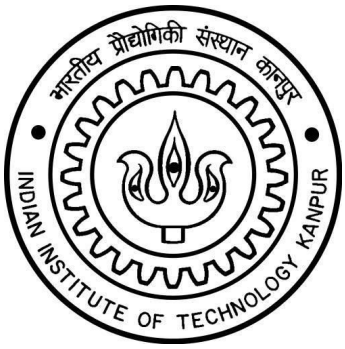


Sai Bhandar

MBA646

Course Project

November 27, 2020

Table of Contents

Salient Features

1. **Deals in non-perishable items**: For the sake of convenience, we will be considering only the non-perishable items.

2. **Variety of products**: The shop is a kind of One Stop Store for retailers and the common public. Some varieties of products are listed below, along with the number of products in that category.

Product Category	No. of products	No. of Brands	Different Sizes of Packaging	Examples
Regular Kitchen Stuffs	50	atleast 5	Small, Medium, Large	Rice, Daal, Besan, Sooji,Tea, Coffee etc
Oils & Refines	4	atleast 5	Small, Medium, Large	Mustard Oil, Soyabean Oil, etc
Dry Fruits	6	atleast 5	Small, Medium, Large	Cashew, Almonds, Resins, etc
Stationery Items	10	atleast 3		Pencils, Pens, Notebooks, etc
Grooming Category	15	atleast 5	Small, Medium, Large	
Snacks	15	atleast 5	Small, Medium, Large	Biscuits, Namkeens etc
Pooja Items	10	atleast 2	Small, Medium, Large	Agarbattis, Dhoop etc

3. **Number of employees**: 8

4. **Sales**: The estimated sale of the shop is approximately INR 2 lakhs/day

5. **Location**: The shop is one of the largest grocery shops in the town. It is located in the building having a godown of dimensions 40 ft * 20 ft, shop on the first floor with dimensions 15 ft * 10 ft, and a spare godown on the second floor with dimensions 15 ft * 10ft.

6. **Mode of Operation**: The transactions take place only in Cash, and the online mode of the payment is not available.

7. **Credit System**: The credit system is available only to the Retailers with an upper cap equivalent to the highest bill he/she has cleared, and has no previous debts.

Challenges to be addressed

1. **Inventory Management**: It is defined as the process of ordering, storing, and using the shop's inventory. As the shop is one of the largest groceries in the town, it must maintain the inventory list to avoid out of stock situations. Moreover, this would also reduce the losses incurred through employee theft, expiration of products, human error, and paperwork snafus. Furthermore, it would also give an idea of the market scenario and plan things accordingly.

2. **Customer Management**: The shop deals with two types of customers, i.e., Retailers (Businessmen) and the common public. It is incredibly crucial to segregate customers to decide the pricing, discounts, and other credit benefits. It also helps to understand the general customer sentiments and forecast future demand and sales. Also, it aids in extending credits and forging long-term relationships to gain the customer base.

3. **Supplier Management**: As the grocery segment deals with a plethora of products from different brands, it becomes quite evident to maintain records of the vendors. It gives an idea of the demand and supply of the various brands amongst the consumers, which proves to be a decision point to extend the product line and sizes.

4. **Store Operations**: The main operation of the shop is the transactions made in the store. As the employees in the store are not very skilled or literate, it is imperative to arrange the products in a manner easily accessible and traceable to by the employee in the store. It will improve the customer experience by reducing the wait time. Moreover, it is also significant to place the items correctly, i.e., the art of shelving. For instance, the products with high fragrance need to be alienated from the common convenience goods like jaggery, pulses, etc.

5. **Financial Management**: The intricacies and nuances of all the operations, the inflow, and the outflow of the capital need to be managed. The regular business sale and purchase, credit lines, and salaries of the employees are the major areas of management. Moreover, there are two types of customers and different suppliers, which require keeping track of everyday operations and credit lines. A sound

monetary idea would be beneficial in making purchase decisions and expansion policies within a product category or business operations.

RDBMS: Selection and Justification

Based on the above salient features and the challenges to be addressed, our database requirements are narrowed down to these **3 RDBMS solutions**:

1. MySQL
2. PostgreSQL
3. MariaDB

1. **MySQL**

- MySQL is one of the most popular and easy to use free RDBMS solutions available.
- MySQL supports a vast amount of embedded applications and operating systems, making it highly flexible and compatible.
- Through a big number of benchmark tests, the MySQL community suggests it to be the one of the fastest database solutions.
- MySQL's best feature tends to be the fact that it can handle a very huge amount of data, more than 50 million rows of data. It does this by supporting multi-threading.

2. **PostgreSQL**

- The central algorithm of PostgreSQL is very efficient so it outperforms many other database systems. This becomes a major factor when working on large datasets since in such cases I/O processes become a bottleneck.
- It is one of the most flexible open source databases around. Functions can be written in a wide range of languages such as Python, Perl, Java, Ruby, C and R.
- It has high community support.

- It is less vulnerable to data corruption since it protects data integrity at the transaction level.

3. **MariaDB**

- MariaDB is a free and open source distribution of MySQL, released under the GNU General Public License.
- MariaDB is a lightweight database software that puts minimal strain on technical resources, it was created as a concern of core developers of MySQL that Oracle might erode the base of its open source philosophy.
- MariaDB has frequent security releases and is compatible with many languages like Ruby and others that are commonly used in MySQL.
- It is highly compatible with MySQL so migrating from one system to another is extremely fast and convenient.

Comparison of the 3 RDBMS solutions yields us the following learnings:

❖ **MariaDB vs PostgreSQL**

1. **Performance** :-

   PostgreSQL handles the read and write queries in a much faster time than MariaDB for the turnaround time parameter. Due to the quick data access, PostgreSQL is often preferred over MariaDB especially when the data to work with has a lot of tables in need of concurrent updation of values.

2. **Data type handling** :-

   MariaDB handles mismatch of data types by giving an alert and automatically fixes the data type whereas PostgreSQL does not make such changes and throws an error in case of type mismatch until the type is same as the data type of the object making PostgreSQL better at handling

data integrity which helps the programmer to debug the code when the user has entered a wrong data type as they can directly go to the line throwing the error which is not shown in MariaDB.

3. **Features** :-

➔ PostgreSQL has an optimized database performance because of materialised views and partial indexes helping PostgreSQL to perform quicker database write queries which is one of the most important requirements for our setup.

➔ MariaDB has a better support for multiple languages compared to PostgreSQL. If our team was to work with Ruby then PostgreSQL won't be an option as it does not support Ruby.

➔ MariaDB does not support many server operating systems like OS X. In case we have to work with OS X based servers we cannot use MariaDB.

➔ MariaDB has smaller size as compared to PostgreSQL. MariaDB is lighter, if we were to work with a light database and fairly non-complex update queries then MariaDB would be preferred.

❖ **PostgreSQL vs MySQL**

1. **Performance** :-

PostgreSQL is faster for concurrent read-write operations because it handles concurrency better as compared to MySQL. Some of the reasons are as follows:

- PostgreSQL implements Multiversion Concurrency Control (MVCC) without read locks.
- It supports parallel queries that can use multiple CPUs/cores.
- It can create indexes in a non-blocking way.

- It can create partial indexes.

2. **Data Type Handling** :-

   Custom data types, operators and index types can be added in PostgreSQL but not in MySQL. PostgreSQL is also highly extensible, it supports many advanced data types that are not available in MySQL such as:

   - Geometric Information
   - Network Address Type, Native UUID (universally unique identifier)
   - JSONB (JavaScript Object Notation in Binary)

3. **Features** :-

   ➔ MySQL is a purely relational database whereas PostgreSQL is an object-relational database, hence PostgreSQL includes features like table inheritance and function overloading.

   ➔ FULL OUTER JOIN, INTERSECT, EXCEPT, partial indexes, bitmap indexes, expression indexes table inheritance are supported in PostgreSQL and not in MySQL.

   ➔ MySQL works on multiple storage engines while PostgreSQL works on a single storage engine.

   ➔ PostgreSQL supports table cascading and truncating and an IP address data type whereas MySQL does not.

❖ **Popularity** and **Documentation** comparison among the 3 solutions:

❏ MySQL is the most popular among the three, followed by PostgreSQL and MariaDB is least popular. Hence, the number of third party tools, or developers/database administrators available for PostgreSQL and MariaDB is smaller as compared to MySQL.

- ❏ The documentation for PostgreSQL is less as compared to MySQL but overall, there is sufficient data about PostgreSQL to learn, implement and experiment with PostgreSQL.

