TÀI LIỆU ĐẶC TẢ YÊU CẦU PHẦN MỀM (SRS)

**Tên dự án**: Hệ thống quản lý tuyển dụng cho công ty IT

**Công nghệ sử dụng**: Java Core, JDBC, MySQL

**Loại ứng dụng:** Ứng dụng console

# Use Case

## 1.1 Chức năng Đăng ký (Register)

## Mô tả: Cho phép ứng viên mới tạo tài khoản trong hệ thống.

## Quy trình:

## Hệ thống yêu cầu ứng viên nhập thông tin cơ bản: tên, email, mật khẩu

## Hệ thống kiểm tra tính hợp lệ của dữ liệu thông qua ValidateCandidateAuthen:

## Tên phải từ 3-100 ký tự

## Email phải đúng định dạng và chưa tồn tại trong hệ thống

## Mật khẩu phải có ít nhất 8 ký tự

## Nếu dữ liệu hợp lệ, hệ thống tạo ứng viên mới trong cơ sở dữ liệu thông qua thủ tục proc\_candidates\_create

## Trạng thái mặc định của tài khoản mới là "active"

## 1.2 Chức năng Đăng nhập (Login)

**Mô tả:** Cho phép ứng viên đã đăng ký đăng nhập vào hệ thống.

**Quy trình:**

* Hệ thống kiểm tra xem đã có phiên đăng nhập hay chưa thông qua **hasActiveSession()**
* Nếu chưa có phiên đăng nhập, hệ thống yêu cầu ứng viên nhập email và mật khẩu
* Hệ thống kiểm tra thông tin đăng nhập thông qua **findByEmailAndPassword()**
* Nếu thông tin chính xác, hệ thống kiểm tra trạng thái tài khoản:
  + Nếu tài khoản "inactive", thông báo tài khoản bị khóa
  + Nếu tài khoản hoạt động, tạo phiên đăng nhập và lưu thông tin vào file session
* Phiên đăng nhập được lưu vào file **candidate\_session.tx**t

## 1.3 Chức năng Đăng xuất (Logout)

**Mô tả:** Cho phép ứng viên đăng xuất khỏi hệ thống.

**Quy trình:**

Hệ thống xóa thông tin ứng viên hiện tại bằng cách đặt **currentCandidate = null**

Hệ thống xóa file phiên đăng nhập **candidate\_session.txt**

Hiển thị thông báo đăng xuất thành công

## 2. Quản lí công nghệ

## 2.1 Hiển thị các công nghệ tuyển dụng

### Chức năng này liệt kê tất cả các công nghệ đang được sử dụng trong hệ thống tuyển dụng.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Xem danh sách công nghệ" trong menu

### Hệ thống gọi phương thức **getAllActiveTechnologies()** từ **TechnologyService**

### Dữ liệu được truy xuất từ database thông qua stored **procedure sp\_GetAllTechnologies()**

### Hệ thống lọc các công nghệ đã bị xóa mềm (những công nghệ có tên kết thúc bằng **"\_deleted"**)

### Hiển thị danh sách các công nghệ còn hoạt động theo định dạng: ID và Tên công nghệ

## 2.2 Thêm công nghệ mới

### Chức năng này cho phép thêm một công nghệ mới vào hệ thống.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Thêm công nghệ mới" trong menu

### Hệ thống yêu cầu nhập tên công nghệ mới

### Kiểm tra tính hợp lệ của tên công nghệ:

### Không được để trống

### Tên phải duy nhất (không trùng với công nghệ hiện có)

### Nếu hợp lệ, hệ thống gọi phương thức **addTechnology()** từ **TechnologyService**

### Dữ liệu được lưu vào database thông qua stored **procedure sp\_AddTechnology()**

### Hiển thị thông báo thành công hoặc thất bại

## 2.3 Xóa công nghệ theo ID

### Chức năng này cho phép xóa một công nghệ khỏi hệ thống dựa trên ID.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Xóa công nghệ" trong menu

### Hệ thống yêu cầu nhập ID công nghệ cần xóa

### Kiểm tra xem công nghệ có tồn tại không bằng cách gọi **getTechnologyById()**

### Yêu cầu xác nhận xóa từ người dùng

### Nếu người dùng xác nhận, hệ thống gọi phương thức **deleteTechnology()** từ **TechnologyService**

### Xử lý xóa công nghệ có 2 trường hợp:

### Xóa cứng: Nếu công nghệ không liên kết với bảng khác, sẽ bị xóa hoàn toàn

### Xóa mềm: Nếu công nghệ đã liên kết với các bảng khác **(candidate\_technology, recruitment\_position\_technology)**, tên công nghệ sẽ được đổi thành **"tên\_cũ\_deleted"**

### Hiển thị thông báo thành công hoặc thất bại

## 2.4 Sửa công nghệ theo ID

### Chức năng này cho phép cập nhật thông tin của một công nghệ dựa trên ID.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Sửa công nghệ" trong menu

### Hệ thống yêu cầu nhập ID công nghệ cần sửa

### Kiểm tra xem công nghệ có tồn tại không bằng cách gọi **getTechnologyById()**

### Hiển thị tên hiện tại của công nghệ

### Hệ thống yêu cầu nhập tên mới

### Kiểm tra tính hợp lệ của tên mới:

### Không được để trống

### Tên phải duy nhất (không trùng với công nghệ khác đang hoạt động)

### Nếu hợp lệ, hệ thống gọi phương thức **updateTechnology()** từ **TechnologyService**

### Dữ liệu được cập nhật vào database thông qua stored **procedure sp\_UpdateTechnology()**

### Hiển thị thông báo thành công hoặc thất bại

## 3. Quản lí ứng viên

### **3.1 Hiển thị các ứng viên**

### Chức năng này liệt kê tất cả các ứng viên trong hệ thống tuyển dụng.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Hiển thị danh sách ứng viên" trong menu

### Hệ thống gọi phương thức **getAllCandidates()** từ **CandidateService**

### Dữ liệu được truy xuất từ database thông qua stored **procedure sp\_GetAllCandidates()**

### Hiển thị danh sách các ứng viên theo định dạng bảng, bao gồm: ID, Tên, Email, SĐT, Kinh nghiệm (năm), Giới tính và Trạng thái

