## **1. Danh sách chức năng (source code hiện tại)**

**Chức năng trang người dùng (User):**

* **Đăng ký tài khoản:** Cho phép người dùng tạo tài khoản mới trên hệ thống.
* **Đăng nhập/Đăng xuất:** Người dùng đăng nhập vào hệ thống bằng tài khoản đã có (bao gồm cả tính năng **quên mật khẩu** với xác thực qua email).
* **Trang chủ (Home):** Hiển thị giới thiệu về website và danh sách một số căn hộ mới nhất.
* **Xem danh sách căn hộ:** Gồm hai loại danh sách: **căn hộ hệ thống** (do admin quản lý) và **căn hộ ký gửi** (do người dùng đăng). Người dùng có thể xem các căn hộ theo từng danh mục này, xem thông tin chi tiết từng căn hộ.
* **Chức năng tìm kiếm/Lọc:** Cho phép tìm kiếm căn hộ (theo tiêu đề, địa điểm, …) và gợi ý từ khóa (autocomplete) khi người dùng nhập liệu.
* **Danh sách yêu thích:** Người dùng lưu các căn hộ ưa thích và xem lại danh sách các căn hộ đã đánh dấu.
* **Trang tài khoản người dùng:** Xem và cập nhật thông tin hồ sơ cá nhân (tên, email, mật khẩu,…).
* **Đăng bài ký gửi căn hộ:** Người dùng đăng tin rao bán/cho thuê căn hộ của mình (nhập thông tin, tải ảnh lên). Sau khi đăng, căn hộ sẽ ở trạng thái chờ duyệt.
* **Xem bài đăng của tôi:** Người dùng xem danh sách các bài đăng căn hộ mà mình đã gửi (kèm trạng thái duyệt/chưa duyệt).
* **Liên hệ (Contact):** Gửi thông tin liên hệ/ góp ý tới ban quản trị. Người dùng không cần đăng nhập vẫn có thể gửi liên hệ.
* **Chat trực tuyến:** Cho phép người dùng trò chuyện trực tiếp với admin (hỗ trợ real-time qua websocket).

**Chức năng trang quản trị (Admin):**

* **Dashboard (Trang chủ admin):** Trang tổng quan hiển thị thông báo mới và thống kê (ví dụ: số căn hộ, số người dùng, liên hệ mới…).
* **Quản lý tài khoản người dùng:** Xem danh sách tài khoản đã đăng ký. Thực hiện các hành động: **chặn/mở khóa tài khoản**, **xóa tài khoản**, **cấp/gỡ quyền admin** cho tài khoản. (Có hỗ trợ tìm kiếm tài khoản theo tên, SĐT).
* **Quản lý căn hộ hệ thống:** Quản lý danh sách căn hộ do hệ thống sở hữu (công ty đăng bán). Admin có thể xem, lọc theo tiêu đề, và xóa căn hộ khi cần.
* **Quản lý căn hộ người dùng:** Quản lý danh sách căn hộ do người dùng đăng đã được duyệt. Có thể lọc, xem chi tiết từng căn hộ người dùng đã duyệt, hoặc gỡ bỏ nếu cần.
* **Quản lý bài đăng chờ duyệt:** Xem danh sách các bài đăng căn hộ do người dùng gửi lên chưa được duyệt. Admin xem **chi tiết bài đăng** và **phê duyệt hoặc từ chối** bài đăng đó. Sau khi duyệt, căn hộ sẽ xuất hiện trong danh sách căn hộ người dùng.
* **Tạo chủ sở hữu mới:** Thêm thông tin chủ sở hữu mới (dành cho căn hộ hệ thống). Chức năng này phục vụ việc quản lý các chủ sở hữu căn hộ thuộc hệ thống.
* **Quản lý chủ sở hữu:** Xem danh sách tất cả các chủ sở hữu căn hộ trong hệ thống, chỉnh sửa thông tin chủ sở hữu hoặc thêm mới (nếu chưa thực hiện ở bước tạo mới).
* **Tạo hợp đồng mới:** Lập hợp đồng mua bán cho một căn hộ khi giao dịch thành công. Admin chọn căn hộ, người mua, người bán (đối với căn hộ ký gửi thì người bán chính là chủ sở hữu do user đăng) và nhập các điều khoản để tạo hợp đồng.
* **Quản lý hợp đồng:** Xem danh sách hợp đồng đã lập. Admin có thể **xem chi tiết/in hợp đồng**, chỉnh sửa thông tin hợp đồng trước khi in nếu cần.
* **Quản lý giao dịch bán:** (Sales) Theo dõi danh sách các giao dịch bán căn hộ đã hoàn tất. Chức năng này liên quan đến hợp đồng – sau khi hợp đồng được ký, hệ thống ghi nhận một giao dịch bán thành công.
* **Quản lý đánh giá:** Xem danh sách các phản hồi/đánh giá của người dùng về hệ thống hoặc về căn hộ. Admin có thể theo dõi nội dung đánh giá và xử lý (ẩn/xóa nếu không phù hợp).
* **Quản lý liên hệ:** Xem danh sách các liên hệ do người dùng gửi từ trang Liên hệ. Admin đánh dấu trạng thái đã xử lý liên hệ hoặc phản hồi ngoài hệ thống.
* **Quản lý chat:** Giao diện chat dành cho admin để trao đổi tin nhắn real-time với người dùng.
* **Quản lý dịch vụ đăng tin:** Quản lý các **gói dịch vụ** bổ sung mà người dùng có thể mua (ví dụ: gói đăng tin nổi bật, quảng cáo căn hộ). Admin có thể tạo mới các dịch vụ, quy định thời hạn hiệu lực (duration) cho mỗi gói.
* **Quản lý giao dịch thanh toán:** Xem danh sách các giao dịch nạp tiền mua dịch vụ của người dùng. Theo dõi số tiền nạp và gói dịch vụ mà người dùng đã mua.
* **Đăng nhập bằng Google:** Cho phép người dùng đăng nhập thông qua tài khoản Google.
* **Hỗ trợ đa ngôn ngữ:** Website hỗ trợ song ngữ Anh – Việt (người dùng có thể chuyển đổi ngôn ngữ giao diện).
* **Nhật ký hệ thống:** Ghi lại log hoạt động (ví dụ: đăng nhập, lỗi) để quản trị viên theo dõi (chức năng nội bộ, không có giao diện riêng cho user).

