**MallEase Application**

**Database Schema Design**

**Tables:**

**1. Users**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | Primary Key, Auto Increment |
| name | VARCHAR(255) | Not Null |
| email | VARCHAR(255) | Not Null, Unique |
| phone | VARCHAR(15) | Not Null, Unique |
| password | VARCHAR(255) | Not Null |
| role | VARCHAR(20) | Not Null (Enum Type) |
| created\_at | DATETIME | Not Null |

**Explanation of Columns:**

* **id**: Unique identifier for each user, automatically generated.
* **name**: User's full name, cannot be null.
* **email**: User's email address, must be unique and cannot be null.
* **phone**: User's phone number, must be unique and cannot be null.
* **password**: User's password, cannot be null.
* **role**: User's role, represented as a string (ENUM type in Java).
* **created\_at**: Timestamp for when the user was created, cannot be nul

**2. Products**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | BIGINT | Primary Key, Auto Increment |
| name | VARCHAR(255) | Not Null |
| description | TEXT | Nullable |
| price | DECIMAL(10, 2) | Not Null |
| qrCode | VARCHAR(255) | Nullable |
| locationInMall | VARCHAR(255) | Nullable |
| stockQuantity | INT | Not Null |

**Explanation of Columns:**

* **id**: Unique identifier for each product, automatically generated.
* **name**: Product name, cannot be null.
* **description**: A detailed description of the product, nullable.
* **price**: The price of the product, cannot be null and formatted to allow two decimal places.
* **qrCode**: A string representing the QR code associated with the product, nullable.
* **locationInMall**: The location of the product within the mall, nullable.
* **stockQuantity**: The number of items available in stock, cannot be null.

**3.Cart**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| cartId | BIGINT | Primary Key, Auto Increment |
| user\_id | BIGINT | Foreign Key, Not Null |
| createdOn | DATETIME | Not Null |
| updatedOn | DATETIME | Not Null |
| status | VARCHAR(10) | Not Null ( *ACTIVE, ORDERED, ABANDONED*) |
| totalPrice | DECIMAL(10, 2) | Nullable |

**Explanation of Columns:**

* **cartId**: Unique identifier for each cart, automatically generated.
* **user\_id**: Foreign key referencing the User entity, cannot be null (links to the User table).
* **createdOn**: Timestamp for when the cart was created, cannot be null.
* **updatedOn**: Timestamp for the last update to the cart, cannot be null.
* **status**: The status of the cart, stored as a string (ENUM type) with possible values like ACTIVE, ORDERED, ABANDONED, cannot be null.
* **totalPrice**: The total price of the items in the cart, nullable.

**4. CartItems**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| cartItemId | BIGINT | Primary Key, Auto Increment |
| cart\_id | BIGINT | Foreign Key, Not Null |
| product\_id | BIGINT | Foreign Key, Not Null |
| quantity | INT | Not Null |
| totalPrice | DECIMAL(10, 2) | Not Null |
| addedOn | DATETIME | Not Null |

**Explanation of Columns:**

* **cartItemId**: Unique identifier for each cart item, automatically generated.
* **cart\_id**: Foreign key referencing the Cart entity, cannot be null (links to the Cart table).
* **product\_id**: Foreign key referencing the Product entity, cannot be null (links to the Product table).
* **quantity**: The quantity of the product in the cart, cannot be null.
* **totalPrice**: The total price for this cart item, cannot be null and is calculated based on the product price and quantity.
* **addedOn**: Timestamp for when the item was added to the cart, cannot be null.

**3. Payments**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| paymentId | UUID | Primary Key |
| user\_id | BIGINT | Foreign Key, Not Null |
| totalAmount | DECIMAL(10, 2) | Not Null |
| paymentDate | DATETIME | Not Null |
| transactionId | VARCHAR(255) | Nullable |
| status | VARCHAR(20) | Not Null |

**Explanation of Columns:**

* **paymentId**: Unique identifier for each payment, generated as a UUID.
* **user\_id**: Foreign key referencing the User entity, cannot be null (links to the User table).
* **totalAmount**: The total amount for the payment, cannot be null and formatted to allow two decimal places.
* **paymentDate**: Timestamp for when the payment was made, cannot be null.
* **transactionId**: A string representing the transaction identifier, nullable.
* **status**: The status of the payment, stored as a string (ENUM type- COMPLETED , PENDING, CANCELLED), cannot be null.

5. \*\*Receipts\*\*

- `receipt\_id` (Primary Key, UUID)

- `payment\_id` (Foreign Key: `Payments.payment\_id`)

- `receipt\_qr\_code` (VARCHAR) // For Security to scan

#### Data Flow

1. \*\*Customer Flow\*\*:

- \*\*Product Browsing & Location\*\*: The user searches for products by category or name and sees real-time product locations.

- \*\*QR Code Scan\*\*: When a user scans the product's QR code, the app retrieves the product details and adds it to the cart.

- \*\*Payment Process\*\*: The user reviews the cart and completes payment. The payment data is stored in the `Payments` table.

- \*\*Receipt Generation\*\*: After payment, a QR code receipt is generated and stored in the `Receipts` table. The user can show this QR to the security checker for verification.

2. \*\*Admin Flow\*\*:

- \*\*Add Products\*\*: Admins can add, modify, or remove products from the inventory.

- \*\*Update Product Location\*\*: Admins can update the real-time location of products within the mall.

- \*\*Inventory Management\*\*: Admins track the stock and add more inventory as needed.

3. \*\*Security Checker Flow\*\*:

- \*\*QR Code Scan for Security\*\*: Security scans the receipt QR code, verifying the purchased products and cross-checking with the cart data. If everything matches, they approve the exit.

#### Services

1. \*\*User Service\*\*:

- Register/Login service for customers and admins.

- Product browsing and search services with real-time location information.

- QR code scanning service for adding products to the cart.

- Payment processing service integrated with a payment gateway.

- Receipt generation service after successful payment.

2. \*\*Admin Service\*\*:

- Admin login.

- Product management (add, update, remove products).

- Update stock and locations of products.

- View sales data and customer orders.

3. \*\*Security Service\*\*:

- QR code scanning service to verify purchased products at the exit.

- Check for any discrepancies in the cart against the receipt QR.

4. \*\*Cart Service\*\*:

- Add products to the cart via QR code scan.

- View and update cart items.

- Checkout and payment initiation.

5. \*\*Payment Service\*\*:

- Payment confirmation.

- Order completion, and receipt generation with QR.

6. \*\*Notification Service\*\* (Optional):

- Notify the user when a payment is completed.

- Notify admins for low stock levels.

#### Security Considerations

- \*\*Role-Based Access Control (RBAC)\*\*:

- Only admins can add/update products.

- Security personnel can only access the verification service.

- Customers can view and purchase products.

- \*\*Secure Payment\*\*:

- Integrate with a secure payment gateway (like Stripe or PayPal).

- \*\*Encryption\*\*:

- Encrypt sensitive data like passwords (using bcrypt) and payment information.