### Nếu không có ứng viên nào, hệ thống hiển thị thông báo "Không có ứng viên nào trong hệ thống"

### **3.2 Khóa/mở khóa tài khoản ứng viên theo id**

### Chức năng này cho phép thay đổi trạng thái hoạt động của tài khoản ứng viên. Cách hoạt động:

### Người dùng chọn tùy chọn "Khóa/Mở khóa tài khoản" trong menu

### Hệ thống yêu cầu nhập ID ứng viên

### Kiểm tra xem ứng viên có tồn tại không bằng cách gọi **getCandidateById()**

### Nếu tồn tại, hiển thị thông tin ứng viên và trạng thái hiện tại (active/inactive)

### Yêu cầu xác nhận thay đổi trạng thái từ người dùng

### Nếu người dùng xác nhận, hệ thống gọi phương thức **toggleCandidateStatus()** từ **CandidateService**

### Dữ liệu được cập nhật vào database thông qua stored **procedure sp\_ToggleCandidateStatus()**

### Hiển thị thông báo "Đã thay đổi trạng thái tài khoản thành công!" hoặc "Không thể thay đổi trạng thái tài khoản!"

### **3.3 Reset mật khẩu ứng viên theo id**

### Chức năng này cho phép reset mật khẩu cho ứng viên.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Reset mật khẩu ứng viên" trong menu

### Hệ thống yêu cầu nhập ID ứng viên

### Kiểm tra xem ứng viên có tồn tại không bằng cách gọi **getCandidateById()**

### Nếu tồn tại, hiển thị thông tin ứng viên

### Yêu cầu xác nhận reset mật khẩu từ người dùng

### Nếu người dùng xác nhận, hệ thống gọi phương thức **resetCandidatePassword()** từ **CandidateService**

### Service tạo mật khẩu ngẫu nhiên bằng phương thức **generateRandomPassword()** với định dạng **"Temp@" + các ký tự ngẫu nhiên**

### Dữ liệu được cập nhật vào database thông qua stored **procedure sp\_ResetCandidatePassword()**

### Hiển thị thông báo "Đã reset mật khẩu thành công!" và "Mật khẩu mới: [mật khẩu]" hoặc "Không thể reset mật khẩu!"

### **3.4 Tìm kiếm ứng viên theo tên**

### Chức năng này cho phép tìm kiếm ứng viên dựa trên tên.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Tìm kiếm theo tên" trong menu

### Hệ thống yêu cầu nhập từ khóa tìm kiếm

### Hệ thống gọi phương thức **searchCandidatesByName()** từ **CandidateServic**e

### Dữ liệu được truy xuất từ database thông qua stored **procedure sp\_SearchCandidatesByName()**

### Hệ thống tìm tất cả ứng viên có tên chứa từ khóa đã nhập (không phân biệt hoa thường)

### Hiển thị kết quả tìm kiếm theo định dạng bảng

### Nếu không tìm thấy ứng viên nào, hiển thị thông báo "Không tìm thấy ứng viên nào phù hợp với từ khóa: [từ khóa]"

### **3.5 Lọc theo tiêu chí**

### Chức năng này cho phép lọc ứng viên theo nhiều tiêu chí khác nhau.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Lọc theo tiêu chí" trong menu

### Hệ thống hiển thị các tiêu chí lọc:

### Kinh nghiệm

### Tuổi

### Giới tính

### Công nghệ

### Người dùng chọn một tiêu chí lọc

### Tùy theo tiêu chí đã chọn:

### Kinh nghiệm: Yêu cầu nhập kinh nghiệm tối thiểu và tối đa (năm), gọi **filterCandidatesByEx**perience() và stored **procedure sp\_FilterCandidatesByExperience()**

### Tuổi: Yêu cầu nhập tuổi tối thiểu và tối đa, gọi **filterCandidatesByAge()** và **stored procedure sp\_FilterCandidatesByAge()**

### Giới tính: Yêu cầu nhập giới tính (Nam/Nữ), gọi **filterCandidatesByGender()** và stored **procedure sp\_FilterCandidatesByGender()**

### Công nghệ: Yêu cầu nhập ID công nghệ, gọi **u**

## 4. Quản lí vị trí tuyển dụng

## 4.1 Thêm vị trí tuyển dụng

**Mô tả:** Chức năng này cho phép thêm một vị trí tuyển dụng mới vào hệ thống cùng với các công nghệ yêu cầu.

**Cách hoạt động:**

* Người dùng chọn tùy chọn "Thêm vị trí tuyển dụng mới" trong menu
* Hệ thống yêu cầu nhập các thông tin của vị trí mới:
  + Tên vị trí (bắt buộc, tối đa 100 ký tự, không chứa "\*deleted")
  + Mô tả (tối đa 500 ký tự)
  + Lương tối thiểu (không âm)
  + Lương tối đa (không âm, phải lớn hơn hoặc bằng lương tối thiểu)
  + Số năm kinh nghiệm tối thiểu (không âm)
  + Ngày hết hạn (phải là ngày trong tương lai)
* Hệ thống tự động thiết lập ngày tạo là ngày hiện tại
* Các dữ liệu đầu vào được xác thực bởi lớp **ValidatePosition**
* Nếu hợp lệ, hệ thống gọi phương thức **addPosition()** từ **PositionService**
* Dữ liệu được lưu vào database thông qua stored procedure **sp\_addPosition()**
* Sau khi lưu vị trí, hệ thống hiển thị danh sách các công nghệ có sẵn
* Người dùng chọn các công nghệ yêu cầu cho vị trí bằng cách nhập các số thứ tự cách nhau bởi dấu phẩy
* Hệ thống gọi phương thức **addPositionTechnology()** để lưu mối quan hệ giữa vị trí và công nghệ
* Hiển thị thông báo thành công hoặc thất bại

## 4.2 Sửa vị trí tuyển dụng theo ID

**Mô tả:** Chức năng này cho phép cập nhật thông tin của một vị trí tuyển dụng dựa trên ID.

**Cách hoạt động:**

* Người dùng chọn tùy chọn "Cập nhật vị trí tuyển dụng" trong menu
* Hệ thống hiển thị danh sách các vị trí đang hoạt động
* Người dùng nhập ID vị trí cần cập nhật
* Hệ thống kiểm tra xem vị trí có tồn tại không bằng cách gọi **getPositionById()**
* Hệ thống hiển thị thông tin hiện tại của vị trí và menu các trường có thể cập nhật:
  1. Tên vị trí
  2. Mô tả
  3. Lương tối thiểu
  4. Lương tối đa
  5. Kinh nghiệm tối thiểu
  6. Ngày hết hạn
  7. Công nghệ yêu cầu
  8. Lưu thay đổi
  9. Hủy thay đổi
* Người dùng chọn trường cần cập nhật và nhập giá trị mới
* Mỗi giá trị được xác thực bởi các phương thức tương ứng trong lớp **ValidatePosition**
* Khi cập nhật công nghệ, người dùng có thể xóa tất cả công nghệ hiện tại và chọn lại
* Khi chọn lưu thay đổi, hệ thống gọi phương thức **updatePosition()** từ **PositionService**
* Dữ liệu được cập nhật vào database thông qua stored procedure **sp\_updatePosition()**
* Hiển thị thông báo thành công hoặc thất bại