★ We **Recommend PostgreSQL** based on the findings we got on analysing and comparing the 3 RDBMS:

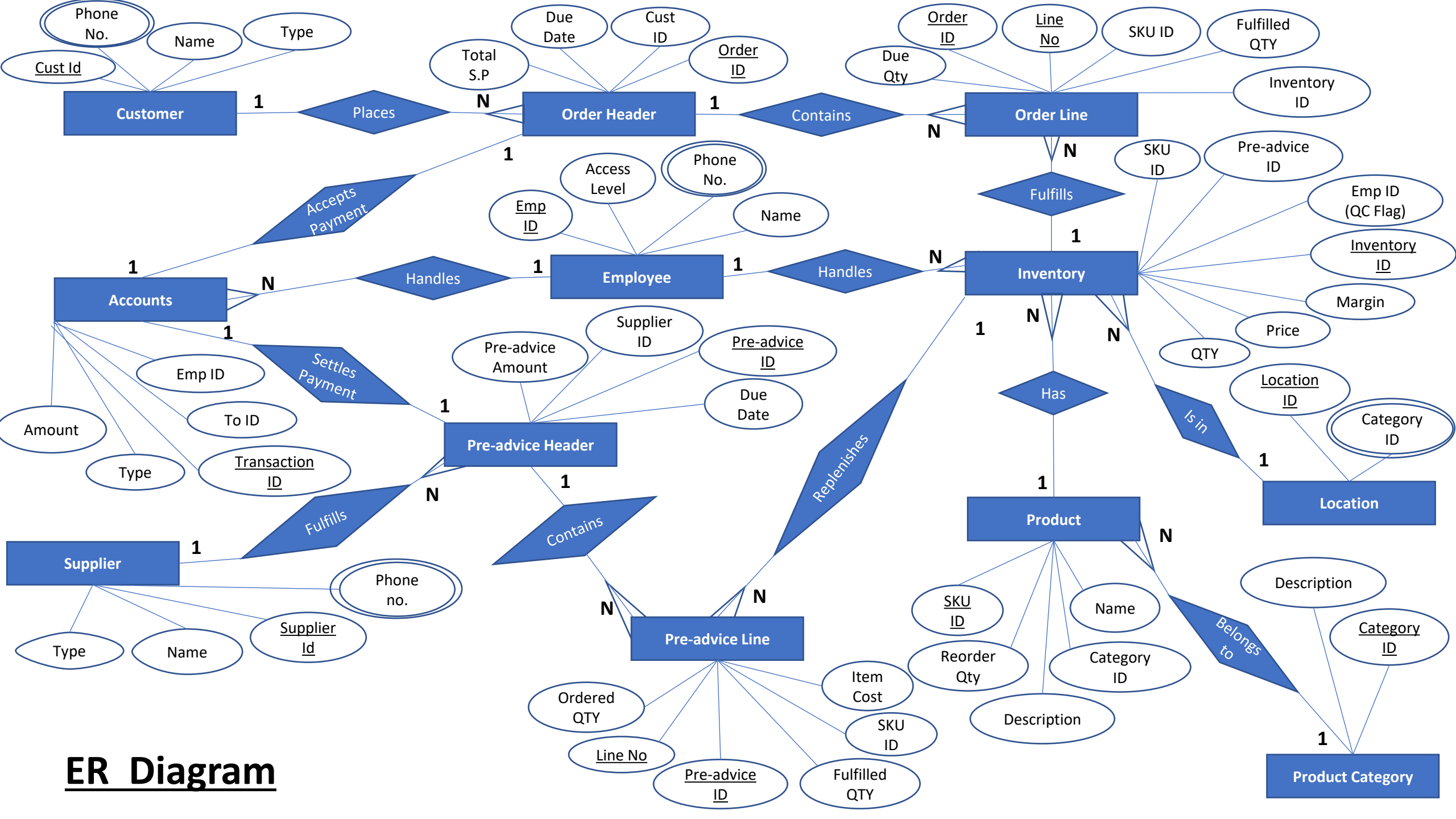
- PostgreSQL is faster for concurrent read-write operations
- Less vulnerable to data corruption since it protects data integrity at the transaction level.
- Supports many additional programming features, making the queries much simpler to write
- Easier debugging for errors due to strict data type handling
- Supports many operating system servers
- Free and open source along with high community support

❖ Additional Reasons why PostgreSQL is better:

PostgreSQL supports JSON and **JSONB**, unlike MySQL or MariaDB. JSONB stands for JSON-Binary (JSON - JavaScript Object Notation)

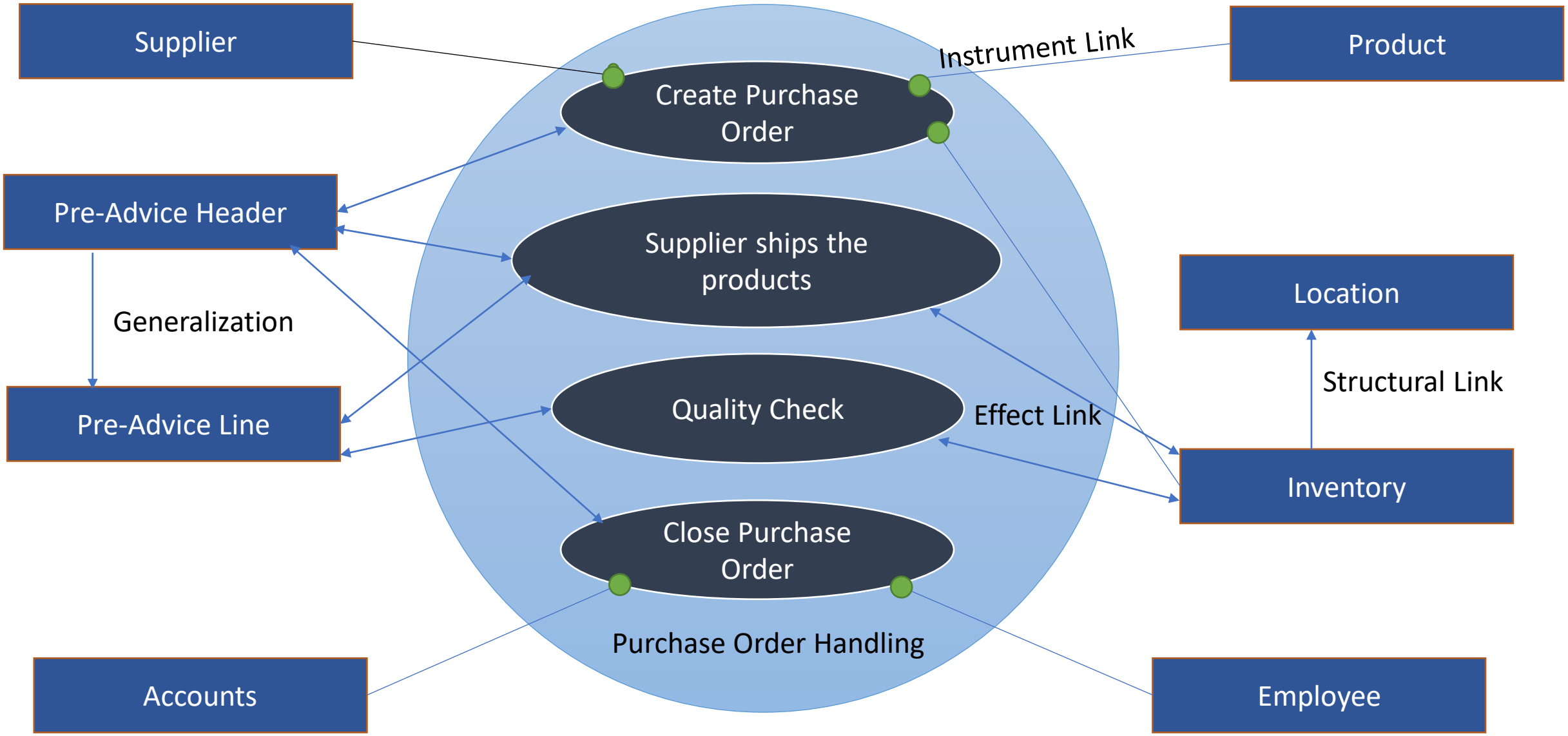
- The JSON data is stored as text - ASCII/UTF-8 string, whereas JSONB stores data in a decomposed binary form, as binary code.
- Benefits of JSONB over simple JSON data:
  - More efficient and faster to process
  - Supports indexing and has simpler schema designs (Replacing Entity-Attribute-Value (EAV) Tables with JSONB columns, which can be queried, indexed and joined, allowing for significant performance improvements)
- Drawbacks:
  - Extra disk space, larger table footprint
- So overall, one can see the JSONB has benefits and drawbacks over just JSON, which make the performance metric uncertain, but for

our model which isn’t large scale and will have simpler operations, JSONB will be very fruitful and improve the database performance.

