**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**2251052069 - Đặng Phương Nam**

**2251052060 - Nguyễn Tấn Lộc**

**ỨNG DỤNG TÌM KIẾM VÀ**

**QUẢN LÝ VIỆC LÀM TRỰC TUYẾN**

**TP. HỒ CHÍ MINH, 2024 - 2025**

**MỤC LỤC**

[DANH MỤC TỪ VIẾT TẮT 5](#__RefHeading___Toc2288_1634557080)

[DANH MỤC HÌNH VẼ 6](#__RefHeading___Toc2290_1634557080)

[DANH MỤC BẢNG 7](#__RefHeading___Toc2292_1634557080)

[Chương 1. GIỚI THIỆU ĐỀ TÀI 8](#__RefHeading___Toc2294_1634557080)

[1.1. Giới thiệu 8](#__RefHeading___Toc2296_1634557080)

[1.2. Phân tích yêu cầu 8](#__RefHeading___Toc2298_1634557080)

[1.2.1. Xác định các chức năng 8](#__RefHeading___Toc2300_1634557080)

[1.2.2. Lược đồ use case 8](#__RefHeading___Toc2302_1634557080)

[1.2.3. Đặc tả use case 8](#__RefHeading___Toc2304_1634557080)

[1.3. Thiết kế hệ thống 8](#__RefHeading___Toc2306_1634557080)

[1.3.1. Sơ đồ lớp UML 8](#__RefHeading___Toc2308_1634557080)

[1.3.2. Lược đồ cơ sở dữ liệu quan hệ 8](#__RefHeading___Toc2310_1634557080)

[1.3.3. Thiết kế giao diện và thiết kế xử lý 8](#__RefHeading___Toc2312_1634557080)

[Chương 2. HIỆN THỰC RESTFUL API 9](#__RefHeading___Toc2314_1634557080)

[2.1. Thiết kế Model 9](#__RefHeading___Toc2316_1634557080)

[2.2. Xây dựng API 9](#__RefHeading___Toc2318_1634557080)

[2.3. Kiểm thử API 9](#__RefHeading___Toc2320_1634557080)

[Chương 3. PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG REACT NATIVE 10](#__RefHeading___Toc2322_1634557080)

[3.1. Cấu trúc ứng dụng 10](#__RefHeading___Toc2324_1634557080)

[3.2. Xây dựng giao diện 10](#__RefHeading___Toc2326_1634557080)

[3.3. Tích hợp API 10](#__RefHeading___Toc2328_1634557080)

[3.4. Tối ưu hiệu suất 10](#__RefHeading___Toc2330_1634557080)

[Chương 4. HỆ THỐNG <TÊN ĐỀ TÀI> 11](#__RefHeading___Toc2332_1634557080)

[4.1. Kết quả đạt được của đề tài 11](#__RefHeading___Toc2334_1634557080)

[4.2. Các chức năng hệ thống 11](#__RefHeading___Toc2336_1634557080)

[4.2.1. Chức năng 1 11](#__RefHeading___Toc2338_1634557080)

[4.2.2. Chức năng 2 11](#__RefHeading___Toc2340_1634557080)

DANH MỤC TỪ VIẾT TẮT

DANH MỤC HÌNH VẼ

[Hình 1.1 : Tên hình 1 6](#_Toc1588)

[Hình 3.1 : Hình chức năng 1 (chụp từ kết quả hệ thống) 9](#_Toc11872)

[Hình 3.2 : Hình chức năng 2 (chụp từ kết quả hệ thống) 9](#_Toc6349)

DANH MỤC BẢNG

[Bảng 1.1 : Tên bảng 1 6](#_Toc27355)

# GIỚI THIỆU ĐỀ TÀI

## Giới thiệu

Sinh viên trình bày vắn tắt nội dung và mục tiêu đề tài (viết đoạn ít nhất 5 dòng)