## **Chức năng hệ thống và các lớp liên quan**

### **1. Quản lý tài khoản (Account Management)**

**Mô tả chức năng:** Hỗ trợ đăng ký, đăng nhập, tìm kiếm và cập nhật thông tin tài khoản người dùng. Tài khoản có thể được liên kết với Gmail và có trạng thái xác thực.

**Các lớp liên quan và nhiệm vụ:**

* **dao.AccountDAO**
  + register(Account account): Tạo tài khoản mới.
  + login(username, password): Kiểm tra thông tin đăng nhập.
* **models.Account**
  + Đại diện cho tài khoản với các thông tin như username, password, email, status, role.
* **dao.AccountGmailDAO**
  + findUserByGmail(email): Kiểm tra tài khoản Gmail liên kết.
* **models.ConnectDB**
  + Hỗ trợ kết nối với cơ sở dữ liệu.

### **2. Quản lý chi tiết tài khoản (Account Details Management)**

**Mô tả chức năng:** Lưu trữ thông tin cá nhân của tài khoản như tên, địa chỉ, số điện thoại, số dư tài khoản.

**Các lớp liên quan và nhiệm vụ:**

* **dao.AccountDetailsDAO**
  + create(Accountdetails accountdetails): Thêm thông tin chi tiết tài khoản.
  + updateBalance(Accountdetails accountdetails): Cập nhật số dư tài khoản.
* **models.Accountdetails**
  + Đại diện cho thông tin tài khoản cá nhân.

### **3. Quản lý tài khoản Gmail liên kết (Account Gmail Management)**

**Mô tả chức năng:** Quản lý tài khoản Gmail được liên kết với tài khoản hệ thống.

**Các lớp liên quan và nhiệm vụ:**

* **dao.AccountGmailDAO**
  + findUserByGmail(email): Tìm tài khoản Gmail liên kết.
  + delete(id): Xóa tài khoản Gmail.
* **models.AccountGmail**
  + Đại diện cho tài khoản Gmail.

### **4. Quản lý hợp đồng thuê căn hộ (Contract Management)**

**Mô tả chức năng:** Lưu trữ và quản lý hợp đồng thuê căn hộ giữa chủ sở hữu và người thuê.

**Các lớp liên quan và nhiệm vụ:**

* **dao.ContractDAO**
  + create(Contract contract): Tạo hợp đồng thuê mới.
  + checkExists(contractID): Kiểm tra tính hợp lệ của hợp đồng.
* **models.Contract**
  + Đại diện cho hợp đồng thuê căn hộ.
* **dao.SystemApartmentDAO**
  + findSystemApartmentByID(id): Kiểm tra thông tin căn hộ.

### **5. Quản lý dịch vụ tài khoản (Account Service Management)**

**Mô tả chức năng:** Theo dõi dịch vụ mà tài khoản đã đăng ký như internet, bảo vệ, điện nước.

**Các lớp liên quan và nhiệm vụ:**

* **dao.AccountServiceDAO**
  + register(AccountService accountService): Đăng ký dịch vụ.
  + findByServiceID(serviceID): Lấy danh sách dịch vụ của tài khoản.
* **models.AccountService**
  + Đại diện cho dịch vụ tài khoản đăng ký.