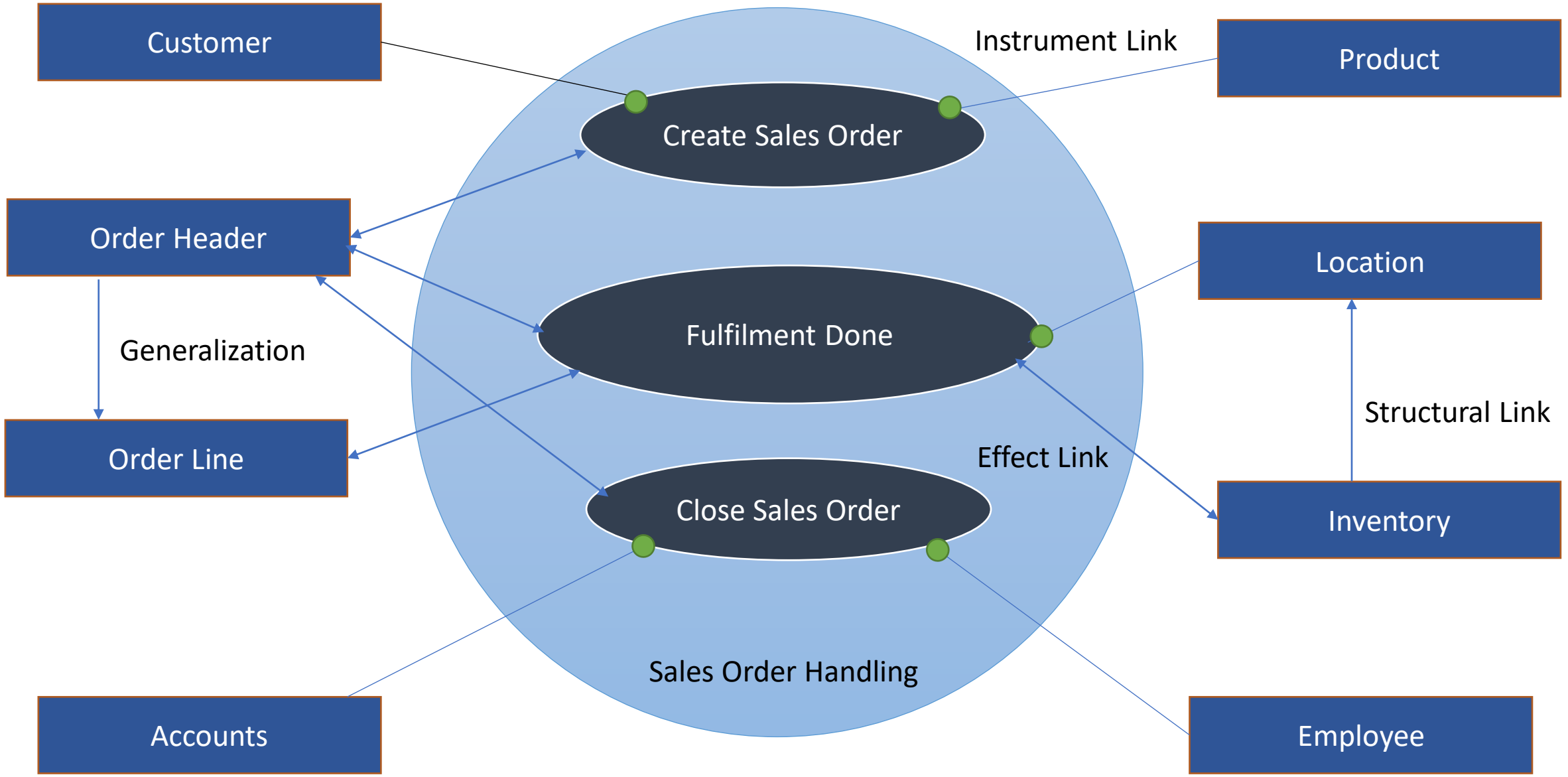

Phone No.
Name
Type
Cust Id
Customer
1
Places
N
Due Date
Cust ID
Total S.P
Order ID
Order Header
1
Contains
N
Order ID
Line No
SKU ID
Fulfilled QTY
Due Qty
Inventory ID
Order Line
N
Fulfills
1
SKU ID
Pre-advice ID
Emp ID (QC Flag)
Inventory ID
Margin
Price
QTY
Location ID
Category ID
Is in
1
Location
Access Level
Phone No.
Emp ID
Name
Employee
1
Handles
N
Inventory
1
Handles
N
Accepts Payment
1
Accounts
N
Emp ID
To ID
Amount
Type
Transaction ID
1
Settles Payment
Supplier ID
Pre-advice Amount
Pre-advice ID
Due Date
1
Pre-advice Header
N
Fulfills
1
Contains
Supplier
1
Phone no.
Type
Name
Supplier Id
N
N
Pre-advice Line
Replenishes
Ordered QTY
Line No
Pre-advice ID
Item Cost
SKU ID
Fulfilled QTY
1
Has
1
Product
N
SKU ID
Name
Reorder Qty
Category ID
Description
Belongs to
Description
Category ID
1
Product Category

ER Diagram

Purchase Order Handling

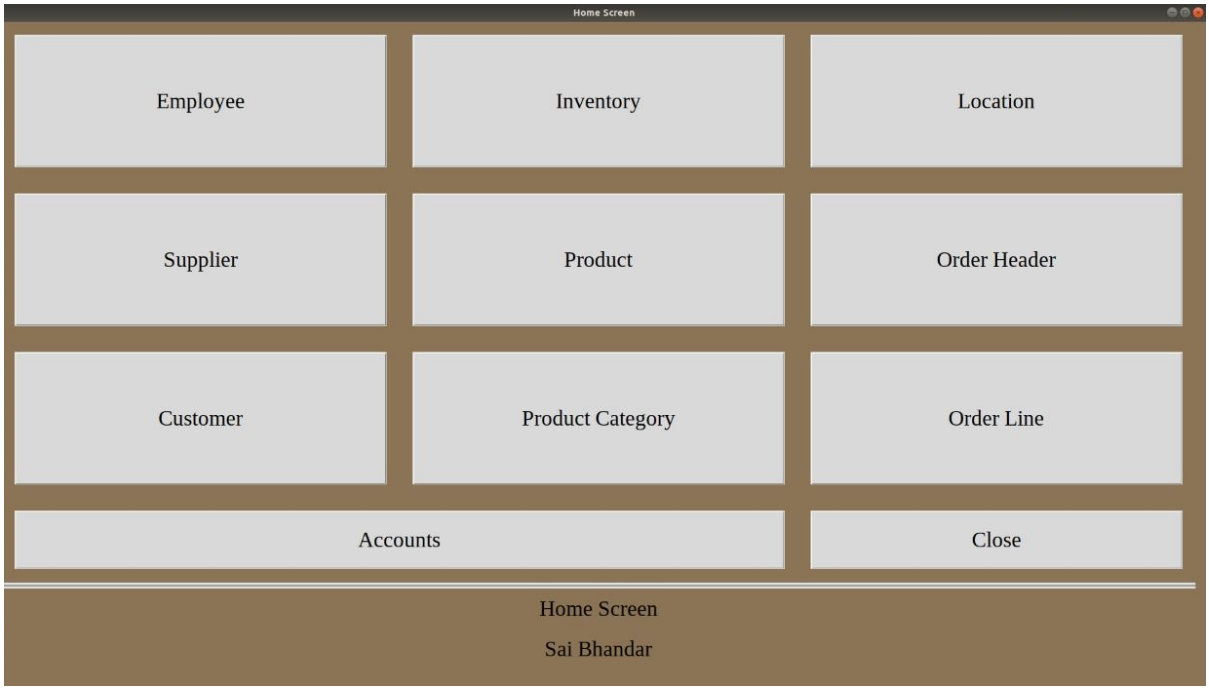


Sales Order Handling



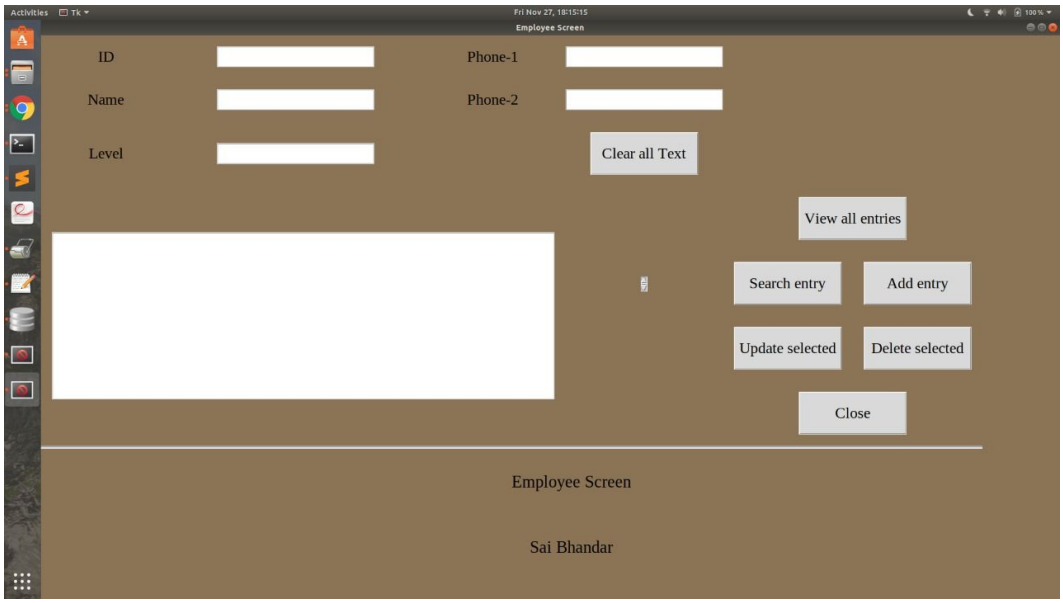
ERP Manual

Homescreen



Upon launching the application we land on the Homescreen page. The home screen allows an user to visit various modules of the ERP application. The home screen also contains the Close option to exit the application.

