

Bar Management

Requirements

1. Users are able to login to the system using credentials.
2. Users Can Add new Items to the database.
3. Users Can Add a new Table in the respective category.
4. Once logged in users can Select an existing table to start taking orders.
5. User can select an existing table to update or edit the order Items.
6. User can print the bill which makes the current table marked for collecting payments.
7. Once User collects the payment the paid option will free the table and get removed from the love dashboard.
8. The user can Download an excel file of the daily sales.
9. Users can add the purchase information from the Bill from the Distributor along with discounts and other options.

Here one of the improvements will be that we are going to provide the government names compared to the local names. So we are maintaining the updated list every time.

UI Wire-Frames.

Balsamiq : Presentation [link](#)

Click on presentation to view the application

Login Page

1 / 5

Bar Management System.

Welcome to bar Mangement

Enter password to continue

Login

Here We are not including the User Information Since we maintain a single login

Add Table

Close

Table Number

15

Category

General

User Name

User1

Save

Cancel

[illegible]

Item Master (Food)

4 / 5

Add Items

Item Category : Food

Product Name :

Product Type : From Govt Name

Counter : Y/N

Product Code : Code

Product SubType :

GST : 0

Type	MRP	Regular	AC	Garden	Barcode
Full	40	45			
Half	38	65			
8pc	50	66			

Save
Delete

Product Type : From Govt Name

Here Based on the Item Category and Product Item Selected

The Other inputs will change accordingly.
Example Product Subtype, Go down etc..

Item Master (Alcohol)

3 / 5

Add Items

Item Category : Alcohol

Product List : Product List

Product Type : From Govt Name

Counter : Y/N

Knot : Y/N

Product Code : Code

Product SubType : From Govt Name

Godown : Y/N

GST : 0

Type	MRP	Regular	AC	Garden	Barcode
Ltr	40	45			
Qtr	38	65			
Point	41	56			
90 Loose	350	400			
Minature					

Save
Delete

Product Type : From Govt Name

Here Based on the Item Category and Product Item Selected

The Other inputs will change accordingly.
Example Product Subtype, Go down etc..

Class Diagrams

Note : Still working on it and will update with final in some time

- **User** (user_id, username, password_hash, role, created_at)
- **Item** (item_id, name, category, size, price, mrp, barcode, stock_quantity, created_at)
- **Table** (table_id, table_number, category, status, created_at)
- **Order** (order_id, table_id, user_id, status, timestamp)
- **OrderItem** (order_item_id, order_id, item_id, quantity, price)
- **Bill** (bill_id, order_id, total_amount, discount, taxes, final_amount, payment_status, generated_at)
- **Payment** (payment_id, bill_id, amount_paid, payment_method, paid_at)
- **SalesReport** (report_id, date, total_sales, generated_by)
- **Purchase** (purchase_id, distributor_name, bill_number, total_amount, discount, final_amount, purchase_date)

Each class has **associations**:

- **User** places **Orders**.
- **Orders** belong to a **Table**.
- **Orders** contain multiple **OrderItems**.
- **Bills** are generated for **Orders**.
- **Payments** are linked to **Bills**.
- **Sales Reports** are generated by **Users**.
- **Purchases** are recorded separately.

Database Design

Note : Still working on it and will update with final one in some time

Let's design our database so for that first we need to have a class diagram.

```
CREATE TABLE users (  
    user_id INT PRIMARY KEY AUTO_INCREMENT,  
    username VARCHAR(50) UNIQUE NOT NULL,  
    password_hash VARCHAR(255) NOT NULL,  
    role ENUM('admin', 'waiter', 'manager') NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);  
  
CREATE TABLE items (  
    item_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(100) NOT NULL,  
    category VARCHAR(50) NOT NULL,  
    size VARCHAR(20) NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
    mrp DECIMAL(10, 2) NOT NULL,  
    barcode VARCHAR(50) NOT NULL,  
    stock_quantity INT NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```

    item_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    category ENUM('beer', 'whiskey', 'vodka', 'wine', 'soft drinks') NOT
NULL,
    size ENUM('Ltr', 'Qrt', 'Point', 'Nip', 'Miniature', '90 Loose',
'Large', 'Small') NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    mrp DECIMAL(10,2) NOT NULL,
    barcode VARCHAR(50) UNIQUE NOT NULL,
    stock_quantity INT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE tables (
    table_id INT PRIMARY KEY AUTO_INCREMENT,
    table_number VARCHAR(10) UNIQUE NOT NULL,
    category ENUM('indoor', 'outdoor', 'VIP', 'bar') NOT NULL,
    status ENUM('available', 'occupied', 'awaiting_payment') DEFAULT
'available',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    table_id INT NOT NULL,
    user_id INT NOT NULL,
    status ENUM('pending', 'completed') DEFAULT 'pending',
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (table_id) REFERENCES tables(table_id) ON DELETE
CASCADE,
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE
);

CREATE TABLE order_items (
    order_item_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT NOT NULL,
    item_id INT NOT NULL,
    quantity INT NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE
CASCADE,
    FOREIGN KEY (item_id) REFERENCES items(item_id) ON DELETE CASCADE
);

CREATE TABLE bills (
    bill_id INT PRIMARY KEY AUTO_INCREMENT,

```

```

    order_id INT NOT NULL,
    total_amount DECIMAL(10,2) NOT NULL,
    discount DECIMAL(10,2) DEFAULT 0.00,
    taxes DECIMAL(10,2) DEFAULT 0.00,
    final_amount DECIMAL(10,2) NOT NULL,
    payment_status ENUM('pending', 'paid') DEFAULT 'pending',
    generated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE CASCADE
);

CREATE TABLE payments (
    payment_id INT PRIMARY KEY AUTO_INCREMENT,
    bill_id INT NOT NULL,
    amount_paid DECIMAL(10,2) NOT NULL,
    payment_method ENUM('cash', 'credit_card', 'UPI') NOT NULL,
    paid_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (bill_id) REFERENCES bills(bill_id) ON DELETE CASCADE
);

CREATE TABLE sales_report (
    report_id INT PRIMARY KEY AUTO_INCREMENT,
    date DATE NOT NULL UNIQUE,
    total_sales DECIMAL(10,2) NOT NULL,
    generated_by INT NOT NULL,
    FOREIGN KEY (generated_by) REFERENCES users(user_id) ON DELETE
CASCADE
);

CREATE TABLE purchases (
    purchase_id INT PRIMARY KEY AUTO_INCREMENT,
    distributor_name VARCHAR(100) NOT NULL,
    bill_number VARCHAR(50) UNIQUE NOT NULL,
    total_amount DECIMAL(10,2) NOT NULL,
    discount DECIMAL(10,2) DEFAULT 0.00,
    final_amount DECIMAL(10,2) NOT NULL,
    purchase_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

•