## Phân tích yêu cầu

### Xác định các chức năng

|  |  |  |
| --- | --- | --- |
| **TT** | **Chức năng** | **Nội dung** |
| 1 | **Xác thực và phân quyền** | **- Đăng nhập, đăng ký với vai trò người tìm việc, nhà tuyển dụng và quản trị viên.**  **- Nhà tuyển dụng cần được quản trị viên xác thực trước khi đăng tin tuyển dụng.** |
| 2 | **Hồ sơ và CV** | **- Người tìm việc tạo hồ sơ cá nhân, tải lên CV hoặc tự thiết kế CV trực tuyến.**  **- Cho phép lưu nhiều phiên bản CV để nộp cho các vị trí khác nhau.** |
| 3 | **Tìm kiếm và lọc công việc** | **- Tìm theo từ khóa, công ty, vị trí, mức lương, loại hình công việc.**  **- Sử dụng AI để gợi ý công việc phù hợp dựa trên hồ sơ người dùng.**  **- Sắp xếp kết quả theo ngày đăng, mức lương, hoặc độ phổ biến** |
| 4 | **Quản lý quy trình ứng tuyển** | **- Ứng tuyển trực tuyến nhanh chóng với một cú nhấp chuột.**  **- Theo dõi trạng thái hồ sơ: đã gửi, đang phỏng vấn, trúng tuyển.** |
| 5 | **Chat và phỏng vấn trực tuyến** | - \*Tích hợp chat thời gian thực bằng Firebase để kết nối ứng viên và nhà tuyển dụng.  - \*Hỗ trợ phỏng vấn trực tuyến qua video call. |
| 6 | **Thông báo và nhắc nhở** | - \*Gửi thông báo về việc làm mới hoặc trạng thái hồ sơ qua email/push notification.  - Nhắc nhở lịch phỏng vấn hoặc hết hạn ứng tuyển |
| 7 | **Thống kê và báo cáo** | - Nhà tuyển dụng xem số lượng hồ sơ đã nhận, tỉ lệ ứng viên đạt yêu cầu.  - Người tìm việc xem số lượt xem hồ sơ, tỉ lệ phản hồi từ nhà tuyển dụng. |

### Lược đồ use case

|  |
| --- |
|  |

Hình 1.1: Lược đồ use case hệ thống

### Đặc tả use case

## Thiết kế hệ thống

### Sơ đồ lớp UML

|  |
| --- |
|  |

Phân tích, giải thích các mối quan hệ thiết lập

### Lược đồ cơ sở dữ liệu quan hệ

Lược đồ cơ sở dữ liệu quan hệ.

|  |
| --- |
| sssss |

Thông tin các bảng.