Employee Screen



The employee screen will allow the user to

1. Create an User
2. Modify an User detail
3. Delete an Existing User
4. Search an User detail

Explanation of the fields

1. ID : Employee Id is a unique 8 digit code which is assigned to a particular user and is an immutable field for the employee once it is created
2. Name : Employee Name; Can be modified to accommodate name change if a particular user undergoes the process
3. Level : Employee access level; Used to specify which all screen an user will have access to and what all functionalities in the screen the user can be able to use. There are 3 distinct user levels: Admin, Super, Floor
4. Phone 1 and Phone 2 : To input/display employee's contact details

Customer Screen

The screen will allow an user to:

1. Add a new customer
2. Delete an existing customer
3. Modify details of an existing customer
4. View details of an existing customer

Explanation of the fields

1. ID : Customer id, unique 6 digit code which is assigned to a particular customer and is an immutable field for the customer once it is created
2. Name : Customer Name; Can be modified to accommodate name change if a particular customer undergoes the process
3. Type : Customer Type; Used to specify the customer type of a particular customer. Based on customer type the store can extend additional discounts and flexibility in payment. There are 2 distinct types : Regular, Privileged

4. Phone 1 and Phone 2 : To input/display customer's contact details

Supplier Screen

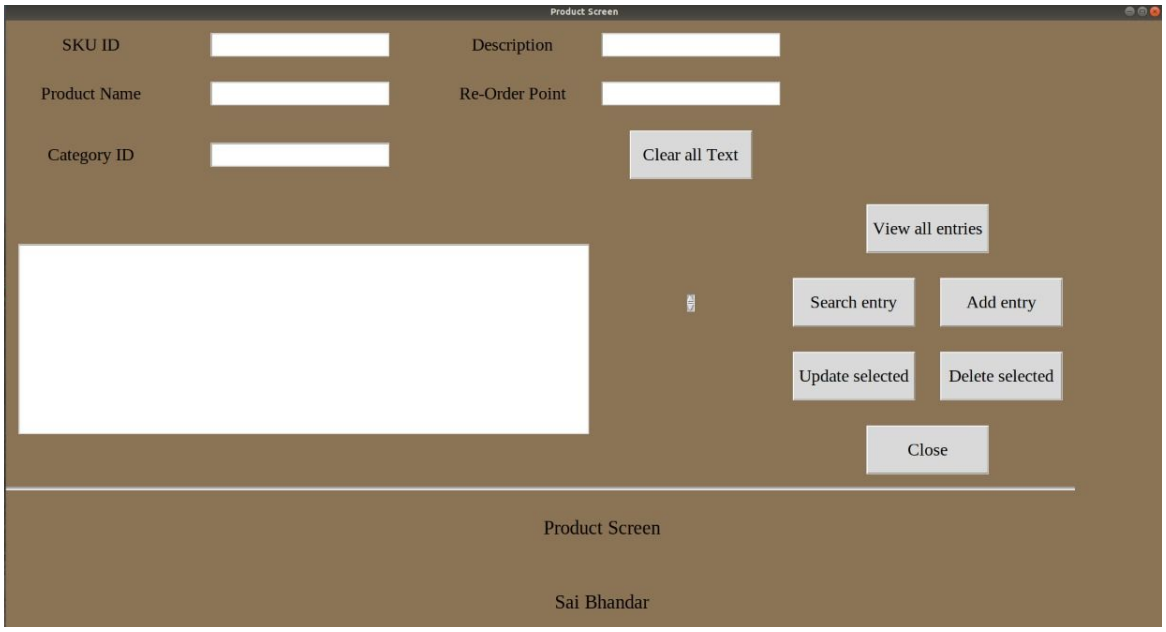
The screen will allow an user to:

1. Add a new Supplier
2. Delete an existing Supplier
3. Modify details of an existing Supplier
4. View details of an existing Supplier

Explanation of the fields

1. ID : Supplier id, unique 6 digit code which is assigned to a particular customer and is an immutable field for the customer once it is created
2. Name : Supplier Name; Cannot be modified
3. Type : Supplier Type; Used to specify the supplier type of a particular supplier. Based on supplier type the store will place requisition for replenishment with appropriate lead time and flexibility in payment. There are 2 distinct types : Regular, Golden
4. Phone 1 and Phone 2 : To input/display Supplier's contact details

Product Screen



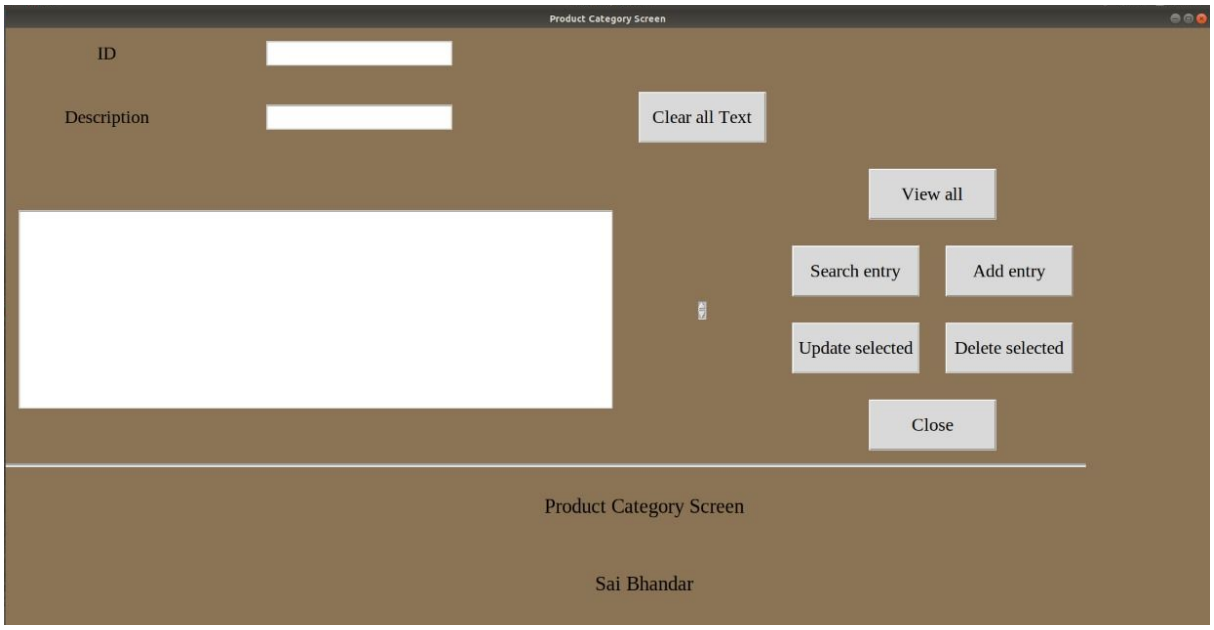
The screen will allow an user to:

1. Add a product
2. Delete a product
3. Modify details of an existing product
4. View products present in the system

Explanation of the fields

1. SKU Id : Stock Keeping Unit (SKU) is a unique set of numbers and letters used to identify, locate and track a product internally within the ERP system
2. Product Name : Name of the product; Will contain the product name along with a brief about the weight category of an article
3. Category Id : Represents the product category. During creating/modifying a particular product, it must be mapped to a valid category id existing in the system
4. Description : Contains a brief description about the product including the list of suppliers
5. Re-order Point : The minimum quantity of inventory, beyond which system will prompt the user to place a replenishment order with the supplier

