Smiling Tailor

User Model

```dart
1   class PktbsUser {
2     //
3     final String id;
4     final DateTime created;
5     DateTime? updated;
6     final String collectionId;
7     final String collectionName;
8     //
9     String name;
10    final String username;
11    final String email;
12    bool emailVisibility;
13    bool verified;
14    String? avatar;
15    UserType type; // enum UserType { admin, manager, operate, dispose }
16  }
```

Vendor Model

```
1   class PktbsVendor {
2     //
3     final String id;
4     final DateTime created;
5     DateTime? updated;
6     final PktbsUser creator;
7     PktbsUser? updator;
8     final String collectionId;
9     final String collectionName;
10    //
11    String name;
12    String phone;
13    String? email;
14    String address;
15    // double openingBalance; -> only add a column in trx table for vendor.
16    // TrxType openingBalanceTrxType; // enum TrxType { debit, credit }
17    String description;
18  }
```

Inventory Model

```
1   class PktbsInventory {
2     //
3     final String id;
4     final DateTime created;
5     DateTime? updated;
6     final PktbsUser creator;
7     PktbsUser? updator;
8     final String collectionId;
9     final String collectionName;
10    //
11    String title;
12    int quantity;
13    Measurement unit;
14    double amount;
15    // double advanceAmount; -> only add a column in trx table for inventory.
16    PktbsVendor from;
17    String? description;
18  }
19
20  class Measurement {
21    //
22    final String name;
23    final String symbol;
24    final String unitOf;
25    final String system;
26  }
```

```dart
1   class PktbsOrder {
2     //
3     final String id;
4     DateTime created;
5     DateTime? updated;
6     PktbsUser creator;
7     PktbsUser? updator;
8     String collectionId;
9     String collectionName;
10    //
11    String customerName;
12    String? customerEmail;
13    String customerPhone;
14    String? customerAddress;
15    String? customerNote;
16    //
17    String? measurement;
18    String? plate;
19    String? sleeve;
20    String? colar;
21    String? pocket;
22    String? button;
23    String? measurementNote;
24    int quantity;
25    //
26    PktbsEmployee? tailorEmployee;
27    // double tailorCharge;  -> only add a column in trx table for tailor.
28    String? tailorNote;
29    //
30    PktbsInventory? inventory;
31    // int? inventoryQuantity;
32    // String? inventoryUnit;
33    // double? inventoryPrice; -> only add a column in trx table for inventory.
34    String? inventoryNote;
35    //
36    PktbsEmployee? deliveryEmployee;
37    String? deliveryAddress;
38    // double? deliveryCharge; -> only add a column in trx table for delivery.
39    String? deliveryNote;
40    //
41    PaymentMethod paymentMethod; // enum PaymentMethod { cash, mfs, card, cheque, others }
42    String? paymentNote;
43    double amount; // total amount = tailorCharge + inventoryPrice + deliveryCharge
44    //
45    DateTime deliveryTime;
46    String? description;
47    OrderStatus status; // enum OrderStatus { pending, processing, ready, shipping, completed, cancelled }
48  }
```

Transaction Model

```dart
class PktbsTrx {
  //
  final String id;
  final DateTime created;
  DateTime? updated;
  final PktbsUser creator;
  PktbsUser? updator;
  final String collectionId;
  final String collectionName;
  //
  String fromId;
  GLType fromType; // enum GLType { vendor, inventory, employee, order, user }
  Map<String, dynamic> from;
  //
  String toId;
  GLType toType; // enum GLType { vendor, inventory, employee, order, user }
  Map<String, dynamic> to;
  //
  double amount;
  Measurement? unit;
  bool isGoods;
  TrxType trxType; // enum TrxType { debit, credit }
  //
  bool isActive;
  String voucher;
  bool isSystemGenerated;
  String? description;
}
```