## 4.3 Xóa vị trí tuyển dụng theo ID

**Mô tả:** Chức năng này cho phép xóa một vị trí tuyển dụng khỏi hệ thống dựa trên ID.

**Cách hoạt động:**

* Người dùng chọn tùy chọn "Xóa vị trí" trong menu
* Hệ thống hiển thị danh sách các vị trí đang hoạt động
* Người dùng nhập ID vị trí cần xóa
* Hệ thống kiểm tra xem vị trí có tồn tại không bằng cách gọi **getPositionById()**
* Hiển thị thông tin vị trí cần xóa và yêu cầu xác nhận từ người dùng
* Nếu người dùng xác nhận, hệ thống gọi phương thức **deletePosition()** từ **PositionService**
* Xử lý xóa vị trí có 2 trường hợp:
  + Xóa cứng: Nếu vị trí không liên kết với bảng khác (thông qua **sp\_checkPositionDependencies()**)
  + Xóa mềm: Nếu vị trí đã có liên kết, vị trí sẽ được đánh dấu là đã xóa (thông qua **sp\_markPositionAsDeleted()**)
* Hiển thị thông báo thành công hoặc thất bại

## 4.4 Xem danh sách vị trí

**Mô tả:** Chức năng này liệt kê tất cả các vị trí tuyển dụng đang hoạt động trong hệ thống.

**Cách hoạt động:**

* Người dùng chọn tùy chọn "Xem danh sách vị trí đang hoạt động" trong menu
* Hệ thống gọi phương thức **getAllActivePositions()** từ **PositionService**
* Dữ liệu được truy xuất từ database thông qua stored procedure sp\_getAllPositions()
* Hệ thống lọc ra các vị trí đang hoạt động (không bị đánh dấu xóa)
* Hiển thị danh sách các vị trí theo định dạng bảng với các thông tin:
  + ID
  + Tên vị trí
  + Lương tối thiểu
  + Lương tối đa
  + Kinh nghiệm tối thiểu
  + Ngày hết hạn

## 4.5 Xem chi tiết vị trí

**Mô tả:** Chức năng này cho phép xem thông tin chi tiết của một vị trí tuyển dụng cụ thể, bao gồm cả các công nghệ yêu cầu.

**Cách hoạt động:**

* Người dùng chọn tùy chọn "Xem chi tiết vị trí" trong menu
* Hệ thống hiển thị danh sách các vị trí đang hoạt động
* Người dùng nhập ID vị trí cần xem chi tiết
* Hệ thống kiểm tra xem vị trí có tồn tại không bằng cách gọi **getPositionById()**
* Hệ thống hiển thị thông tin chi tiết của vị trí:
  + ID
  + Tên vị trí
  + Mô tả
  + Lương tối thiểu
  + Lương tối đa
  + Kinh nghiệm tối thiểu
  + Ngày tạo
  + Ngày hết hạn
* Hệ thống gọi phương thức **getPositionTechnologies()** để lấy danh sách công nghệ yêu cầu
* Hiển thị danh sách các công nghệ yêu cầu cho vị trí

## 5. Quản lí đơn ứng tuyển

## 5.1 Hiển thị các đơn ứng tuyển

### **Chức năng này** liệt kê tất cả các đơn ứng tuyển đang được quản lý trong hệ thống tuyển dụng (không bao gồm đơn đã hủy).