Product Category Screen



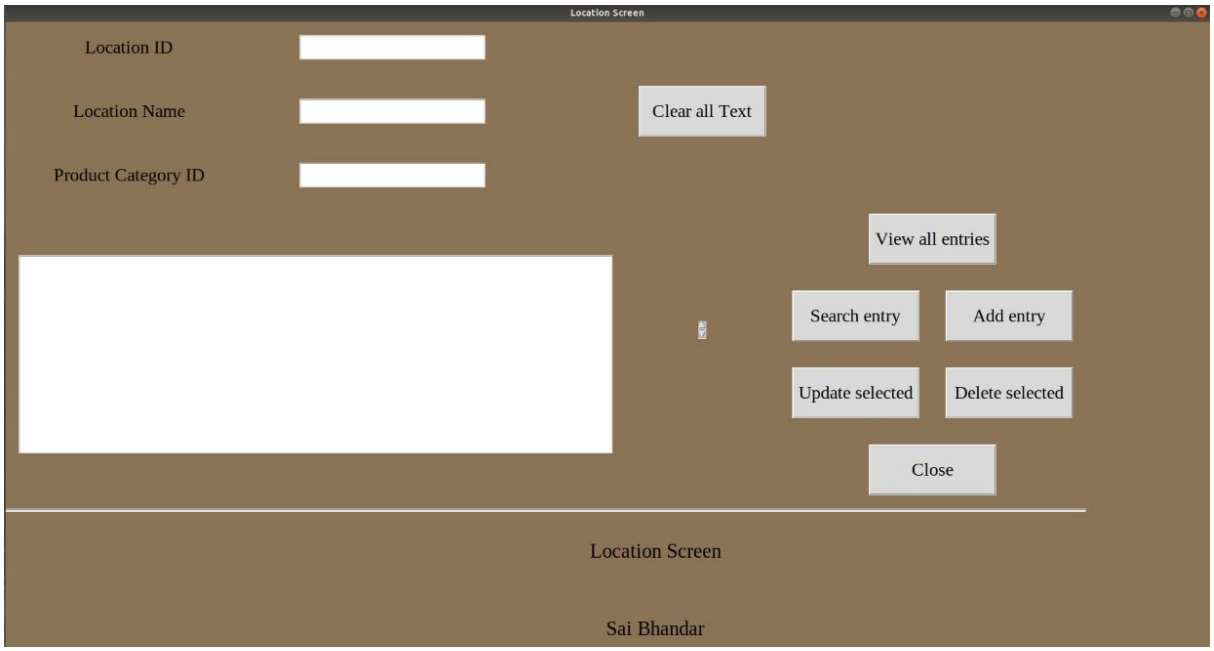
The screen will allow an user to:

1. Add a new product category
2. Delete an existing product category
3. Modify the description of an existing product category
4. View the details of the product categories present in the system

Explanation of the fields:

1. Id : A 2 digit unique id representing the product category id. This is an immutable field.
2. Description : Refers to a short textual description of the corresponding product category. This is a mutable field.

Location Screen



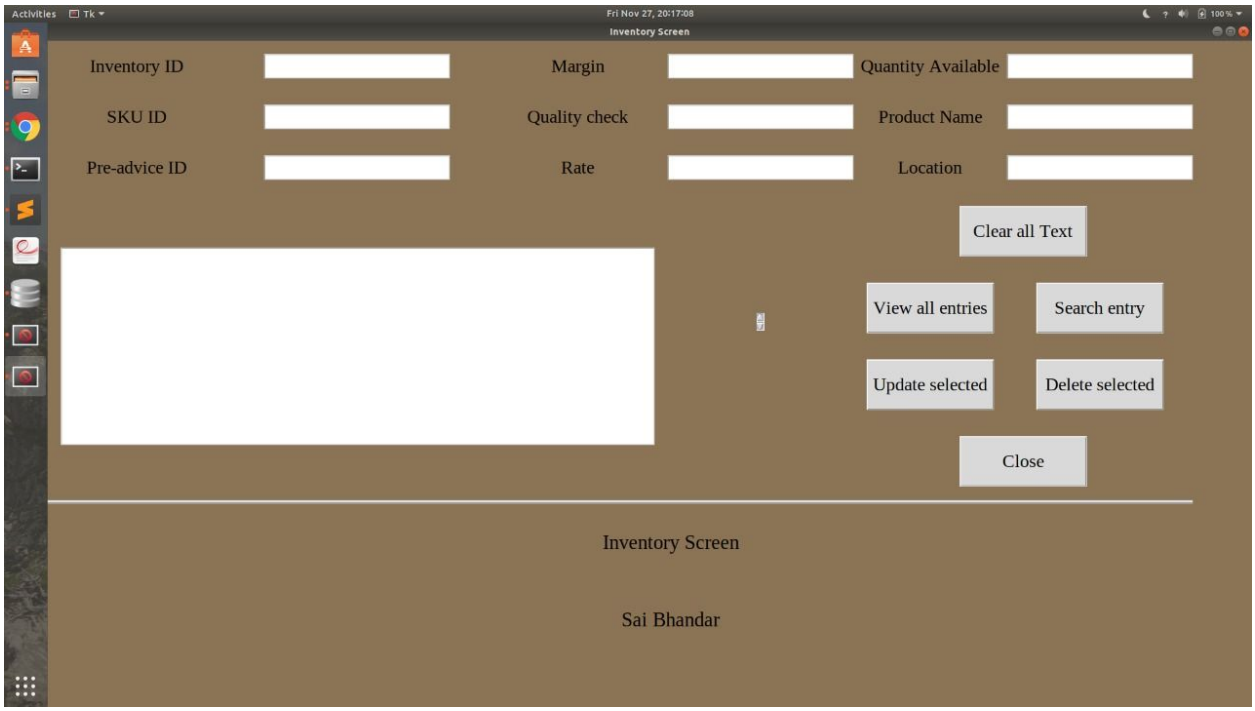
The screen will allow an user to

1. Add a new location
2. Delete an existing location
3. Modify details of an existing location
4. View details of the locations present in the system

Explanation of the fields:

1. Location Id : Unique 2 digit id representing an unique location. This is an immutable field.
2. Location Name : Name of the location. Location refers to a particular location on the store floor. This is an immutable field
3. Product Category Id : Will contain the product category id of the class of products that can be stored in that particular physical space. This is a mutable field.

Inventory Screen



The screen will allow an user to

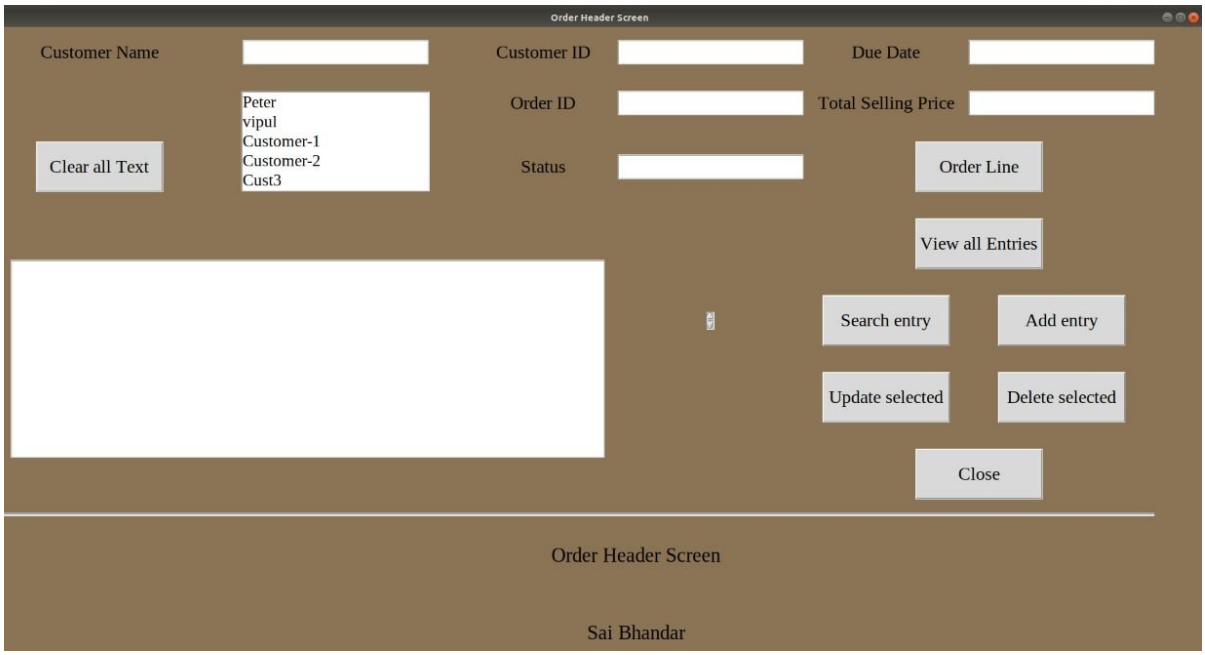
1. View all the inventory present in the store
2. Modify an existing inventory
3. Delete an existing inventory

Explanation of the fields

1. Inventory Id: Unique 11 digit code representing the inventory in the system. The format of the id being ddmmyyyy<sequence of entry in the system for that date>
2. SKU Id: Unique id representing the product
3. Pre-advice Id: Unique 4 digit id representing the pre-advice against which the inventory was received
4. Margin: A 3 digit number representing the profit margin on the inventory. This will be used to calculate the selling price of an inventory

5. Quality Check: Will contain 8 digit employee id corresponding to the employee who performed quality check for that particular inventory on receipt
6. Rate: Represents the price per unit of the item at which the product was procured
7. Quantity Available: Will represent the available inventory quantity for that particular product
8. Product Name: Field will be updated with the product name of the inventory corresponding to the SKU id
9. Location: Represents the location where the product is present. This is automatically procured by linking sku id-product category id-location id-location name

Order Header Screen



The screen will allow an user to

1. Create a new order
2. Modify an existing order
3. Delete an existing order
4. View existing orders present in the system

