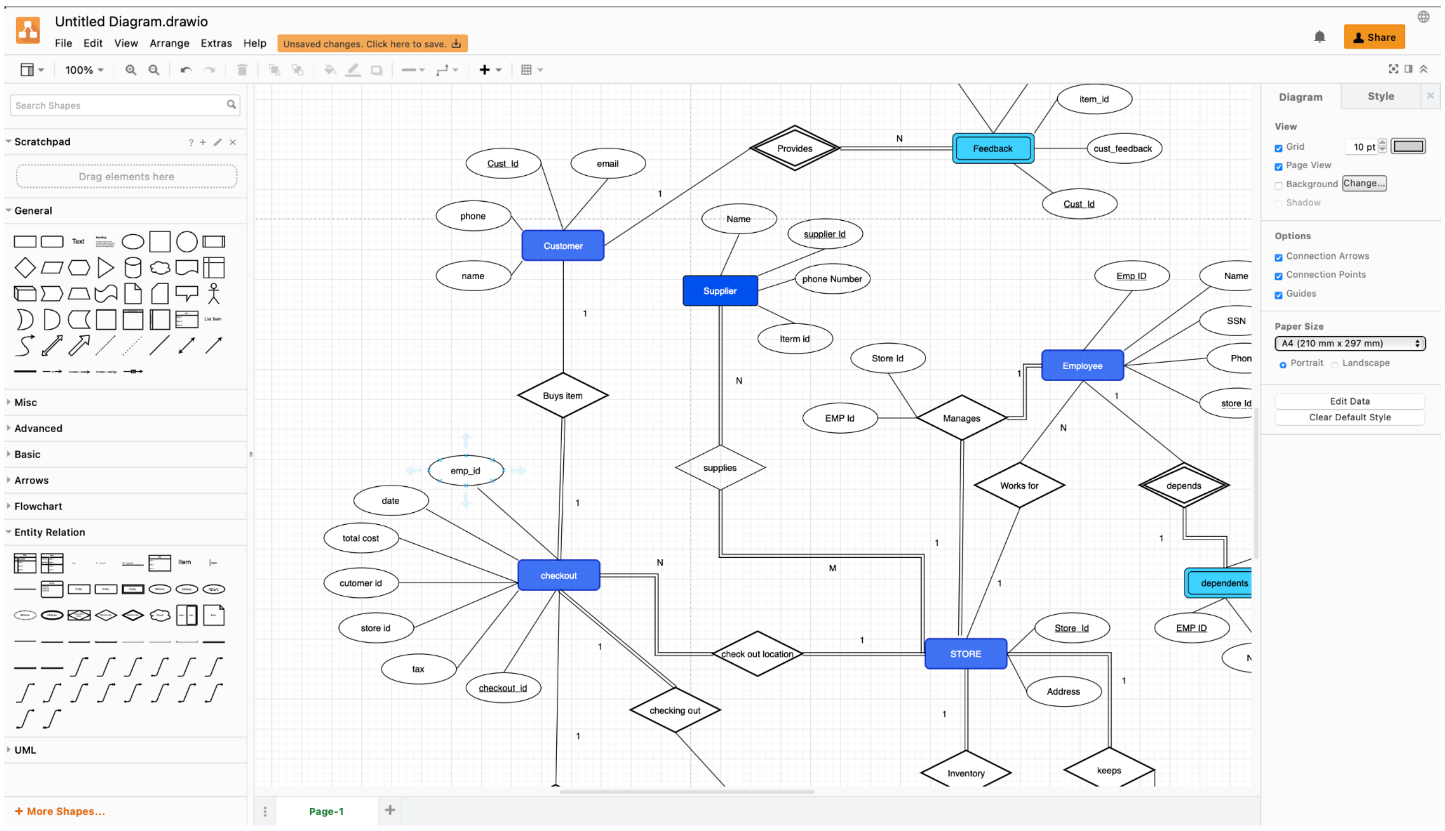
ER DIAGRAM SOFTWARE BASED



ER Diagram represents the relationship of entity sets stored in the database.This visual representation helps in understanding the logical structure of the database. In the above diagram we have entities like store,supplier,customer,checkout,cancellation,Employee and dependents. These entities are represented as rectangle blocks.Some entities are related to other entities and relationships are represented by the rhombus shape. Participation can be total or partial and they are represented by double and single lines respectively.

Tool Used: Draw.io
draw.io automatic layouts allow users to quickly rearrange our diagrams on the draw.io canvas. Depending upon the layout we choose, our diagram can be reorganized following the rules and spacing information that are appropriate for our needs.

Layout of the software:



The shapes required by our ER diagrams can be easily used by ‘drag and drop’ method or selecting them individually and there are options to change the style of shapes and text.

Reason to use this tool:
Draw.io is a tool which we can learn easily by just using them couple of times and this software is available as both web app and downloadable software

Link: https://drawio-app.com

Member contribution and Total Time Spent

PES1UG19CS174	Guruprasad N B	Prepared the final copy of ER diagram on software and Identified all the relations between entities and added cardinality and other important things
PES1UG19CS176	H V Sashikant Reddy	Sketched ER diagram roughly on paper and finalized the ER diagram.(Needs many attempt to finalize and make it correct)
PES1UG19CS191	Ishwar Joshi	Defined and analyzed the problem statement, Identified the Entities and requirements for the project(Entities,attributes, and software tools) and made the final report.