Bảng 1.2: Tên bảng 2

### Thiết kế giao diện và thiết kế xử lý

Thiết kế cac giao diện chính và các thiết kế xử lý tương ứng từng màn hình

# **HIỆN THỰC RESTFUL API**

## Thiết kế Model

|  |
| --- |
| from django.db import models from django.contrib.auth.models import AbstractUser from django.conf import settings from django.utils.text import slugify import uuid from cloudinary.models import CloudinaryField   class BaseModel(models.Model):  created\_at = models.DateTimeField(auto\_now\_add=True)  updated\_at = models.DateTimeField(auto\_now=True)   class Meta:  abstract = True   class Role(models.Model):  NTV = 'NTV'  NTD = 'NTD'  ADMIN = 'ADMIN'  ROLE\_CHOICES = [  (NTV, 'Người tìm việc'),  (NTD, 'Nhà tuyển dụng'),  (ADMIN, 'Quản trị viên'),  ]  role\_name = models.CharField(max\_length=20, choices=ROLE\_CHOICES, unique=True)   def \_\_str\_\_(self):  return self.get\_role\_name\_display()   class Meta:  verbose\_name = "Vai trò"  verbose\_name\_plural = "Các vai trò"  ordering = ['role\_name']   class User(AbstractUser):  email = models.EmailField(unique=True)  avatar = CloudinaryField(null=True, blank=True, folder='avatars')  active\_role = models.ForeignKey(Role, on\_delete=models.SET\_NULL, null=True, blank=True)   def \_\_str\_\_(self):  return self.username   @property  def avatar\_url(self):  return self.avatar.url if self.avatar else settings.STATIC\_URL + 'images/default\_avatar.png'   class Meta:  verbose\_name = "Người dùng"  verbose\_name\_plural = "Người dùng"  ordering = ['username']   class UserRole(models.Model):  user = models.ForeignKey(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE, related\_name='user\_roles')  role = models.ForeignKey(Role, on\_delete=models.CASCADE, related\_name='role\_users')  is\_approved = models.BooleanField(default=False)  approved\_at = models.DateTimeField(null=True, blank=True)  approved\_by = models.ForeignKey(settings.AUTH\_USER\_MODEL, on\_delete=models.SET\_NULL, null=True, blank=True,  related\_name='approved\_roles')   class Meta:  unique\_together = ('user', 'role')  verbose\_name = "Vai trò người dùng"  verbose\_name\_plural = "Các vai trò người dùng"  ordering = ['user\_\_username']   def \_\_str\_\_(self):  status = '(Đã phê duyệt)' if self.is\_approved else ''  return f"{self.user.username} - {self.role.get\_role\_name\_display()} {status}"   class Skill(models.Model):  name = models.CharField(max\_length=100, unique=True)   def \_\_str\_\_(self):  return self.name   class Meta:  verbose\_name = "Kỹ năng"  verbose\_name\_plural = "Các kỹ năng"  ordering = ['name']   class NtvProfile(BaseModel):  user = models.OneToOneField(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE, primary\_key=True,  related\_name='ntv\_profile')  summary = models.TextField(blank=True, null=True)  experience = models.TextField(blank=True, null=True)  education = models.TextField(blank=True, null=True)  skills = models.ManyToManyField(Skill, blank=True)  phone\_number = models.CharField(max\_length=15, blank=True, null=True)  date\_of\_birth = models.DateField(null=True, blank=True)  GENDER\_CHOICES = [('M', 'Nam'), ('F', 'Nữ'), ('O', 'Khác')]  gender = models.CharField(max\_length=1, choices=GENDER\_CHOICES, blank=True, null=True)   def \_\_str\_\_(self):  return f"Hồ sơ NTV của {self.user.username}"   class Meta:  verbose\_name = "Hồ sơ Người tìm việc"  verbose\_name\_plural = "Hồ sơ Người tìm việc"  ordering = ['user\_\_username']   class NtdProfile(BaseModel):  user = models.OneToOneField(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE, primary\_key=True,  related\_name='ntd\_profile')  company\_name = models.CharField(max\_length=255)  company\_website = models.URLField(blank=True, null=True)  company\_description = models.TextField(blank=True, null=True)  industry = models.CharField(max\_length=100, blank=True, null=True)  address = models.CharField(max\_length=255, blank=True, null=True)  company\_logo = CloudinaryField(blank=True, null=True, folder='company\_logos')   def \_\_str\_\_(self):  return f"{self.company\_name} ({self.user.username})"   class Meta:  verbose\_name = "Hồ sơ Nhà tuyển dụng"  verbose\_name\_plural = "Hồ sơ Nhà tuyển dụng"  ordering = ['company\_name']   class CV(BaseModel):  ntv\_profile = models.ForeignKey(NtvProfile, on\_delete=models.CASCADE, related\_name='cvs')  file\_name = models.CharField(max\_length=255, blank=True, null=True)  file\_path = CloudinaryField(resource\_type='raw', folder='cvs')  version\_name = models.CharField(max\_length=100, blank=True, null=True)  is\_default = models.BooleanField(default=False)  is\_deleted = models.BooleanField(default=False)   def \_\_str\_\_(self):  return f"CV {self.version\_name or self.file\_name or self.id}"   class Meta:  verbose\_name = "CV"  verbose\_name\_plural = "Các CV"  ordering = ['file\_name']   class JobPosting(BaseModel):  id = models.UUIDField(primary\_key=True, default=uuid.uuid4, editable=False)  ntd\_profile = models.ForeignKey(NtdProfile, on\_delete=models.CASCADE, related\_name='job\_postings')  title = models.CharField(max\_length=255)  slug = models.SlugField(max\_length=255, unique=True, blank=True)  description = models.TextField()  location = models.CharField(max\_length=255)  salary\_min = models.FloatField(null=True, blank=True)  salary\_max = models.FloatField(null=True, blank=True)  experience\_required = models.CharField(max\_length=100, blank=True, null=True)  JOB\_TYPE\_CHOICES = [('Full-time', 'Toàn thời gian'), ('Part-time', 'Bán thời gian'), ('Freelance', 'Freelance'),  ('Internship', 'Thực tập')]  job\_type = models.CharField(max\_length=20, choices=JOB\_TYPE\_CHOICES)  is\_active = models.BooleanField(default=True)   def save(self, \*args, \*\*kwargs):  if not self.slug:  base\_slug = slugify(self.title)  slug = base\_slug  count = 1  while JobPosting.objects.filter(slug=slug).exists():  slug = f"{base\_slug}-{count}"  count += 1  self.slug = slug  super().save(\*args, \*\*kwargs)   def \_\_str\_\_(self):  return f"{self.title} tại {self.ntd\_profile.company\_name}"   class Meta:  verbose\_name = "Bài đăng tuyển dụng"  verbose\_name\_plural = "Các bài đăng tuyển dụng"  ordering = ['created\_at']   class Application(BaseModel):  user = models.ForeignKey(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE, related\_name='applications')  job\_posting = models.ForeignKey(JobPosting, on\_delete=models.CASCADE, related\_name='applications')  cv = models.ForeignKey(CV, on\_delete=models.SET\_NULL, null=True, blank=True, related\_name='applications')  STATUS\_CHOICES = [  ('Applied', 'Đã ứng tuyển'),  ('Viewed', 'NTD đã xem'),  ('Interviewing', 'Đang phỏng vấn'),  ('Offered', 'Đã mời nhận việc'),  ('Rejected', 'Đã từ chối'),  ('Withdrawn', 'Ứng viên đã rút')  ]  status = models.CharField(max\_length=20, choices=STATUS\_CHOICES, default='Applied')  cover\_letter = models.TextField(blank=True, null=True)   class Meta:  unique\_together = ('user', 'job\_posting')  verbose\_name = "Đơn ứng tuyển"  verbose\_name\_plural = "Các đơn ứng tuyển"  ordering = ['user\_\_username']   def \_\_str\_\_(self):  return f"{self.user.username} ứng tuyển vào {self.job\_posting.title}"   class Message(BaseModel):  sender = models.ForeignKey(settings.AUTH\_USER\_MODEL, related\_name='sent\_messages', on\_delete=models.CASCADE)  recipient = models.ForeignKey(settings.AUTH\_USER\_MODEL, related\_name='received\_messages', on\_delete=models.CASCADE)  content = models.TextField()  is\_read = models.BooleanField(default=False)  read\_at = models.DateTimeField(null=True, blank=True)   def \_\_str\_\_(self):  return f"From {self.sender} to {self.recipient}: {self.content[:20]}..."   class Meta:  verbose\_name = "Tin nhắn"  verbose\_name\_plural = "Các tin nhắn"  ordering = ['created\_at']   class Interview(BaseModel):  application = models.ForeignKey(Application, on\_delete=models.CASCADE, related\_name='interviews')  scheduled\_time = models.DateTimeField()  platform\_link = models.URLField(blank=True, null=True)  STATUS\_CHOICES = [('Scheduled', 'Đã lên lịch'), ('Completed', 'Đã hoàn thành'), ('Cancelled', 'Đã hủy')]  status = models.CharField(max\_length=20, choices=STATUS\_CHOICES, default='Scheduled')  notes\_ntd = models.TextField(blank=True, null=True)  notes\_ntv = models.TextField(blank=True, null=True)  RESULT\_CHOICES = [('Passed', 'Đạt'), ('Failed', 'Không đạt'), ('Pending', 'Chờ kết quả')]  result = models.CharField(max\_length=20, choices=RESULT\_CHOICES, default='Pending', blank=True, null=True)   def \_\_str\_\_(self):  return f"Phỏng vấn {self.application}"   class Meta:  verbose\_name = "Cuộc phỏng vấn"  verbose\_name\_plural = "Các cuộc phỏng vấn"  ordering = ['scheduled\_time']   class Notification(BaseModel):  recipient = models.ForeignKey(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE, related\_name='notifications')  sender = models.ForeignKey(settings.AUTH\_USER\_MODEL, null=True, blank=True, on\_delete=models.SET\_NULL,  related\_name="sent\_notifications")  message = models.TextField()  TYPE\_CHOICES = [  ('NewJob', 'Việc làm mới phù hợp'),  ('StatusUpdate', 'Cập nhật trạng thái ứng tuyển'),  ('InterviewReminder', 'Nhắc lịch phỏng vấn'),  ('ChatMessage', 'Tin nhắn mới'),  ('System', 'Thông báo hệ thống')  ]  type = models.CharField(max\_length=20, choices=TYPE\_CHOICES)  related\_url = models.URLField(blank=True, null=True)  is\_read = models.BooleanField(default=False)  read\_at = models.DateTimeField(null=True, blank=True)   def \_\_str\_\_(self):  return f"Thông báo cho {self.recipient.username}: {self.get\_type\_display()}"   class Meta:  verbose\_name = "Thông báo"  verbose\_name\_plural = "Các thông báo"  ordering = ['created\_at'] |