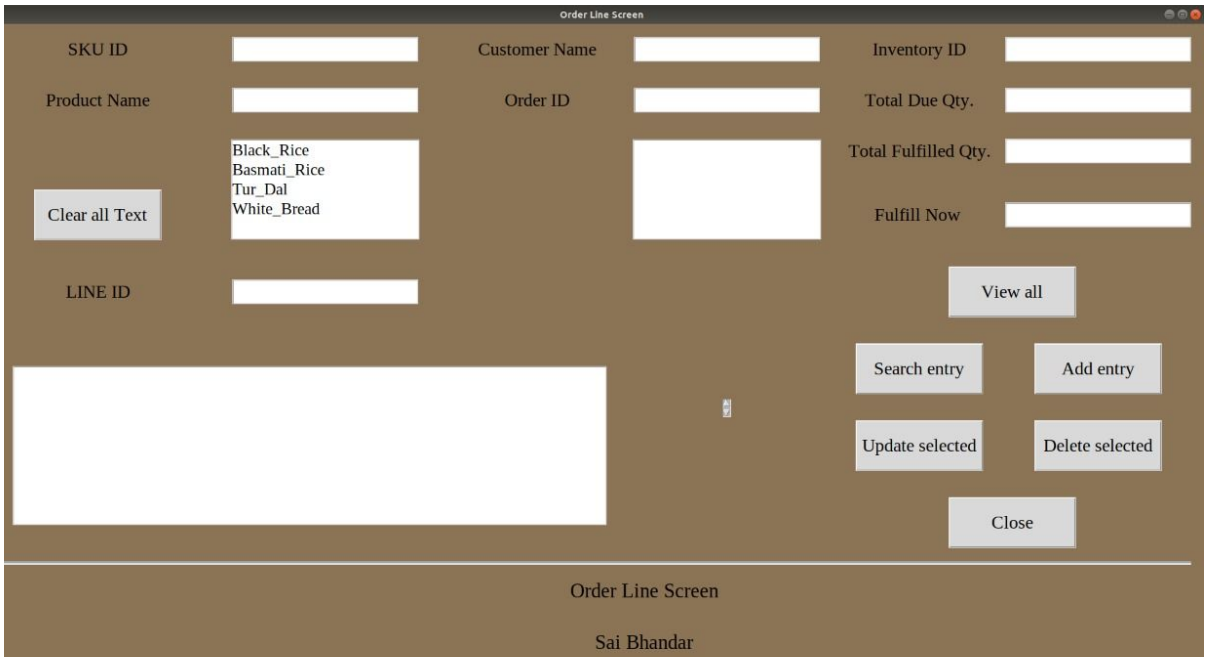
Explanation of the fields

1. Customer Name: A drop down field to select existing customers. For a new entry, user will enter the customer's name in the field.
2. Customer ID: An auto generated 6 digit code representing an unique customer.
3. Order ID: A 11 digit unique in the format yyyymmdd<sequence of the order created on that day>
4. Status: Contains the status of a particular order. Status can be: Order pending, Order closed, Order paid
5. Due Date: The date at which the order is to be delivered

6. Total Selling Price: Represents total selling price of that particular order. For an order having 2 order lines with
    a. products A having quantity as 2, margin as 20%, rate as 100
    b. product B having quantity as 3, margin as 30%, rate as 50

    It is calculated as follows: 2*30*1.2+3*50*1.3

Order Line Screen



The screen will allow an user to:

1. Add a order line to an existing order
2. Modify the order line details of an existing order
3. Delete an order line of an existing order
4. View order lines of an existing order

Explanation of the fields

1. SKU Id: Unique product id of the product for which the order is being placed
2. Product Name: Name of the product present in the order line. For creating a new order line user will be provided with the option of selecting from one of the products present in the system
3. Line Id: Represents the line number corresponding to the particular order
4. Customer Name: Contains the name of the customer who has placed the order. This field will be auto populated with the customer name present in the order header details
5. Order Id: Contains the unique order id for the current order line

6. Inventory Id: Represents the inventory which was used to fulfill the particular ordered product. For creating an order line, users need not fill in the inventory id. Once the process for order fulfillment begins, only then will the user populate the inventory id from which the ordered product was fulfilled
7. Total Due Qty: Represents the total due quantity, i.e, the number of quantity of the item for which the order has been placed. The field is to be populated at the time of order line creation.
8. Total Fulfilled Qty: Represents the total fulfilled quantity, i.e, the number of items which has been so far allocated corresponding to the ordered quantity. This field will be auto populated while user modifies the Fulfill Now field.
9. Fulfill Now: To initiate and complete product allocation corresponding to the ordered quantity. This field is to be filed by the user once he/she has validated the number of items that are present in the inventory for the corresponding product. Once the user enters a particular quantity in this field, the current inventory decreases by that amount and the fulfilled quantity increases by the same amount. This field is only to be accessed at the time of order fulfillment through the **Modify** option and need not be addressed while performing any of the other operations

Account Screen

The screen will allow user to

1. Add a transaction
2. View past transactions
3. Modify a past transaction

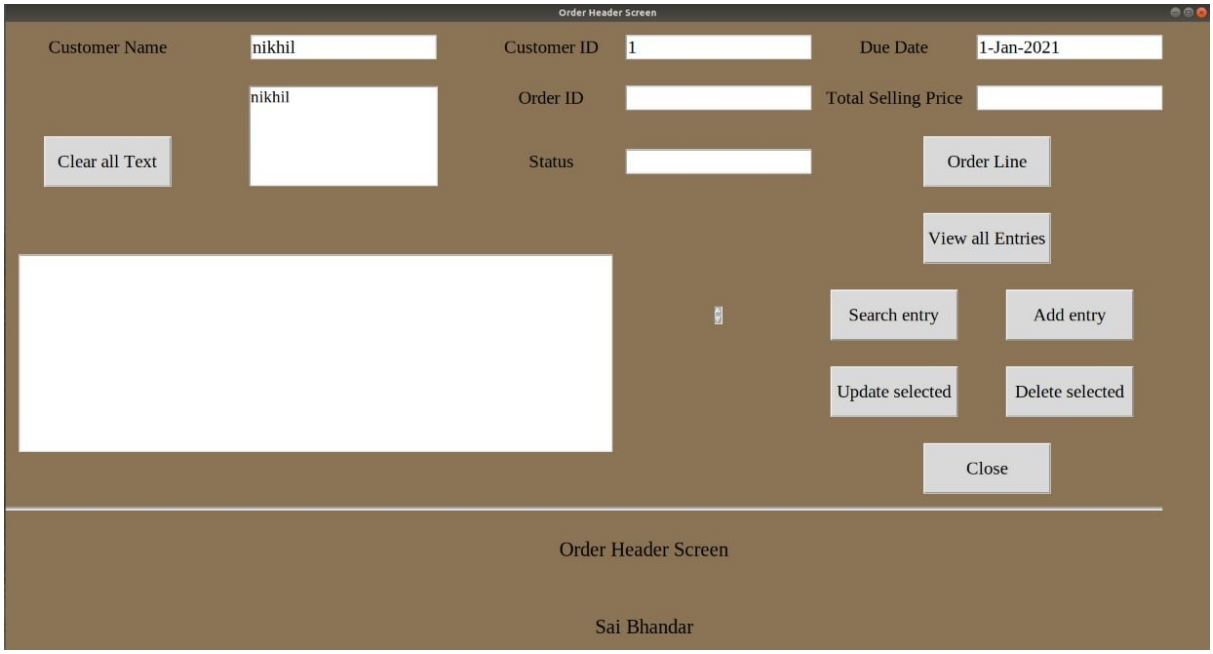
Explanation of the fields

1. Customer Id: Represents the customer id against whom the transaction is being performed
2. Order Id: Represents the order id whose payment is being fulfilled
3. Total Due Amount: Represents the total selling price of the order which is to be settled. This is an immutable field.
4. Total Paid Amount: Represents the total amount which the customer has so far paid for the order. This is an immutable field.
5. Settle Now Amount: User will enter the amount which has being received for the order