### **Cách hoạt động:**

### Người dùng chọn tùy chọn "Xem danh sách đơn ứng tuyển" trong menu

### Hệ thống gọi phương thức **getAllApplications()** từ **ApplicationFormService**

### Dữ liệu được truy xuất từ database thông qua stored procedure **sp\_GetAllActiveApplications()**

### Các đơn ứng tuyển được hiển thị dưới dạng bảng với các thông tin:

### ID đơn

### Tên ứng viên

### Vị trí ứng tuyển

### Trạng thái đơn (**pending**, **handling**, **interviewing**, **done**)

## 5.2 Lọc đơn theo trạng thái

### **Chức năng này** cho phép người dùng lọc danh sách đơn ứng tuyển theo trạng thái xử lý.

### **Cách hoạt động:**

### Người dùng chọn tùy chọn "Lọc đơn theo trạng thái" trong menu

### Hệ thống hiển thị danh sách các trạng thái có thể lọc:

### Pending - Chờ xử lý

### Handling - Đang xử lý

### Interviewing - Đang phỏng vấn

### Done - Hoàn thành

### Người dùng chọn một trạng thái

### Hệ thống gọi phương thức **filterByProgress()** từ **ApplicationFormService**

### Dữ liệu được truy xuất từ database thông qua stored procedure **sp\_FilterApplicationsByProgress()**

### Hiển thị danh sách đơn ứng tuyển theo trạng thái được chọn

## 5.3 Lọc đơn theo kết quả

### **Chức năng này** cho phép người dùng lọc danh sách đơn ứng tuyển theo kết quả phỏng vấn.

### **Cách hoạt động:**

### Người dùng chọn tùy chọn "Lọc đơn theo kết quả" trong menu

### Hệ thống hiển thị danh sách các kết quả có thể lọc:

### Pass - Đậu

### Fail - Trượt

### Người dùng chọn một kết quả

### Hệ thống gọi phương thức **filterByResult()** từ **ApplicationFormService**

### Dữ liệu được truy xuất từ database thông qua stored procedure **sp\_FilterApplicationsByResult()**

### Hiển thị danh sách đơn ứng tuyển theo kết quả được chọn

## 5.4 Hủy đơn ứng tuyển

### **Chức năng này** cho phép người dùng hủy một đơn ứng tuyển dựa trên ID.

### **Cách hoạt động:**

### Người dùng chọn tùy chọn "Hủy đơn" trong menu

### Hệ thống yêu cầu nhập ID đơn cần hủy

### Kiểm tra xem đơn có tồn tại không bằng cách gọi **viewApplicationDetail()**

### Hệ thống yêu cầu người dùng nhập lý do hủy đơn

### Kiểm tra tính hợp lệ của lý do:

### Không được để trống

### Nếu hợp lệ, hệ thống gọi phương thức **cancelApplication()** từ **ApplicationFormService**

### Dữ liệu được cập nhật vào database thông qua stored procedure **sp\_CancelApplication()**

### Hiển thị thông báo thành công hoặc thất bại

## 5.5 Xem chi tiết đơn ứng tuyển

### **Chức năng này** cho phép người dùng xem thông tin chi tiết của một đơn ứng tuyển dựa trên ID.

### **Cách hoạt động:**

### Người dùng chọn tùy chọn "Xem chi tiết đơn" trong menu

### Hệ thống yêu cầu nhập ID đơn cần xem

### Hệ thống gọi phương thức **viewApplicationDetail()** từ **ApplicationFormService**

### Dữ liệu được truy xuất từ database thông qua stored procedure **sp\_ViewApplicationDetail()**

### Khi xem chi tiết, trạng thái đơn sẽ tự động chuyển từ "pending" sang "handling" (được xử lý bởi stored procedure)

### Hiển thị chi tiết đơn ứng tuyển bao gồm:

### Thông tin cơ bản đơn (ID, trạng thái)

### Thông tin ứng viên (tên, email, điện thoại, kinh nghiệm, CV)

### Thông tin vị trí ứng tuyển (tên vị trí, mô tả, lương, yêu cầu kinh nghiệm)

### Thông tin phỏng vấn (thời gian, link, kết quả, ghi chú)

## 5.6 Gửi thông tin phỏng vấn

### **Chức năng này** cho phép người dùng gửi thông tin phỏng vấn cho một đơn ứng tuyển.

### **Cách hoạt động:**

### Người dùng chọn tùy chọn "Gửi thông tin phỏng vấn" trong menu

### Hệ thống yêu cầu nhập ID đơn

### Kiểm tra xem đơn có tồn tại không bằng cách gọi **viewApplicationDetail()**

### Kiểm tra trạng thái hiện tại của đơn:

### Không cho phép gửi thông tin nếu đơn đã ở trạng thái "interviewing" hoặc "done"

### Hệ thống yêu cầu nhập:

### Link phỏng vấn (không được để trống)

### Thời gian phỏng vấn (định dạng: dd/MM/yyyy HH )

### Nếu thông tin hợp lệ, hệ thống gọi phương thức **sendInterviewInfo()** từ **ApplicationFormService**

### Dữ liệu được cập nhật vào database thông qua stored procedure **sp\_SendInterviewInfo()**

### Stored procedure cũng sẽ tự động chuyển trạng thái đơn từ "handling" sang "interviewing"

### Hiển thị thông báo thành công hoặc thất bại

## 5.7 Cập nhật kết quả phỏng vấn

### **Chức năng này** cho phép người dùng cập nhật kết quả phỏng vấn cho một đơn ứng tuyển.

### **Cách hoạt động:**

### Người dùng chọn tùy chọn "Cập nhật kết quả phỏng vấn" trong menu

### Hệ thống yêu cầu nhập ID đơn

### Kiểm tra xem đơn có tồn tại không bằng cách gọi **viewApplicationDetail()**

### Kiểm tra trạng thái hiện tại của đơn:

### Chỉ cho phép cập nhật kết quả nếu đơn đang ở trạng thái "interviewing"

### Hệ thống yêu cầu chọn kết quả phỏng vấn:

### Pass - Đậu

### Fail - Trượt

### Hệ thống yêu cầu nhập ghi chú kết quả phỏng vấn

### Hệ thống gọi phương thức **updateInterviewResult()** từ **ApplicationFormService**

### Dữ liệu được cập nhật vào database thông qua stored procedure **sp\_UpdateInterviewResult()**

### Stored procedure cũng sẽ tự động chuyển trạng thái đơn từ "interviewing" sang "done"

### Hiển thị thông báo thành công hoặc thất bại

## 6. Quản lí thông tin cá nhân

### **6.1 Cập nhật thông tin cá nhân**

**Mô tả:** Cho phép ứng viên cập nhật thông tin cá nhân của mình.

**Cách hoạt động:**

* Ứng viên chọn mục "Cập nhật thông tin cá nhân" trong menu
* Hệ thống hiển thị thông tin hiện tại của ứng viên
* Ứng viên có thể chọn cập nhật từng thông tin riêng biệt hoặc cập nhật tất cả thông tin
* Các thông tin có thể cập nhật bao gồm:
  + Họ tên
  + Số điện thoại (có kiểm tra định dạng)
  + Số năm kinh nghiệm
  + Giới tính (chỉ chấp nhận "Nam" hoặc "Nữ")
  + Mô tả bản thân
  + Ngày sinh (định dạng YYYY-MM-DD)
* Dữ liệu được cập nhật thông qua phương thức **updateCandidateInfo()** của **CandidateService**
* Hiển thị thông báo cập nhật thành công hoặc thất bại và thông tin sau khi cập nhật

### **6.2 Thay đổi mật khẩu**

**Mô tả:** Cho phép ứng viên thay đổi mật khẩu đăng nhập.

**Cách hoạt động:**

* Ứng viên chọn mục "Đổi mật khẩu" trong menu
* Ứng viên nhập định danh (email hoặc số điện thoại) để xác thực (chỉ được đổi mật khẩu cho tài khoản của chính mình)
* Hệ thống kiểm tra định danh có hợp lệ và tồn tại trong hệ thống không
* Ứng viên nhập mật khẩu cũ (tối đa 3 lần thử)
* Hệ thống xác thực mật khẩu cũ thông qua phương thức **verifyPassword()** của **CandidateService**
* Nếu đúng, ứng viên nhập mật khẩu mới và xác nhận mật khẩu mới
* Hệ thống kiểm tra tính hợp lệ của mật khẩu mới (ít nhất 6 ký tự, bao gồm chữ hoa, chữ thường và số)
* Dữ liệu được cập nhật thông qua phương thức **changePassword()** của **CandidateService**
* Hiển thị thông báo đổi mật khẩu thành công hoặc thất bại

### **6.3 Quản lý công nghệ**

**Mô tả:** Cho phép ứng viên quản lý các công nghệ mà mình có kỹ năng.

**Cách hoạt động:**

* Ứng viên chọn mục "Quản lý công nghệ" trong menu
* Hệ thống hiển thị menu con với các chức năng:
  + Xem danh sách công nghệ đã đăng ký
  + Thêm công nghệ mới
  + Xóa công nghệ

#### 6.3.1 Xem danh sách công nghệ đã đăng ký

**Mô tả:** Hiển thị danh sách các công nghệ mà ứng viên đã đăng ký.

**Cách hoạt động:**

* Hệ thống gọi phương thức **getCandidateTechnologies()** từ **CandidateTechnologyService**
* Hiển thị danh sách công nghệ bao gồm ID và tên công nghệ
* Nếu chưa có công nghệ nào được đăng ký, hiển thị thông báo tương ứng

#### 6.3.2 Thêm công nghệ mới

**Mô tả:** Cho phép ứng viên thêm công nghệ mới vào hồ sơ của mình.

**Cách hoạt động:**

* Hệ thống lấy danh sách tất cả công nghệ thông qua phương thức **getAllTechnologies()** từ **TechnologyService**
* Hệ thống lấy danh sách công nghệ đã đăng ký của ứng viên thông qua phương thức **getCandidateTechnologies()**
* Hiển thị danh sách tất cả công nghệ và đánh dấu những công nghệ đã đăng ký
* Ứng viên chọn ID công nghệ muốn thêm
* Hệ thống kiểm tra:
  + Công nghệ có tồn tại không
  + Ứng viên đã đăng ký công nghệ này chưa
* Nếu hợp lệ, hệ thống gọi phương thức **addTechnologyToCandidate()** từ **CandidateTechnologyService**
* Hiển thị thông báo thêm công nghệ thành công hoặc thất bại