## Xây dựng API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Endpoint | Phương thức | Mô tả | Yêu cầu quyền truy cập |
| 1 | /nguoidung/ | GET | Lấy danh sách người dùng | Admin |
| 2 | /nguoidung/ | POST | Tạo tài khoản người dùng mới | Mọi người (AllowAny) |
| 3 | /nguoidung/{id}/ | PUT/PATCH | Cập nhật người dùng | Admin |
| 4 | /nguoidung/{id}/ | DELETE | Xóa người dùng | Admin |
| 5 | /nguoidung/nguoi\_dung\_hien\_tai/ | GET | Lấy thông tin người dùng hiện tại | Đã đăng nhập (IsAuthenticated) |
| 6 | /nguoitimviec/ | GET | Lấy danh sách người tìm việc | Admin |
| 7 | /nguoitimviec/ | POST | Tạo mới người tìm việc | Đã đăng nhập |
| 8 | /nguoitimviec/{id}/ | PUT/PATCH/DELETE | Cập nhật/Xóa người tìm việc | Người tìm việc (IsNguoiTimViec) |
| 9 | /nhatuyendung/ | GET | Lấy danh sách nhà tuyển dụng | Admin |
| 10 | /nhatuyendung/ | POST | Tạo mới nhà tuyển dụng | Đã đăng nhập |
| 11 | /nhatuyendung/{id}/ | PUT/PATCH/DELETE | Cập nhật/Xóa nhà tuyển dụng | Nhà tuyển dụng (IsNhaTuyenDung) |
| 12 | /vieclam/ | GET | Lấy danh sách việc làm đang hoạt động | Công khai (AllowAny) hoặc Admin |
| 13 | /vieclam/ | POST | Tạo việc làm mới | Nhà tuyển dụng hoặc Admin |
| 14 | /vieclam/{id}/ | PUT/PATCH/DELETE | Cập nhật/Xóa việc làm | Nhà tuyển dụng hoặc Admin |
| 15 | /vieclam/{id}/ | GET | Lấy chi tiết việc làm | Công khai hoặc Admin |
| 16 | /cv/ | GET | Lấy danh sách CV thuộc người tìm việc hoặc Admin | Người tìm việc / Admin |
| 17 | /cv/ | POST | Tạo mới CV | Người tìm việc |
| 18 | /cv/{id}/ | PUT/PATCH/DELETE | Cập nhật/Xóa CV | Người tìm việc / Admin |
| 19 | /yeucautuyendung/ | GET | Lấy danh sách yêu cầu tuyển dụng | Admin hoặc chủ sở hữu (tùy loại tài khoản) |
| 20 | /yeucautuyendung/ | POST | Tạo yêu cầu tuyển dụng (ứng tuyển việc làm) | Người tìm việc |
| 21 | /yeucautuyendung/{id}/ | GET | Lấy chi tiết yêu cầu tuyển dụng | Người tìm việc hoặc Nhà tuyển dụng (theo quyền custom) |
| 22 | /yeucautuyendung/{id}/ | DELETE | Xóa yêu cầu tuyển dụng | Admin |
| 23 | /yeucautuyendung/{id}/cap-nhat-trang-thai/ | PATCH | Nhà tuyển dụng cập nhật trạng thái ứng tuyển | Nhà tuyển dụng (IsNhaTuyenDungXemVaChamKetQua) |