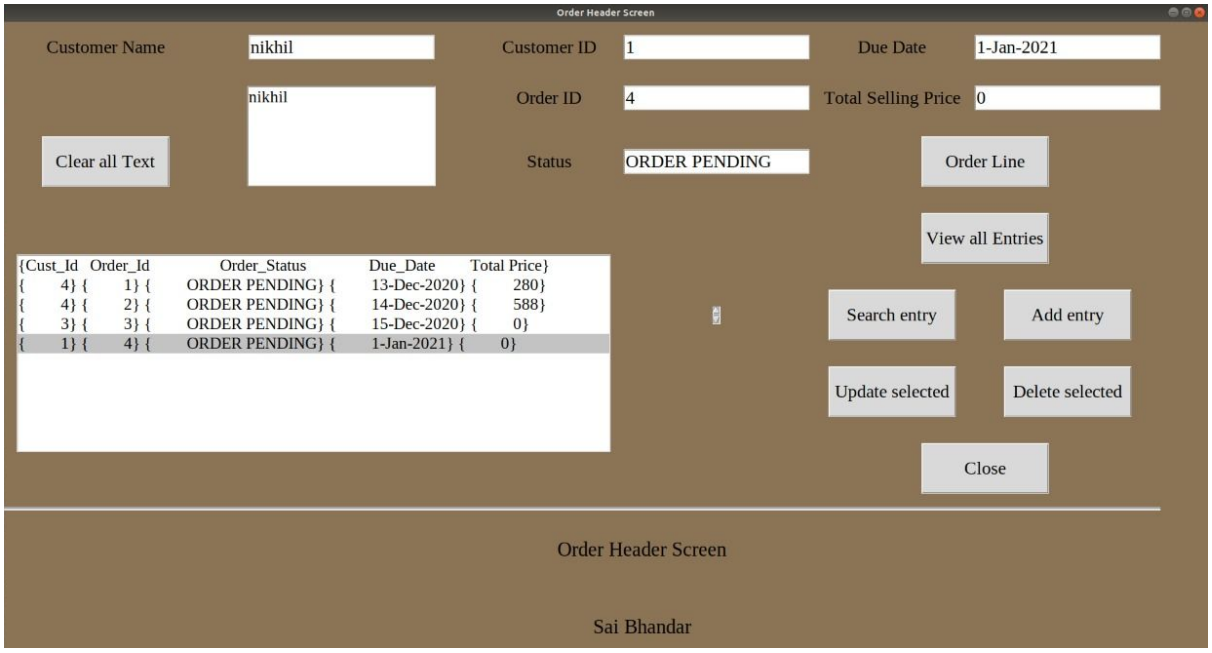
Process Demo : Order Fulfillment

1. User lands on the homescreen

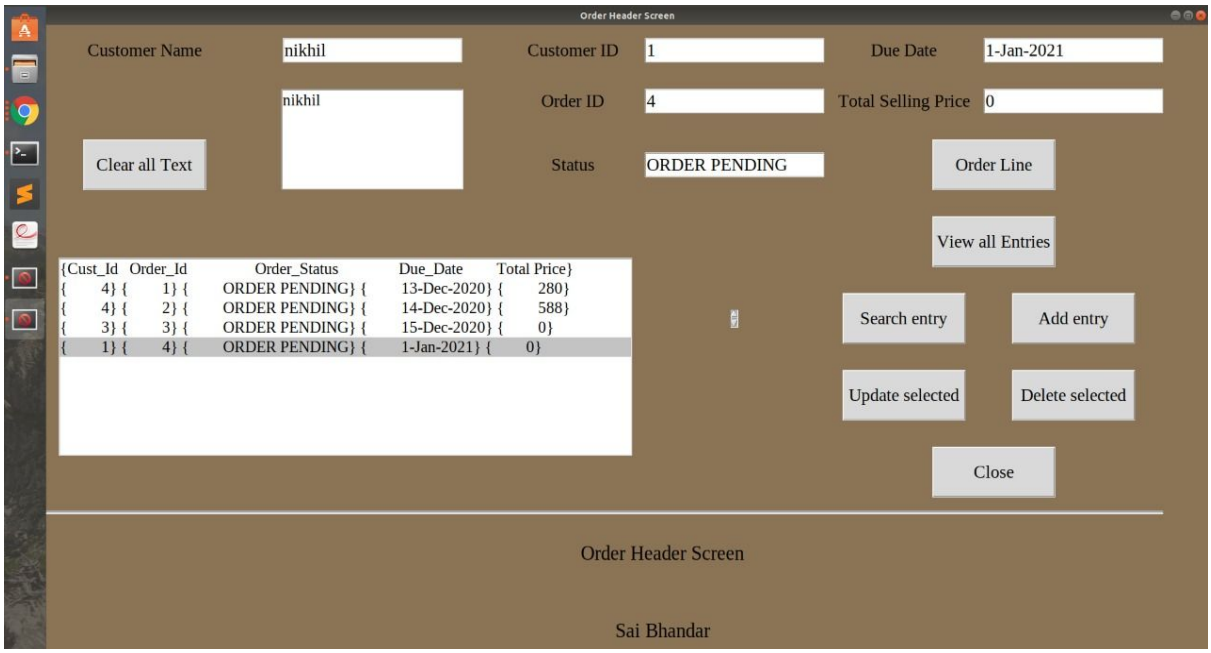
2. Select Order Header



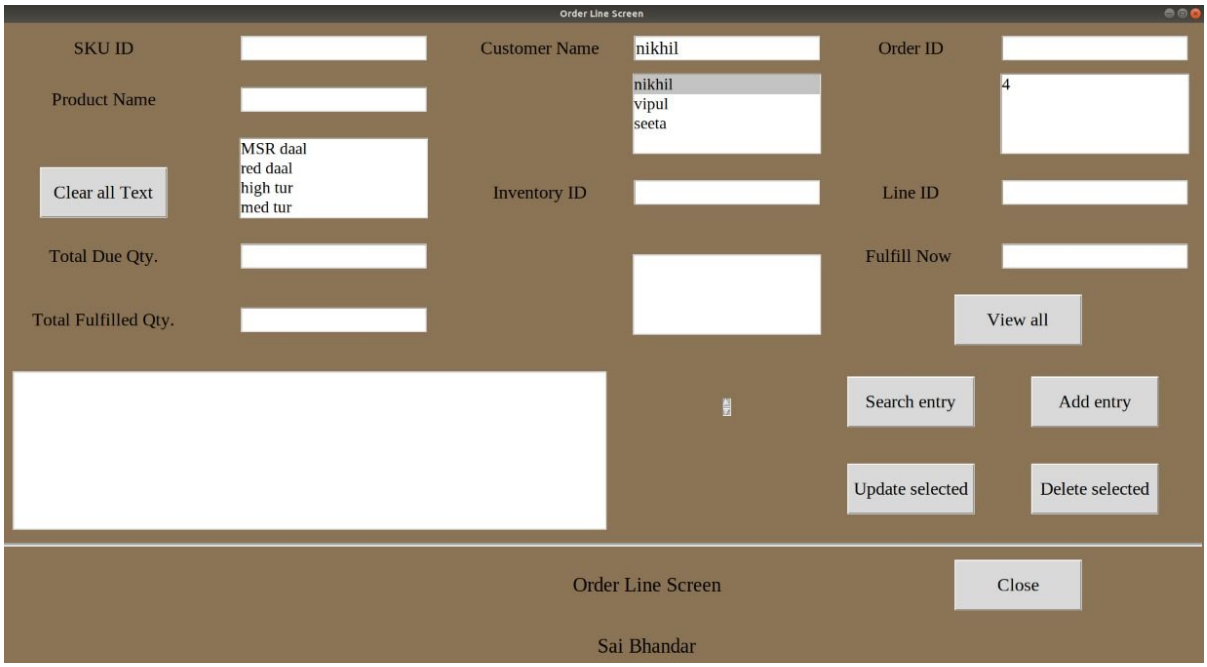
3. Order gets created with status as ‘Order Pending’



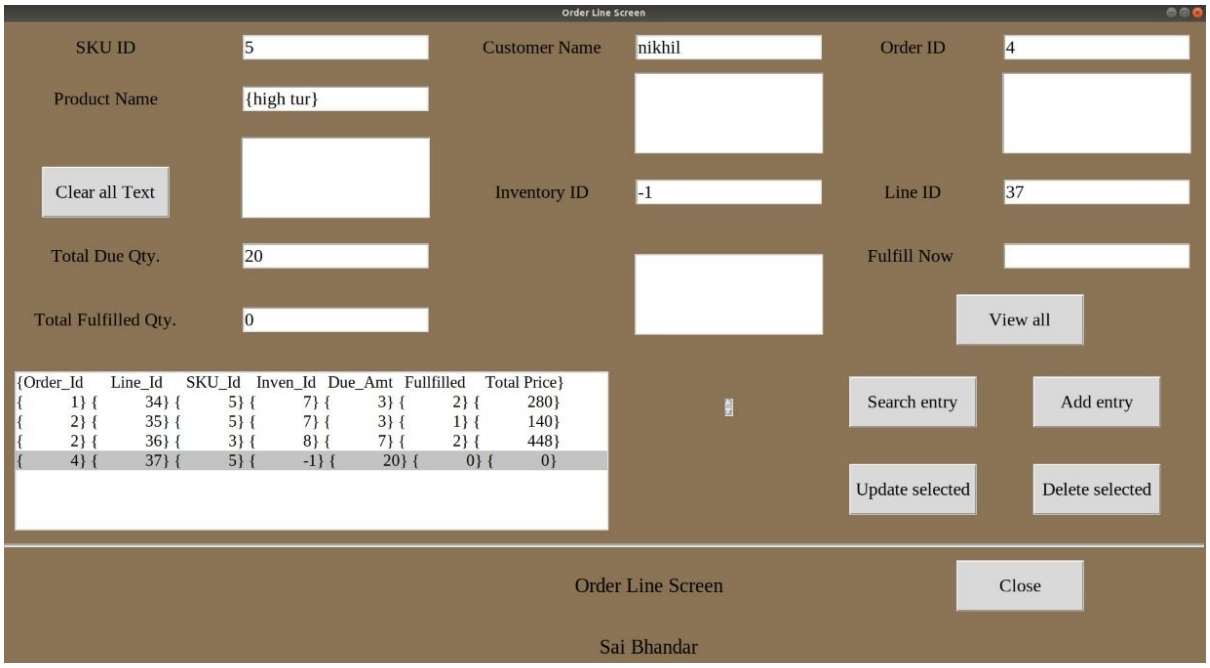
4. Enter the customer name to see the ERP search through the list of available names in the system. Customer Id is auto populated. We select a due date. And hit Add entry. This will create a new order with an auto generated order id and status by default taking up the value of ‘Order Pending’