#### 6.3.3 Xóa công nghệ

**Mô tả:** Cho phép ứng viên xóa công nghệ khỏi hồ sơ của mình.

**Cách hoạt động:**

* Hệ thống hiển thị danh sách công nghệ đã đăng ký của ứng viên
* Ứng viên chọn ID công nghệ muốn xóa
* Hệ thống kiểm tra công nghệ có trong danh sách đã đăng ký không
* Hệ thống yêu cầu xác nhận xóa từ ứng viên
* Nếu ứng viên xác nhận, hệ thống gọi phương thức **removeTechnologyFromCandidate()** từ **CandidateTechnologyService**
* Hiển thị thông báo xóa công nghệ thành công hoặc thất bại

## 7. Quản lí đơn ứng tuyển của bản thân (ứng viên)

### **7.1 Xem danh sách đơn đã nộp**

### Chức năng này liệt kê tất cả các đơn ứng tuyển mà ứng viên đã nộp.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Xem danh sách đơn đã nộp" trong menu

### Hệ thống gọi phương thức **getApplicationsByCandidateId()** từ **ApplicationService**

### Dữ liệu được truy xuất từ database thông qua stored **procedure sp\_GetApplicationsByCandidateId()**

### Hệ thống hiển thị danh sách các đơn đã nộp theo định dạng: ID, Vị trí, Tiến độ, và Ngày nộp

### Trường hợp chưa có đơn ứng tuyển nào, hiển thị thông báo "Bạn chưa nộp đơn ứng tuyển nào!"

### Vị trí tuyển dụng được lấy từ **getPositionById()** và được cắt ngắn nếu tên quá dài bằng phương thức **truncateString()**

### **1.2 Xem chi tiết đơn**

### Chức năng này cho phép xem thông tin chi tiết của một đơn ứng tuyển cụ thể dựa trên ID.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Xem chi tiết đơn" trong menu

### Hệ thống yêu cầu nhập ID đơn muốn xem chi tiết

### Kiểm tra xem đơn có tồn tại và thuộc về ứng viên hiện tại không bằng cách lọc từ danh sách **getApplicationsByCandidateId()**

### Nếu tìm thấy đơn, hiển thị thông tin chi tiết bao gồm:

### ID đơn

### Vị trí tuyển dụng (lấy từ **getPositionById()**)

### Tiến độ xử lý đơn

### URL của CV đã nộp

### Ngày tạo đơn

### Nếu đơn đang ở trạng thái "interviewing" và có ngày yêu cầu phỏng vấn, hệ thống sẽ hiển thị thêm:

### Ngày yêu cầu phỏng vấn

### Thời gian phỏng vấn (nếu có)

### Link phỏng vấn (nếu có)

### Phản hồi phỏng vấn (nếu đã có)

### Nếu ứng viên chưa phản hồi yêu cầu phỏng vấn, hệ thống cho phép:

### Xác nhận tham gia phỏng vấn: gọi **updateInterviewResponse()** với trạng thái "Đã xác nhận"

### Từ chối phỏng vấn: gọi **updateInterviewResponse()** với trạng thái "Từ chối" và lý do từ chối

### Nếu đơn đang ở trạng thái "done", hiển thị thêm:

### Kết quả tuyển dụng

### Ghi chú từ nhà tuyển dụng (nếu có)

## 8. Xem và nộp đơn.

### **7.1 Xem danh sách vị trí đang hoạt động**

### Chức năng này liệt kê tất cả các vị trí tuyển dụng đang còn hoạt động trong hệ thống.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Xem danh sách vị trí đang hoạt động" trong menu