### **6. Quản lý chi nhánh (Branch Management)**

**Mô tả chức năng:** Lưu trữ thông tin về các chi nhánh của hệ thống.

**Các lớp liên quan và nhiệm vụ:**

* **dao.BranchDAO**
  + findBranchByID(branchID): Tìm chi nhánh theo ID.
* **models.Branch**
  + Đại diện cho một chi nhánh.

### **7. Quản lý tin nhắn hỗ trợ (Chat Management)**

**Mô tả chức năng:** Lưu trữ tin nhắn giữa khách hàng và quản trị viên.

**Các lớp liên quan và nhiệm vụ:**

* **dao.ChatDAO**
  + newChat(Chat chat): Lưu tin nhắn vào database.
  + findChatByUserID(userID): Lấy lịch sử tin nhắn của người dùng.
* **models.Chat**
  + Đại diện cho tin nhắn giữa khách hàng và admin.

### **8. Quản lý phản hồi và liên hệ (Feedback & Contact Management)**

**Mô tả chức năng:** Xử lý phản hồi và liên hệ từ khách hàng.

**Các lớp liên quan và nhiệm vụ:**

* **dao.FeedbackDAO**
  + submitFeedback(Feedback feedback): Lưu phản hồi của khách hàng.
* **dao.ContactDAO**
  + update(Contact contact): Cập nhật trạng thái liên hệ.
* **models.Feedback, models.Contact**
  + Đại diện cho phản hồi và liên hệ của khách hàng.

### **9. Quản lý hóa đơn (Invoice Management)**

**Mô tả chức năng:** Quản lý hóa đơn dịch vụ của khách hàng.

**Các lớp liên quan và nhiệm vụ:**

* **dao.InvoiceDAO**
  + register(Invoice invoice): Tạo hóa đơn mới.
  + findByID(invoiceID): Lấy hóa đơn theo ID.
* **models.Invoice**
  + Đại diện cho hóa đơn dịch vụ.

### **10. Quản lý nhật ký hệ thống (Log Management)**

**Mô tả chức năng:** Ghi lại lịch sử hoạt động của hệ thống.

**Các lớp liên quan và nhiệm vụ:**

* **dao.LogDAO**
  + create(Log log): Ghi nhật ký vào database.
  + findAll(): Lấy danh sách nhật ký hệ thống.
* **models.Log**
  + Đại diện cho một bản ghi nhật ký hệ thống.

### **11. Quản lý chủ sở hữu căn hộ (Owner Management)**

**Mô tả chức năng:** Lưu trữ thông tin về chủ sở hữu căn hộ.

**Các lớp liên quan và nhiệm vụ:**

* **dao.OwnerDAO**
  + findById(ownerID): Tìm chủ sở hữu căn hộ theo ID.
* **models.Owner**
  + Đại diện cho chủ sở hữu căn hộ.

### **12. Quản lý thanh toán (Payment Management)**

**Mô tả chức năng:** Xử lý các giao dịch thanh toán trong hệ thống, bao gồm lưu trữ thông tin thanh toán và kết nối với cổng thanh toán VNPay.

**Các lớp liên quan và nhiệm vụ:**

* **dao.TransactionDAO**
  + create(transaction): Thêm giao dịch mới vào bảng transaction.
  + findAll(): Lấy danh sách tất cả giao dịch.
* **models.Transaction**
  + Đại diện cho giao dịch với các thuộc tính như id, type, price, date, accountID, orderInfo, paymentType, transactionNo.
* **service.Config**
  + hashAllFields(fields): Mã hóa dữ liệu gửi đến VNPay.
  + hmacSHA512(key, data): Mã hóa dữ liệu với thuật toán HMAC-SHA512.

### **13. Quản lý tải lên file trong tin nhắn (Chat File Upload)**

**Mô tả chức năng:** Cho phép người dùng tải lên file trong tin nhắn khi trò chuyện với quản trị viên.

**Các lớp liên quan và nhiệm vụ:**

* **controller.UploadFileChatServlet**
  + doPost(request, response): Xử lý yêu cầu tải file lên server.
  + extractFileName(filePart): Lấy tên file từ dữ liệu tải lên.
* **service.FileHelper**
  + generateFileName(fileName): Sinh tên file ngẫu nhiên để tránh trùng lặp.

**Lớp hỗ trợ khác:**

* **service.MailHelper**: Hỗ trợ gửi email qua SMTP.
* **service.IPAddressUtil**: Lấy địa chỉ IP công cộng của người dùng.

