1. **Phân tích các yêu cầu / Xác định các luồng hoạt động chính**
2. **Đặt bàn** : chọn tầng => chọn bàn => thay đổi trạng thái của bàn
3. **Đặt món** : mở menu ngay khi chọn bàn thành công => chọn món ăn, số lượng => lưu các món được chọn vào order hiện tại của bàn (*OrderItems*) => lưu thông tin order hiện tại vào lịch sử hoạt động với trạng thái ‘Đang hoạt động’
4. **Huỷ bàn** : bên trong menu chọn món sẽ có tuỳ chọn huỷ bàn => thay đổi trạng thái của bàn => lưu vào lịch sử hoạt động (*Orders*)
5. **Gọi thêm món** : một bàn có người trước đó đã hoàn thành việc gọi món muốn gọi thêm món => chọn món ăn, số lượng => thay đổi order trước đó của bàn
6. **Thanh toán** : thực hiện thanh toán hoá đơn => thay đổi trạng tháu của order tương ứng trong lịch sử hoạt động thành ‘Đã thanh toán’ => thay đổi trạng thái của bàn
7. **Theo dõi ca làm việc** : hiển thị thông tin nhân viên cũng như ca làm việc nhằm dễ dàng quản lý
8. **Update doanh thu theo thời gian thực** : tổng kết doanh thu theo thời gian thực, update liên tục theo thời gian thanh toán, lưu lại lịch sử hoạt động theo ngày tháng (*DailyReport*)
9. **Xác định các gói / các lớp cần thiết**
10. **Gói utils (gói tiện ích):**

* **DatabaseConnector** : lớp tiện ích cho phép tương tác qua lại giữa những dòng code java với cơ sở dữ liệu để lấy dữ liệu cũng như cập nhật cơ sở dữ liệu nhằm phục vụ những luồng hoạt động khác

1. **Gói models : các lớp cốt lõi, chứa các khuôn mẫu đối tượng cũng như những phương thức/logic cơ bản**

* Lớp **Tables**
* Thuộc tính : id, floor, status
* markAsOccupied(), markAsAvailable(): cập nhật trạng thái của bàn sao cho phù hợp vào CSDL
* static findById (id) : cho phép tìm 1 bàn cụ thể bằng id
* static findAll() : hiển thị tất cả các bàn có trong CSDL
* Lớp **MenuItems**
* Thuộc tính : id, name, price
* static findById (id) : phương thức tìm 1 món ăn cụ thể bằng id
* static findAll() : phương thức trả về tất cả các món ăn có trong CSDL
* Lớp **Employee**
* Thuộc tính : id, full\_name, workshift
* static findById(id) : phương thức tìm kiếm 1 nhân viên bằng 1 id cụ thể
* static findAll() : phương thức tìm kiếm tất cả các nhân viên có trong CSDL
* Lớp **Orders**
* Thuộc tính : id, tableId, status, total\_amount, created\_at, List<OrderItems>
* addItem() : phương thức cho phép thêm 1 món vào 1 order mới được tạo hoặc vào 1 order đã có sẵn trước đó
* calculateTotal() : phương thức cho phép tính tổng số tiền phải thanh toán của 1 hoá đơn tính đến thời điểm mà khách hàng thanh toán
* save() : phương thức cho phép thao tác vào cơ sở dữ liệu, lưu trữ order mới được tạo nhằm phục vụ những luồng hoạt động khác
* updateStatus() : phương thức thay đổi trạng thái cho 1 order, tham gia vào các luồng hoạt động như thanh toán hoá đơn, huỷ bàn, đồng thời tính toán tổng tiền của hoá đơn tại thời điểm mới nhất
* fetchOrderItem() : phương thức thao tác với cơ sở dữ liêu, dựa vào id của 1 order mà cho phép GUI hiển thị ra tất cả các OrderItems của order đó
* findById() : phương thức thao tác với csdl, cho phép truy vấn tất cả thông tin của 1 order cụ thể bằng id
* findActiveByTableId() : phương thức hỗ trợ GUI thao tác với csdl nhằm hiển thị ra 1 order cụ thể của 1 bàn cụ thể bằng ID
* updateTotalAmount() : phương thức cập nhật tổng tiền hoá đơn tính tới món ăn mới nhất được thêm vào
* Lớp **OrderItems**
* Thuộc tính : id, orderId, menu\_item\_id, quantity, price,
* save() : phương thức tự động tạo 1 dòng OrderItem mới để lưu vào CSDL
* update() : phương thức tự động cập nhật thông tin, chủ yếu là quantity của chính nó trong CSDL
* Lớp **DailyReport**
* Thuộc tính: List<Orders> completedOrders, totalRevenue: danh sách các Order đã được thực hiện trong 1 ngày cùng những thông tin liên quan, cùng với tổng thu nhập có trong 1 ngày cụ thể
* List<Orders> getCompletedOrder : trả về danh sách order đã được thực hiện trong 1 ngày nhất định
* getTotalRevenue() : trả về tổng doanh thu trong 1 ngày cụ thể

1. **Gói Manager : Các lớp đặc thù, không có thuộc tính nhất định, nhưng có những phương thức kết hợp nhiều lớp có trong models lại với nhau để hoàn thành luồng hoạt động**

* **OrderManager**
* startNewOrder(tableId) : phương thức thực hiện luồng Đặt món. Nó sẽ tìm đối tượng *Table* và gọi *table.markAsOccupied()*, sau đó sẽ tạo đối tượng *Order* và gọi *Order.save()*
* checkOut(orderId) : phương thức thực hiện luồng thanh toán, nó sẽ tìm *Order*, gọi *order.markAsPaid()*, sau đó tìm tới *Table* tương ứng và thay đổi trạng thái của bàn
* cancelOrder(orderId) : phương thức thực hiện luồng hoạt động Huỷ bàn, khi này nó sẽ tìm tới *Order*, gọi *order.markAsCancel*, sau đó tìm tới *Table* tương ứng để thay đổi trạng thái
* **ReportManager**
* genDailyReport(LocalDate date) : phương thức cho phép truy vấn vào cơ sở dữ liệu để lấy danh sách những order đã được thực hiện trong ngày *date* và tổng doanh thu của ngày hôm đó

1. **Gói Main : Lớp chứa hàm để khởi chạy**