### Hệ thống gọi phương thức **getAllActivePositions()** từ **ApplicationService**

### Dữ liệu được truy xuất từ database thông qua stored **procedure sp\_GetAllActivePositions()**

### Hệ thống hiển thị danh sách các vị trí tuyển dụng đang hoạt động theo định dạng bảng gồm các cột:

### ID vị trí

### Tên vị trí (được cắt ngắn nếu quá dài bằng **truncateString()**)

### Phạm vi lương (minSalary - maxSalary)

### Kinh nghiệm tối thiểu (tính theo năm)

### Hạn nộp đơn (expiredDate)

### Trường hợp không có vị trí nào đang hoạt động, hiển thị thông báo "Không có vị trí nào đang hoạt động!"

### **8.2 Xem chi tiết và apply**

### Chức năng này cho phép xem thông tin chi tiết của một vị trí tuyển dụng và nộp đơn ứng tuyển.

### Cách hoạt động:

### Người dùng chọn tùy chọn "Xem chi tiết và apply" trong menu

### Hệ thống yêu cầu nhập ID vị trí muốn xem chi tiết

### Kiểm tra xem vị trí có tồn tại không bằng cách gọi **getPositionById()** từ **ApplicationService**

### Nếu tìm thấy vị trí, hiển thị thông tin chi tiết bằng phương thức **formatPositionDetails()**, bao gồm:

### ID vị trí

### Tên vị trí

### Mô tả vị trí

### Phạm vi lương

### Kinh nghiệm tối thiểu

### Ngày tạo

### Ngày hết hạn

### Hệ thống hỏi người dùng có muốn nộp đơn vào vị trí này không

### Nếu người dùng chọn "Y" (đồng ý nộp đơn):

### Yêu cầu nhập URL CV

### Kiểm tra URL CV không được để trống

### Kiểm tra xem ứng viên đã nộp đơn vào vị trí này chưa bằng cách:

### Lấy danh sách đơn ứng tuyển của ứng viên qua **getApplicationsByCandidateId()**

### Kiểm tra xem có đơn nào có cùng recruitmentPositionId và chưa bị xóa (destroyAt == null) không

### Nếu ứng viên chưa nộp đơn vào vị trí này, gọi phương thức **applyForPosition()** với các tham số:

### ID của ứng viên hiện tại

### ID của vị trí tuyển dụng

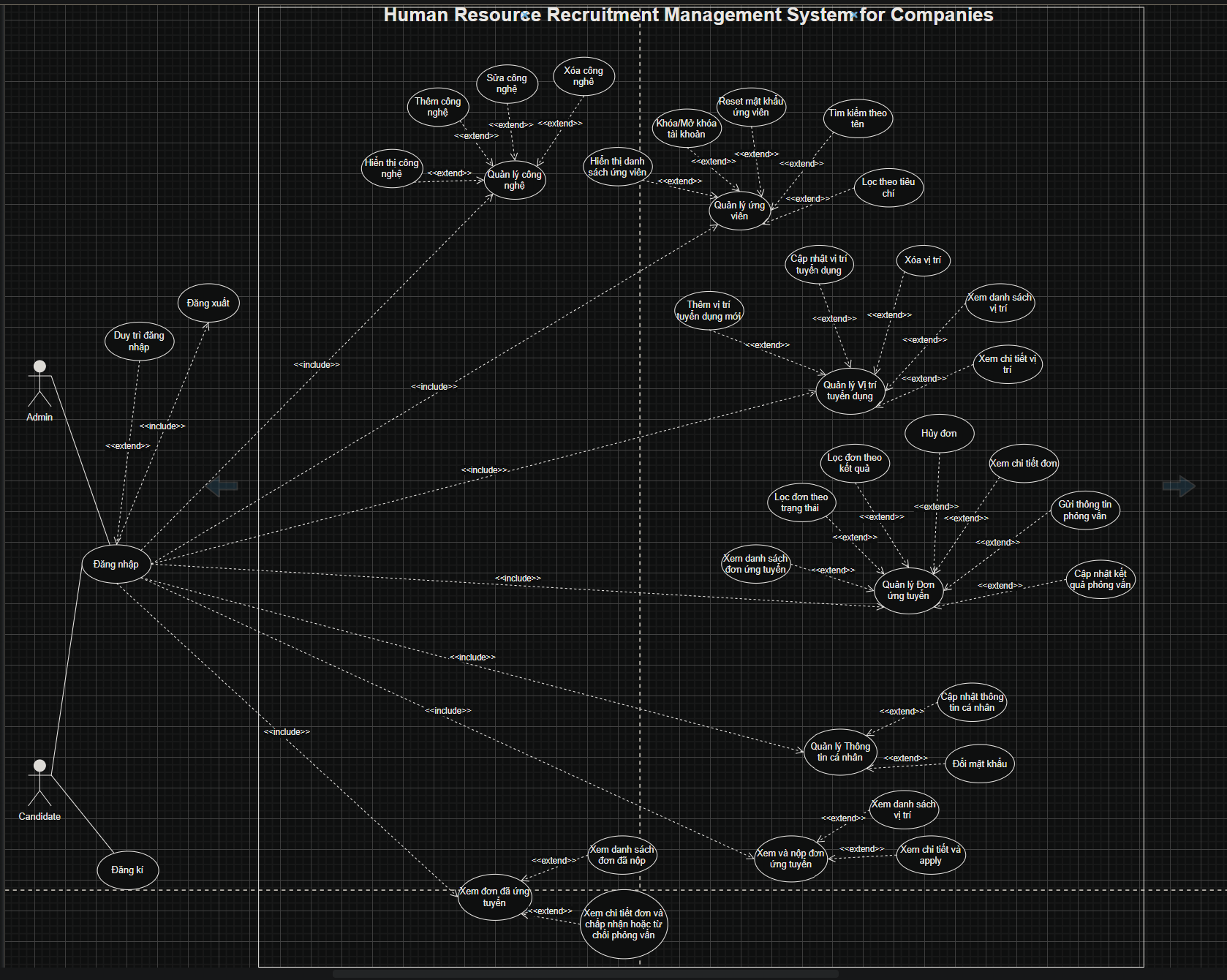
### URL CV đã nhập

### Nếu nộp đơn thành công, hiển thị thông báo thành công và gợi ý theo dõi đơn trong mục "Quản lý đơn ứng tuyển của tôi"

### Nếu đã nộp đơn trước đó, hiển thị thông báo "Bạn đã nộp đơn vào vị trí này rồi!"

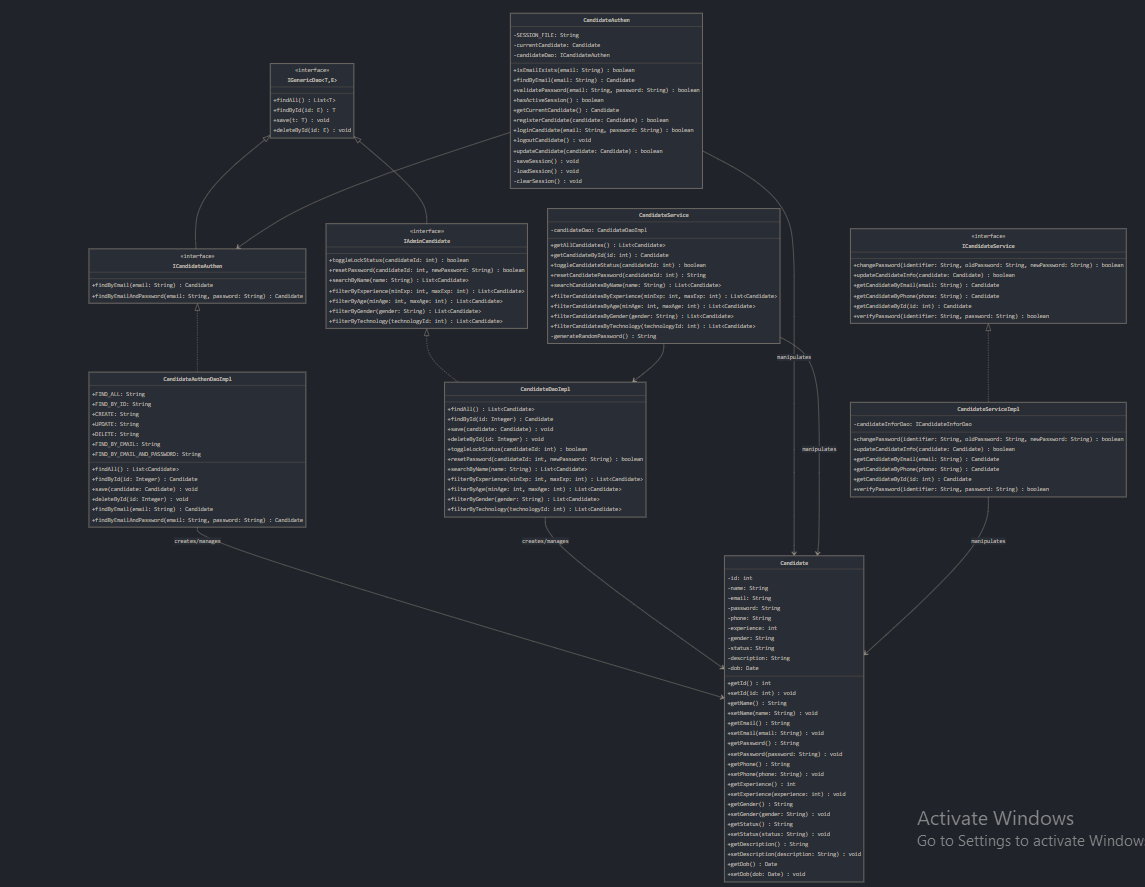
### Nếu có lỗi xảy ra, hiển thị thông báo lỗi