### 

### **Management Application Apartment**

### **Phân Chia Công Việc và Danh Sách Chức Năng**

* **Ngày hiện tại:** 25/03/2025

**Nhóm thực hiện:**

* Phan Thị Ngọc Quyên
* Nguyễn Song Gia Huy
* Lê Hữu Nguyên Khang
* Lê Tự Thanh Huy
* Nguyễn Phương Thảo
* Lê Ngọc Thiên

### **1. Overview**

Ứng dụng "Management Application Apartment" là một hệ thống quản lý căn hộ trực tuyến, hỗ trợ người dùng đăng ký, quản lý thông tin cá nhân, bài đăng căn hộ, dịch vụ, hợp đồng, và các tính năng nâng cao như chat, thanh toán. Admin và superadmin có thể quản lý tài khoản, bài đăng, dịch vụ, doanh thu, phản hồi, và hợp đồng. Dự án được phát triển bởi nhóm 6 thành viên, kết hợp Frontend (JSP, HTML, CSS, JS) và Backend (Java Servlet, DAO).

### **2. Nguyên Tắc Phân Chia Công Việc**

* **Tính liên quan:** Các chức năng trong cùng luồng nghiệp vụ (ví dụ: quản lý tài khoản, quản lý dịch vụ) được giao cho cùng một người.
* **Cân bằng khối lượng:** Mỗi thành viên đảm nhận 2-3 nhóm chức năng hoặc 4-5 chức năng nhỏ, tùy độ phức tạp.
* **Kết hợp Frontend và Backend:** Mỗi người chịu trách nhiệm cả giao diện (JSP) và logic (Servlet, DAO).
* **Ưu tiên:** Chức năng nền tảng (đăng nhập, tài khoản) và chức năng quản lý cốt lõi được phân bổ đều.

### **3. Phân Công Cụ Thể Cho Từng Thành Viên**

#### **Thành viên 1: Phan Thị Ngọc Quyên**

**Chức năng được giao:**

1. **Chức năng đăng nhập và từ chối truy cập**
   * **Backend:** LoginServlet (xác thực tài khoản, phân quyền admin/user), AccessDeniedAdminServlet (xử lý từ chối truy cập).
   * **Frontend:** login.jsp, accessdenied.jsp.
   * **Công nghệ:** Java Servlet, JSP, BCrypt (mã hóa mật khẩu).
2. **Chức năng gửi email xác nhận OTP**
   * **Backend:** EmailServlet (gửi OTP), OTPServlet (xác nhận OTP).
   * **Frontend:** email.jsp, otp.jsp.
   * **Công nghệ:** JavaMail API, RandomStringHelper.
3. **Chức năng thay đổi mật khẩu**
   * **Backend:** ChangePasswordServlet (xử lý cập nhật mật khẩu).
   * **Frontend:** changePassword.jsp.
   * **Công nghệ:** Java Servlet, JSP, BCrypt, AccountDAO.

**Lý do:**

* Các chức năng liên quan đến xác thực người dùng (đăng nhập, OTP, mật khẩu) có tính kết nối chặt chẽ.
* Khối lượng vừa phải, phù hợp với thành viên có kinh nghiệm về luồng xác thực.

#### **Thành viên 2: Nguyễn Song Gia Huy**

**Chức năng được giao:**

1. **Chức năng quản lý truy cập (Access Control)**
   * **Backend:** AdminFilter (kiểm tra quyền admin), SuperAdminFilter (kiểm tra quyền superadmin).
   * **Frontend:** Chuyển hướng đến login.jsp khi không đủ quyền.
   * **Công nghệ:** Java Servlet Filter, HttpSession.
2. **Chức năng quản lý tài khoản (Account Management - phía người dùng)**
   * **Backend:** AccountServlet (cập nhật thông tin cá nhân, upload ảnh).
   * **Frontend:** account.jsp.
   * **Công nghệ:** Java Servlet, JSP, MultipartConfig, BCrypt, AccountDAO.
3. **Chức năng thay đổi ngôn ngữ**
   * **Backend:** LanguageServlet (lưu ngôn ngữ vào session).
   * **Frontend:** home.jsp (hiển thị giao diện đa ngôn ngữ).
   * **Công nghệ:** Java Servlet, JSP, PostDAO.

**Lý do:**

* Quản lý truy cập là nền tảng bảo mật, kết hợp tốt với quản lý tài khoản người dùng.
* Thay đổi ngôn ngữ là chức năng nhẹ, bổ sung để cân bằng khối lượng.

