**Các package và chức năng**

**1. bean: Quản lí, lưu trữ dữ liệu của một danh mục trong giao diện java swing.**

**- DanhMucBean.java:**

+ Quản lí thông tin của các chức năng chính trong menu

+ kind: dữ liệu String trả về khi click vào chức năng

+ jpn: JPanel của chức năng đó **<Hiển thị giao diện>**

+ jlb: JLable của chức năng đó **<Hiển thị chữ>**

**- KhoaHocBean.java:**

+ Quản lí thông tin, của một khóa học: tên khóa học, ngày bắt đầu, ngày kết thúc.

**- LopHocBean.java:**

+ Quản lí thông tin liên quan đến lớp học: ngày đăng kí, số lượng sinh viên đăng kí.

**2. model: Dùng để định nghĩa, khai báo cho các lớp đại diện**

**- KhoaHoc.java**

**- LopHoc.java**

+ Chứa hết các thông tin của sinh viên trong lớp

< lấy từ class KhoaHoc và class SinhVien>

+ Có thể coi class LopHoc là cha của SinhVien và KhoaHoc.

+ Thiết lập các giá trị liên quan đến khóa học  
 (mã sinh viên, tên khóa học, ngày bắt đầu, kết thúc,...)

+ In các giá trị đó ra.

**- Person.java**

+ Là 1 lớp abstract, chỉ dùng 1 lần để cho các lớp con thừa kế lại và không sử dụng trực tiếp trong bài

**- GiangVien.java**

**- SinhVien.java**

+ Thiết lập các giá trị liên quan đến sinh viên  
 (mã sv, tên, sđt, ngày sinh, ...)

**- TaiKhoan.java**

**3. dao: Data Access Objec, chỉ để quản lí các kết nối tới DataBase.**

**Sau đó đưa tới Service để thực hiện các logic riêng.**

**- DBConnect.java** + Dùng để kết nối project với Database

**- SinhVienDAO.java**

+ Là một interface

+ Chứa các phương thức chung chung, cơ bản :

**Trả về danh sách sinh viên vào DataBase**

**Cập nhật thông tin sinh viên vào DataBase**

**- SinhVienDAOImpl.java** + Triển khai và xử lí các phương thức cụ thể, tỉ mỉ hơn so   
 với SinhVienDao.

+ Truy cập vào bảng **sinh\_vien** ở trong DataBase  
 **- ThongKeDAO.java**

+ Là một interface

+ Gọi ra các phương thức chung chung:  
 1.public List<LopHocBean> getListByLopHoc();

Thống kê số sinh viên đăng kí

2.public List<KhoaHocBean> getListByKhoaHoc();

Thống kê các khóa học và thời gian học

+ Nó sẽ lấy dữ liệu trong database để gửi tới cho service xử lí logic

**- ThongKeDAOImpl.java**

+ Truy xuất, quản lí dữ liệu thống kê trong database

**- TaiKhoanDAO.java**

+ Tương tự như trên, nó có phương thức xác nhận việc đăng nhập của người dùng, truy vấn tới DataBase.

**- TaiKhoanDAOImpl.java**

+ Truy cập vào phần tai\_khoan trong DataBase để tìm kiếm tài khoản – mật khẩu được lưu từ trước.

**4. service: thực hiện các chức năng, logic riêng để xử lí   
 (Gọi vào để xếp hàng chuẩn bị)  
 Sau đó sẽ đưa tới Controller để xử lí in ra ngoài giao diện.**

**- SinhVienService.java:**

+ Là một interface

+ Nó dùng để “Giao Nhiệm Vụ”, cho Impl trong service phải thực hiện

+ Dùng để hiển thị sinh viên ra giao diện.

+ Cập nhật thông tin sinh viên hiển thị ra giao diện

**- SinhVienServiceImpl.java:**

+ Implementation: thực thi, thực hiện

+ Có nhiệm vụ là thực thi lại những nhiệm vụ mà **SinhVienService** giao cho.

**- ThongKeService.java**

+ Yêu cầu file Impl trả ra giao diện của Lượng sinh viên đăng kí và thống kê khóa học.

**- ThongKeServiceImpl.java**

+ Thực thi nhiệm vụ **ThongKeService** giao cho  
 + Trả ra các thống kê về sinh viên, khóa học.

**- TaiKhoanService.java**

+ Là một interface

+ Chứa các phương thức chung

+ Giao nhiệm vụ cho Impl thực thi

**- TaiKhoanServiceImpl.java**

+ Sẽ xử lí kĩ hơn so với TaiKhoanService

VD: Sau khi người dùng nhập tk-mk, thì phần DAO sẽ truy cập để truy vấn tới DataBase để lấy thông tin rồi gửi tới Service để xác nhận, xử lí luồng “ Có thằng vừa nhập tk-mk”  
 Sau khi đã nhập xong tk-mk rồi thì nó sẽ đưa nốt tới **Controller** để xử lí, in cái gì ra ngoài giao diện phụ thuộc vào yêu cầu trong phần **Controller** xử lí.

**5. controller: Điểu khiển hành động cho các giao diện.  
(Nhận các yêu cầu từ Service, xử lí xong sẽ in ra màn hình)**

**- ChuyenManHinhController.java:**

+ Dùng để chuyển giữa các chức năng trong menu.

**- QuanLySinhVienController.java:**

+ Dùng để quản lí, điều khiển chức năng: Quản lí sinh viên.

**- SinhVienController.java:**

+ Dùng để quản lí, điều khiển nơi: Nhập thông tin sinh viên.

**- QuanLyGiangVienController.java**

**- GiangVienController.java**

**- QuanLyThongKeController.java:**

+ Quản lí các chức năng hiển thị của: Thống kê dữ liệu

**- TaiKhoanController.java:**

+ Dùng để quản lí, điều khiển nơi: Đăng nhập.

**6. images: Dùng để lưu chữ các ảnh cho giao diện**

**7. main**

**- Main.java** + Chạy chương trình chính

**8. utility**

**- ClassTableModel.java**

+ Dùng để hiển thị danh sách sinh viên, được cấu hình theo một dạng bảng.

**9. view: Thiết kế các giao diện**

**- MainJFrame.java**

+ Hiện thị giao diện chính của ứng dụng

**- TrangChuJPanel.java**

+ Hiện thông tin của chức năng: Màn hình chính

**- SinhVienJFrame.java**

+ Nhập thông tin sinh viên.

**- SinhVienJPanel.java**

+ Hiện ra thông tin sinh viên của lớp.

**- GiangVienJFrame.java:**

+ Nhập thông tin giảng viên.

**- GiangVienJPanel.java:**

+ Hiển thị thông tin giảng viên ra màn hình

**- ThongKeJPanel.java**

+ Hiển thị giao diện thống kê sinh viên, khóa học

**- KhoaHocJPanel.java**

**- DangNhapJDialog.java:**

+ Hiển thị giao diện đăng nhập