# Use Case Diagram

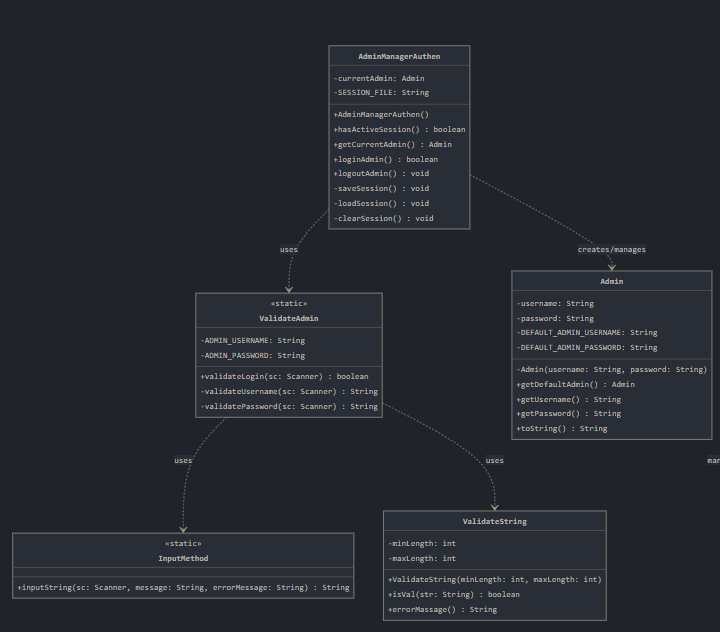


# Class Diagram

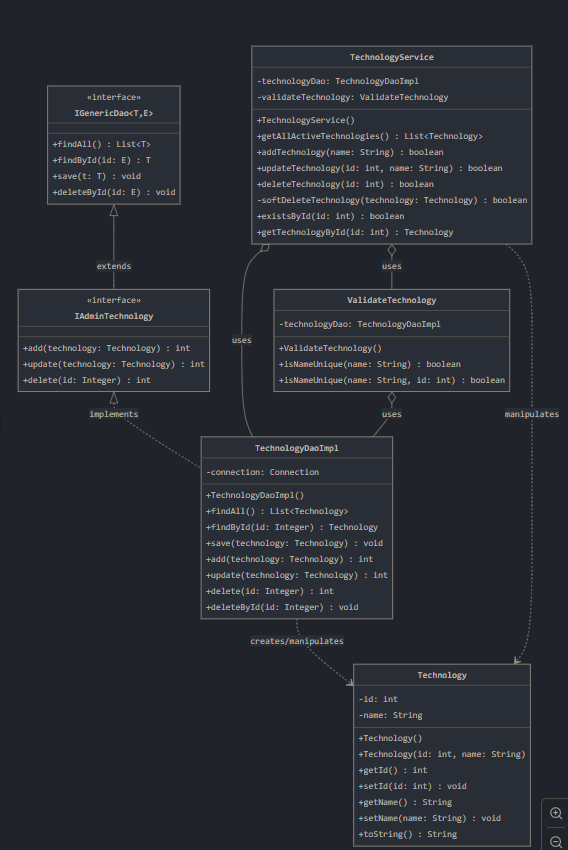
Candidate



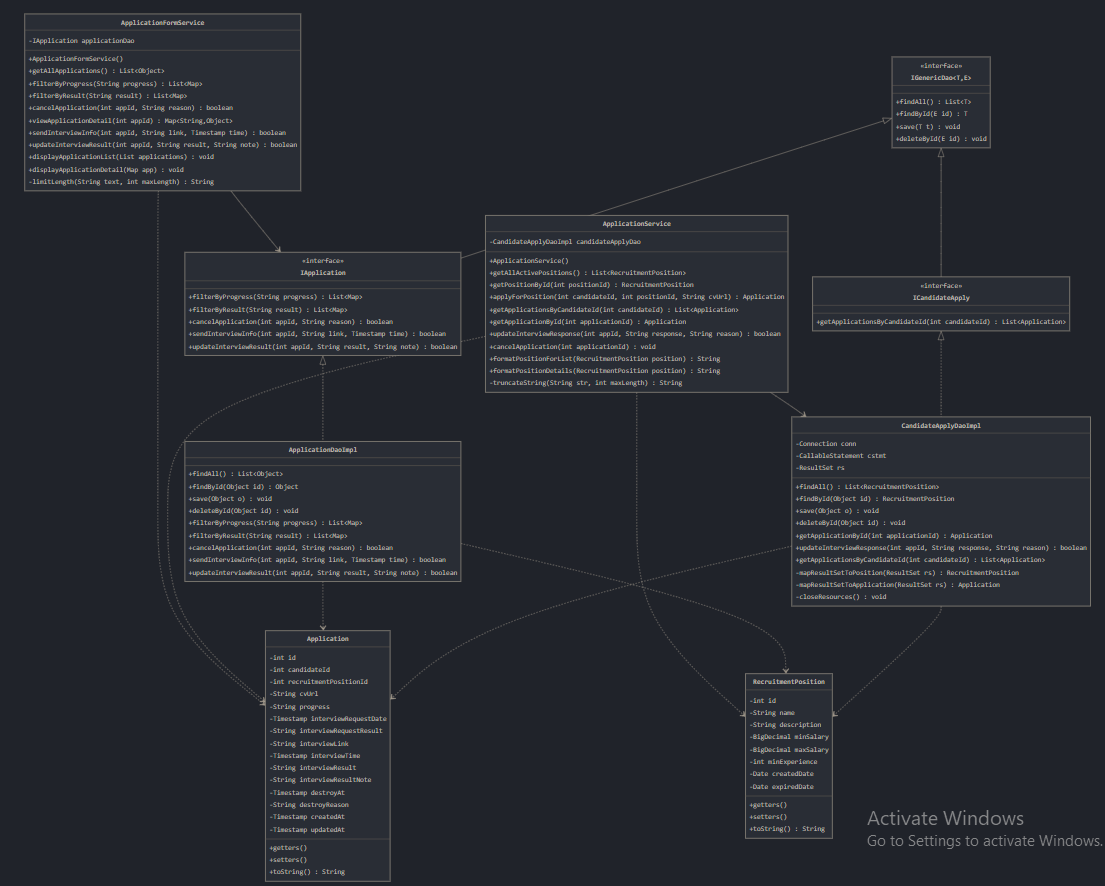
AdminAuthen

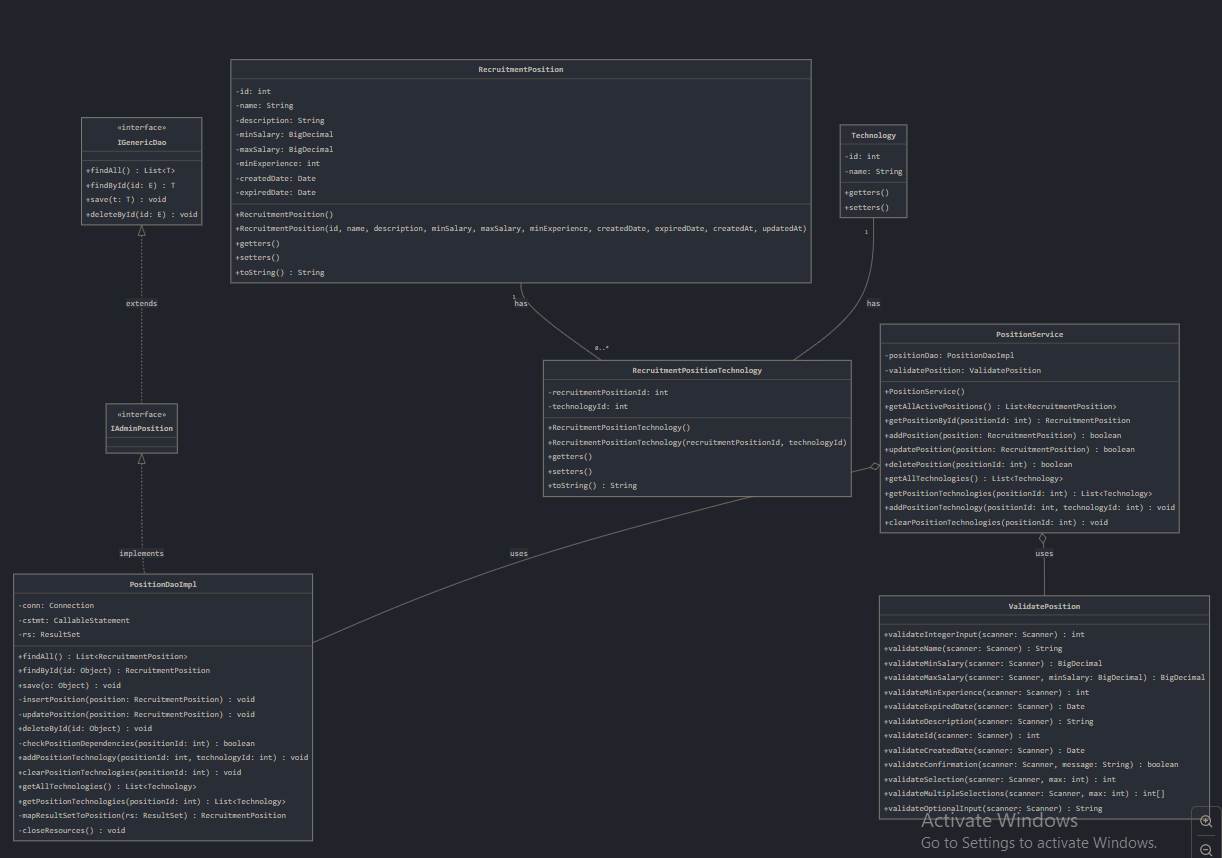