#### **Thành viên 3: Lê Tự Thanh Huy**

**Chức năng được giao:**

1. **Chức năng quản lý bài đăng (Post Management)**
   * **Backend:** PostApartmentAdminServlet (xác thực, xóa bài đăng), DashboardAdminServlet (tổng quan bài đăng), UserApartmentAdminServlet (quản lý căn hộ người dùng).
   * **Frontend:** postapartment.jsp, dashboard.jsp, userapartment.jsp.
   * **Công nghệ:** Java Servlet, JSP, Gson, PostDAO, PostImageDAO.
2. **Chức năng tự động hoàn thành (Autocomplete)**
   * **Backend:** AutocompleteServlet (tìm kiếm chủ sở hữu theo từ khóa).
   * **Frontend:** Gợi ý từ khóa qua Ajax.
   * **Công nghệ:** Java Servlet, Gson, OwnerDAO.
3. **Quản lý thanh toán (Payment Management)  
   Backend: PaymentServlet (xử lý thanh toán VNPay).**

**Frontend: payment.jsp.**

**Công nghệ: Java Servlet, VNPay API, TransactionDAO.**

**Lý do:**

* Quản lý bài đăng là chức năng lớn, liên quan đến dashboard và căn hộ người dùng.
* Autocomplete bổ sung tính năng tìm kiếm, phù hợp với luồng quản lý bài đăng.

#### **Thành viên 4: Nguyễn Phương Thảo**

**Chức năng được giao:**

1. **Chức năng quản lý dịch vụ (Service Management)**
   * **Backend:** ServiceAdminServlet (thêm, xóa gói dịch vụ), ServiceAccountAdminServlet (đăng ký, hủy dịch vụ), AccountServiceAdminServlet (hủy dịch vụ tài khoản).
   * **Frontend:** service.jsp, serviceAccount.jsp, newService.jsp.
   * **Công nghệ:** Java Servlet, JSP, ServiceDAO, AccountServiceDAO.
2. **Chức năng quản lý thời hạn dịch vụ (Duration Management)**
   * **Backend:** DurationAdminServlet (thêm, xóa thời hạn).
   * **Frontend:** duration.jsp, newDuration.jsp.
   * **Công nghệ:** Java Servlet, JSP, DurationDAO.
3. **Chức năng quản lý hợp đồng (Contract Management)**
   * **Backend:** NewContractApartmentAdminServlet (tạo hợp đồng), NewContractUserApartmentAdminServlet (thêm chủ sở hữu), OwnersAdminServlet (quản lý chủ sở hữu), ContractDetailsAdminServlet, EditContractAdminServlet (xem, sửa hợp đồng).
   * **Frontend:** newcontract.jsp, newcontractuser.jsp, owners.jsp.
   * **Công nghệ:** Java Servlet, JSP, MultipartConfig, ContractDAO, OwnerDAO.

**Lý do:**

* Quản lý dịch vụ và thời hạn có liên quan chặt chẽ, tạo thành một nhóm chức năng hoàn chỉnh.
* Khối lượng vừa phải, phù hợp với việc phát triển song song.

#### **Thành viên 5: Lê Ngọc Thiên**

**Chức năng được giao:**

1. **Chức năng quản lý tài khoản (Account Management - phía admin/superadmin)**
   * **Backend:** AccountAdminServlet (tìm kiếm, chặn, cấp/gỡ quyền admin).
   * **Frontend:** accounts.jsp.
   * **Công nghệ:** Java Servlet, JSP, Gson, BCrypt, AccountDAO.
2. **Chức năng danh sách yêu thích (Wishlist)**
   * **Backend:** WishlistApartmentServlet (thêm, xóa bài đăng yêu thích).
   * **Frontend:** wishlist.jsp.
   * **Công nghệ:** Java Servlet, JSP, PostDAO.
3. **Chức năng quản lý doanh thu (Sale Management)**
   * **Backend:** SaleAdminServlet (tìm kiếm, xóa lịch sử doanh thu).
   * **Frontend:** sales.jsp.
   * **Công nghệ:** Java Servlet, JSP, SaleDAO.

**Lý do:**

* Quản lý tài khoản phía admin là chức năng lớn, kết hợp với wishlist và tải tin nhắn để cân bằng khối lượng.
* Các chức năng đều liên quan đến trải nghiệm người dùng.

#### **Thành viên 6: Lê Hữu Nguyên Khang**

**Chức năng được giao:**

1. **Chức năng quản lý hợp đồng (Contract Management)**
   * **Backend:** NewContractApartmentAdminServlet (tạo hợp đồng), NewContractUserApartmentAdminServlet (thêm chủ sở hữu), OwnersAdminServlet (quản lý chủ sở hữu), ContractDetailsAdminServlet, EditContractAdminServlet (xem, sửa hợp đồng).
   * **Frontend:** newcontract.jsp, newcontractuser.jsp, owners.jsp.
   * **Công nghệ:** Java Servlet, JSP, MultipartConfig, ContractDAO, OwnerDAO.
2. **Chức năng quản lý doanh thu (Sale Management)**
   * **Backend:** SaleAdminServlet (tìm kiếm, xóa lịch sử doanh thu).
   * **Frontend:** sales.jsp.
   * **Công nghệ:** Java Servlet, JSP, SaleDAO.
3. **Chức năng quản lý email (Email Receiver)**
   * **Backend:** ReceiverEmailAdminServlet (nhận, xem email).
   * **Frontend:** receiverEmail.jsp, emailDetails.jsp.
   * **Công nghệ:** Java Servlet, JavaMail API, JSP.