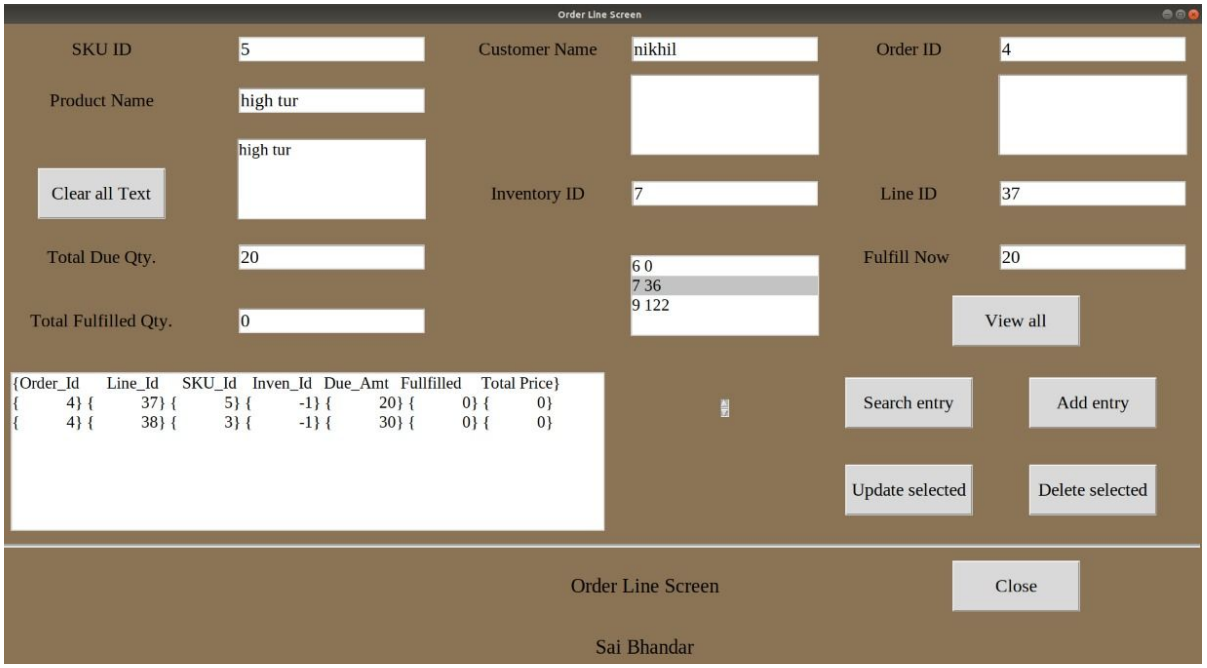
5. Navigate to the order line screen to create order lines for the corresponding order.
6. Enter customer name for the ERP to suggest the available customers. Displayed will be the order ids of the orders that the customer has placed arranged as : Recent order on the top



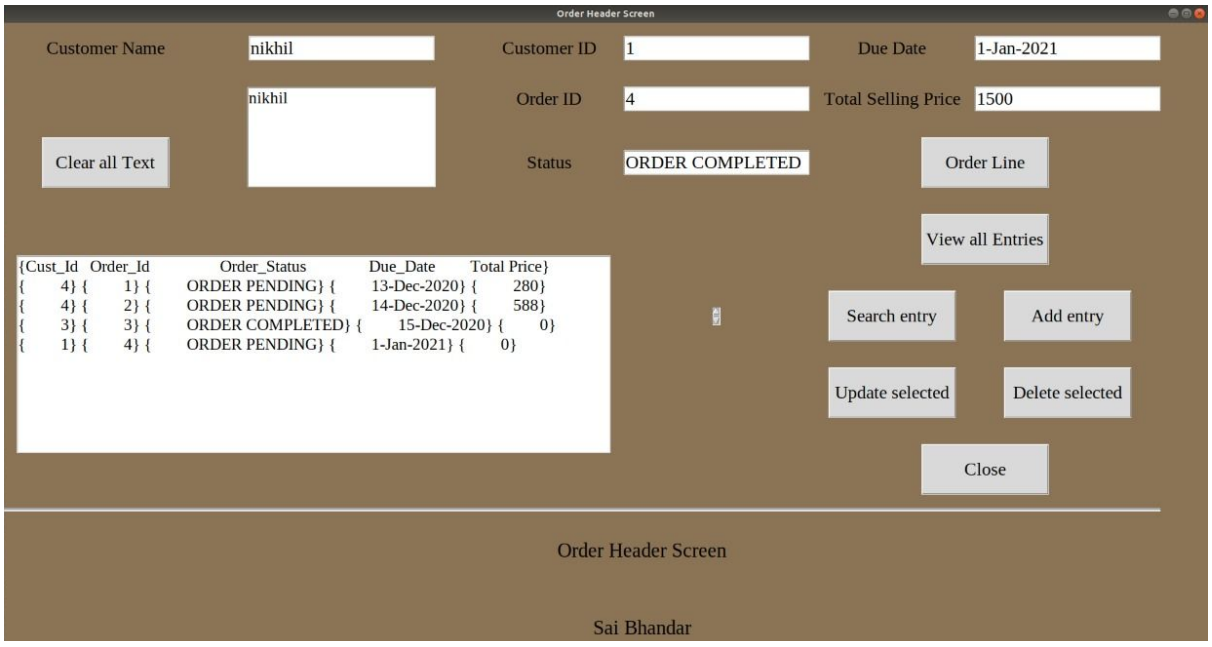
7. By default it assumes fulfilled qty as 0 and since inventory is not yet mapped to the order line yet, it fills the value as -1. Line id too is auto generated



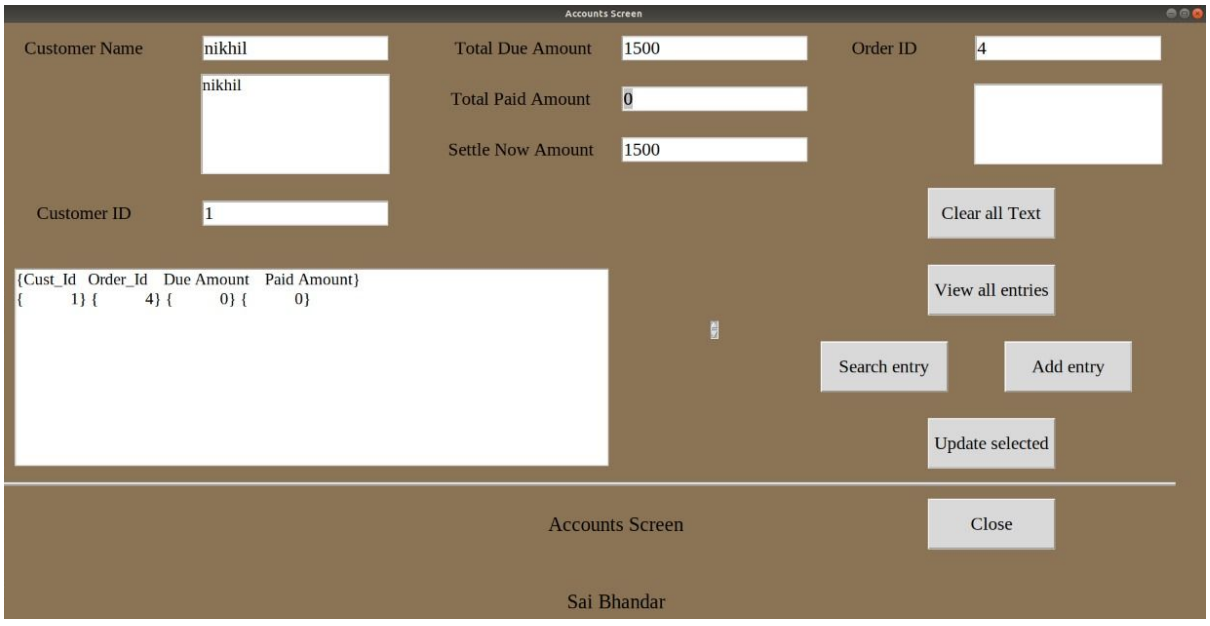
8. Now we start fulfilling the order, by inventory id 7 which contains 36 items



9. Correspondingly the order header screen gets updated with the total selling price and order status gets updated to ‘Order Completed’



10. Also the accounts screen is ready to accept payment against the order. We will be fully settling the payment



11. After the entire order is paid, the due amount reduces to 0 and total paid amount gets updated with the payment received