Technology

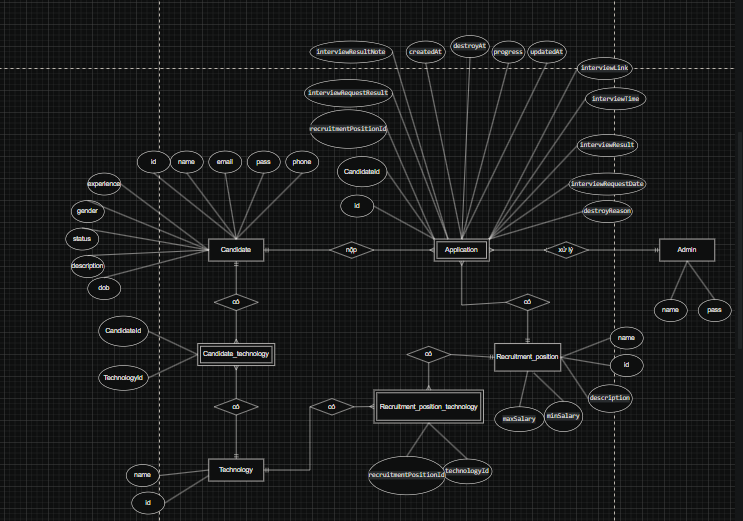


Model ApplicationAndRecruitnmentposition

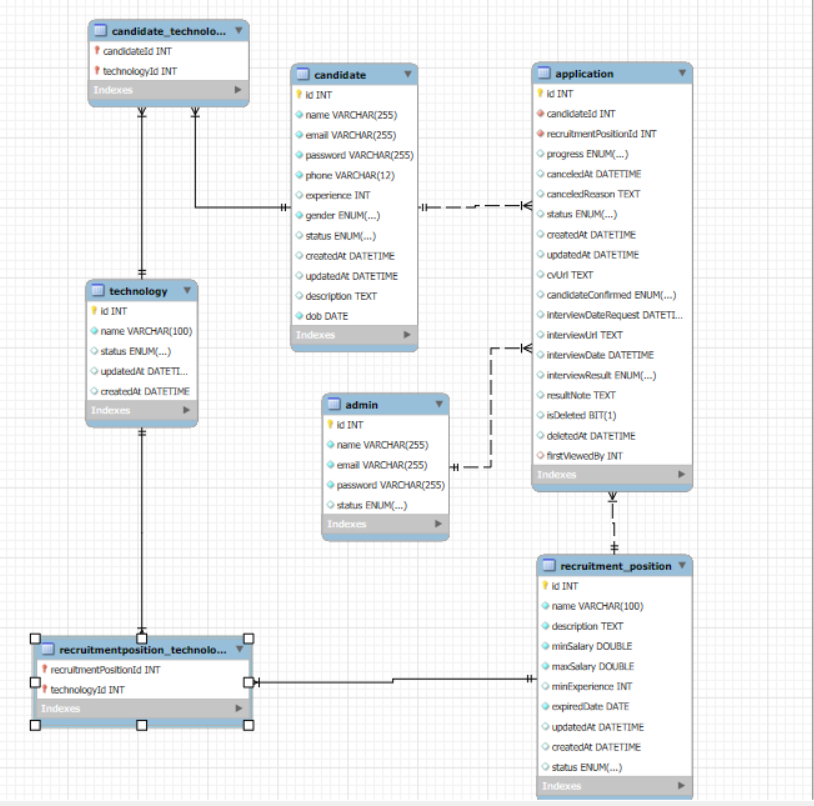




# Entity Relationship Diagram



# Enhanced Entity Relationship - EER



# Cấu trúc thư mục