**Lý do:**

* Quản lý hợp đồng là chức năng phức tạp, kết hợp với doanh thu và email tạo thành nhóm chức năng quản lý tài chính và thông tin.
* Đảm bảo khối lượng công việc tương đương với các thành viên khác.

### **4. Tổng Quan Công Nghệ**

* **Framework:** Java Servlet, JSP.
* **Thư viện:** Gson (JSON), BCrypt (mã hóa), JavaMail (email), MultipartConfig (upload file).
* **Mô hình MVC:**
  + **Model:** Các lớp models (Account, Post, Service, v.v.) và DAO.
  + **View:** JSP trong /WEB-INF/views/.
  + **Controller:** Servlet trong package controller và controller.admin.
* **Luồng hoạt động:** Filter kiểm tra quyền → Servlet xử lý logic → DAO truy vấn dữ liệu → JSP hiển thị.

# Assignment-JAVA-WEB

# **MANAGEMENT APPLICATION APARTMENT**

# 1. Overview

Dự án **Management Application Apartment** là một hệ thống quản lý căn hộ trực tuyến, hỗ trợ người dùng đăng ký tài khoản, quản lý thông tin cá nhân, đăng bài ký gửi căn hộ, quản lý dịch vụ, hợp đồng, và sử dụng các tính năng nâng cao như chat trực tuyến, thanh toán. Đối với admin và superadmin, hệ thống cung cấp các công cụ để quản lý tài khoản, bài đăng, dịch vụ, doanh thu, phản hồi và hợp đồng. Dự án được phát triển bởi nhóm 6 thành viên, kết hợp giữa Frontend (JSP, HTML, CSS, JS) và Backend (Java Servlet, DAO).

## Members:

* **Phan Thị Ngọc Quyên** - Leader
* **Nguyễn Song Gia Huy**
* **Lê Hữu Nguyên Khang**
* **Lê Tự Thanh Huy**
* **Nguyễn Phương Thảo**
* **Lê Ngọc Thiện**

#### Các chức năng chính:

* **Frontend**:
  + Hiển thị trang chủ với giới thiệu website và danh sách căn hộ mới nhất.
  + Xem danh sách căn hộ (căn hộ hệ thống và căn hộ ký gửi), chi tiết từng căn hộ.
  + Tìm kiếm/lọc căn hộ theo tiêu đề, địa điểm, với gợi ý từ khóa (autocomplete).
  + Quản lý danh sách yêu thích (lưu và xem lại căn hộ đã đánh dấu).
  + Trang tài khoản người dùng (xem và cập nhật thông tin cá nhân).
  + Đăng bài ký gửi căn hộ (nhập thông tin, tải ảnh, trạng thái chờ duyệt).
  + Xem bài đăng của tôi (danh sách bài đăng kèm trạng thái duyệt).
  + Gửi thông tin liên hệ/góp ý (không cần đăng nhập).
  + Chat trực tuyến với admin (real-time qua websocket).
  + Đăng nhập bằng Google và hỗ trợ đa ngôn ngữ (Anh - Việt).
* **Backend**:
  + Quản lý tài khoản người dùng (đăng ký, đăng nhập, xác thực qua email, cập nhật thông tin).
  + Quản lý căn hộ hệ thống và căn hộ ký gửi (CRUD: Create, Read, Update, Delete).
  + Quản lý bài đăng chờ duyệt (phê duyệt/từ chối).
  + Quản lý hợp đồng và giao dịch bán (tạo hợp đồng, theo dõi doanh thu).
  + Quản lý dịch vụ đăng tin và giao dịch thanh toán (gói dịch vụ, nạp tiền).
  + Xử lý chat, phản hồi, liên hệ từ người dùng.
  + Quản lý phân quyền (admin, user) và nhật ký hệ thống (log hoạt động).
* **Database**:
  + Lưu trữ thông tin tài khoản (Account, Accountdetails), căn hộ (SystemApartment, Post), hợp đồng (Contract), dịch vụ (Service, AccountService), giao dịch (Transaction), liên hệ (Contact), phản hồi (Feedback), và nhật ký (Log).
  + Sử dụng bảng trung gian để quản lý quan hệ nhiều-nhiều (ví dụ: AccountService, Wishlist).
  + Thuộc tính trạng thái (status) để theo dõi bài đăng, hợp đồng, giao dịch.

#### Công nghệ sử dụng:

* **Backend**: Java, Servlet, JSP, Maven, JDBC, BCrypt (mã hóa mật khẩu), JavaMail (email), Gson (JSON), Websocket (chat).
* **Frontend**: JSP, HTML, CSS, JavaScript, Ajax (autocomplete, load more).
* **Database**: MySQL.

# 2. Getting Started

#### Yêu cầu hệ thống:

* **Máy chủ**: Java Runtime Environment (JRE) 8+, Apache Tomcat.
* **Môi trường phát triển**: JDK 8+, Maven, IDE (IntelliJ IDEA, Eclipse).
* **Frontend**: Trình duyệt web hiện đại (Chrome, Firefox).
* **Database**: MySQL 5.7+.

#### Quy trình phát triển:

* **Phân tích yêu cầu**: Xác định chức năng (đăng ký, quản lý căn hộ, chat).
* **Thiết kế cơ sở dữ liệu**: Tạo ERD và script SQL cho bảng dữ liệu.
* **Phát triển backend**: Xây dựng logic bằng Servlet, DAO, xử lý yêu cầu HTTP.
* **Phát triển frontend**: Thiết kế giao diện JSP, tích hợp Ajax cho tính năng động.
* **Tích hợp**: Kết nối frontend-backend qua Servlet và JSP.
* **Kiểm thử**: Kiểm tra chức năng (đăng nhập, tìm kiếm, duyệt bài).
* **Triển khai**: Cài đặt trên Tomcat với MySQL làm cơ sở dữ liệu.

#### Ưu tiên:

* Đảm bảo bảo mật (mã hóa mật khẩu, xác thực OTP).
* Tối ưu hóa tìm kiếm/lọc căn hộ và chat real-time.
* Ưu tiên công cụ quản trị (quản lý tài khoản, bài đăng).
* Hỗ trợ mở rộng cho số lượng người dùng và căn hộ tăng dần.

# 3. Database Design

(Phần này có thể được bổ sung sau nếu bạn cung cấp sơ đồ ERD cụ thể.)

# 4. Use Cases

## Use-case 1: Sign Up

**Mô tả**: Cho phép người dùng tạo tài khoản mới để truy cập hệ thống.

**Input**:

* Username
* Password
* Email

**Quy trình xử lý**:

1. Kiểm tra username và email:
   * Nếu đã tồn tại: Thông báo "Tài khoản đã tồn tại" và yêu cầu nhập lại.
   * Nếu chưa tồn tại: Tiếp tục bước tiếp theo.
2. Gửi email xác thực:
   * Tạo mã OTP qua RandomStringHelper, gửi qua MailHelper.
   * Người dùng nhập OTP tại /otp, nếu khớp thì kích hoạt tài khoản (verify = true).

**Output**:

* Thông báo "Đăng ký thành công" và chuyển hướng đến trang đăng nhập.

## Use-case 2: Sign In

**Mô tả**: Người dùng đăng nhập vào hệ thống bằng tài khoản đã đăng ký.

**Input**:

* Username
* Password

**Quy trình xử lý**:

1. Kiểm tra thông tin đăng nhập:
   * Nếu đúng: Lưu Account vào session, kiểm tra vai trò (role).
   * Nếu sai: Hiển thị "Sai username hoặc mật khẩu", tải lại trang.
2. Phân quyền:
   * role = admin: Chuyển đến dashboard admin.
   * role = user: Chuyển đến trang chủ người dùng.

**Output**:

* Chuyển hướng đến trang tương ứng theo vai trò.

## Use-case 3: Forgot Password

**Mô tả**: Hỗ trợ người dùng khôi phục mật khẩu qua email.

**Input**:

* Email

**Quy trình xử lý**:

1. Kiểm tra email:
   * Nếu tồn tại: Gửi OTP qua EmailServlet.
   * Nếu không: Hiển thị "Email không tồn tại".
2. Xác nhận OTP:
   * Nếu khớp: Chuyển đến trang đổi mật khẩu.
   * Nếu sai: Hiển thị "OTP không hợp lệ".
3. Đổi mật khẩu:
   * Nếu "confirm password" khớp: Cập nhật qua AccountDAO, thông báo "Đổi mật khẩu thành công".
   * Nếu không khớp: Hiển thị "Xác nhận mật khẩu không đúng".

**Output**:

* Thông báo "Đổi mật khẩu thành công" hoặc lỗi tương ứng.

## Use-case 4: Post Apartment

**Mô tả**: Người dùng đăng bài ký gửi căn hộ để bán/cho thuê.

**Input**:

* Tiêu đề
* Địa điểm
* Mô tả
* Ảnh (upload)

**Quy trình xử lý**:

1. Kiểm tra thông tin:
   * Nếu thiếu: Hiển thị "Vui lòng nhập đầy đủ thông tin".
   * Nếu đủ: Lưu bài đăng vào Post với trạng thái "chờ duyệt".
2. Upload ảnh:
   * Sử dụng UploadFileHelper để lưu ảnh vào server.

**Output**:

* Chuyển hướng đến trang "Bài đăng của tôi" với thông báo "Đăng bài thành công, chờ duyệt".

## Use-case 5: Search/Filter Apartments

**Mô tả**: Người dùng tìm kiếm và lọc căn hộ theo tiêu chí.

**Input**:

* Tiêu đề
* Địa điểm
* Loại căn hộ (hệ thống/ký gửi)

**Quy trình xử lý**:

1. Nhập tiêu chí tìm kiếm:
   * Gợi ý từ khóa qua AutocompleteServlet.
2. Lọc và hiển thị:
   * Truy vấn PostDAO hoặc SystemApartmentDAO, trả về danh sách phù hợp.

**Output**:

* Hiển thị danh sách căn hộ theo bộ lọc.

## Use-case 6: Manage Wishlist

**Mô tả**: Người dùng lưu và quản lý danh sách căn hộ yêu thích.

**Input**:

* ID căn hộ

**Quy trình xử lý**:

1. Thêm vào yêu thích:
   * Nhấn nút "Thêm", lưu vào Wishlist qua WishlistApartmentServlet.
2. Xem danh sách:
   * Hiển thị các căn hộ đã lưu từ PostDAO.

**Output**:

* Danh sách căn hộ yêu thích được cập nhật.

## Use-case 7: Approve Post

**Mô tả**: Admin duyệt bài đăng căn hộ ký gửi từ người dùng.

**Input**:

* Chấp nhận/Từ chối (button)

**Quy trình xử lý**:

1. Truy cập danh sách chờ duyệt:
   * Lấy từ PostDAO các bài có trạng thái "chờ duyệt".
2. Quyết định:
   * Chấp nhận: Chuyển trạng thái thành "đã duyệt".
   * Từ chối: Xóa bài đăng, thông báo cho người dùng.

**Output**:

* Thông báo "Đã duyệt" hoặc "Bị từ chối" gửi đến người dùng.

## Use-case 8: Create Contract

**Mô tả**: Admin tạo hợp đồng mua bán căn hộ khi giao dịch thành công.

**Input**:

* ID căn hộ
* Người mua
* Người bán
* Điều khoản

**Quy trình xử lý**:

1. Nhập thông tin:
   * Chọn căn hộ từ SystemApartmentDAO hoặc PostDAO.
2. Lưu hợp đồng:
   * Gọi ContractDAO.create() để lưu vào cơ sở dữ liệu.

**Output**:

* Thông báo "Hợp đồng đã được tạo".

## Use-case 9: Chat with Admin

**Mô tả**: Người dùng trò chuyện real-time với admin.

**Input**:

* Tin nhắn (văn bản/file)

**Quy trình xử lý**:

1. Gửi tin nhắn:
   * Sử dụng Websocket qua ChatDAO.
2. Upload file:
   * Xử lý qua UploadFileChatServlet.

**Output**:

* Tin nhắn hiển thị real-time trên giao diện chat.

## Use-case 10: Manage Services

**Mô tả**: Admin quản lý các gói dịch vụ đăng tin.

**Input**:

* Tên dịch vụ
* Thời hạn (duration)
* Giá

**Quy trình xử lý**:

1. Thêm dịch vụ:
   * Lưu vào ServiceDAO.
2. Cập nhật thời hạn:
   * Sử dụng DurationAdminServlet.

**Output**:

* Danh sách dịch vụ được cập nhật.

# 5. How to Install the Project

* **Tải mã nguồn**:
  + git clone <[git@github.com](mailto:git@github.com):detne/Assignment-JAVA-WEB.git>
* **Thiết lập cơ sở dữ liệu**:
  + Tạo database: CREATE DATABASE apartment\_management;
  + Chạy script SQL để tạo bảng và dữ liệu mẫu.
* **Thiết lập Backend**:
  + Mở dự án trong IDE (IntelliJ/Eclipse).
  + Cấu hình Maven và Tomcat, cập nhật kết nối MySQL trong ConnectDB.
* **Thiết lập Frontend**:
  + JSP nằm trong /WEB-INF/views/, không cần cài đặt riêng.
* **Chạy ứng dụng**:
  + Triển khai trên Tomcat, truy cập qua <http://localhost:8080>.