2.2.1. Đăng nhập người dùng

- Mô tả: API Đăng Nhập Người Dùng sử dụng JWT (JSON Web Token) để chứng thực và phân quyền người dùng. Chỉ admin mới có quyền xem, thêm, sửa, và xóa danh sách người dùng, trong khi khách truy cập có thể tự tạo tài khoản người dùng qua API đăng ký.

- URL: /token/ - POST

- Body data

|  |
| --- |
| {  "username": "pnam", // Tên người dùng  "password": "0212" // Mật khẩu người dùng } |

- Trả Lời Thành Công (200 OK): Nếu thông tin đăng nhập chính xác, API sẽ trả về JWT token cho người dùng. Token này sẽ được sử dụng để thực hiện các yêu cầu bảo mật trong hệ thống.

Mã trạng thái HTTP: 200 OK

Trả về hai token:

- access: Token dùng để chứng thực người dùng trong các yêu cầu tiếp theo.

- refresh: Token dùng để làm mới access token khi hết hạn.

{

"access": "your\_jwt\_access\_token\_here", // Access Token

"refresh": "your\_jwt\_refresh\_token\_here" // Refresh Token

}

- Trả Lời Lỗi (400 Bad Request)

Nếu thông tin đăng nhập không chính xác (tên người dùng hoặc mật khẩu sai), hệ thống sẽ trả về lỗi 400.

Mã trạng thái HTTP: 400 Bad Request

{

"detail": "No active account found with the given credentials"

}

- Trả Lời Lỗi (401 Unauthorized)

Nếu người dùng cung cấp thông tin không hợp lệ hoặc không có quyền truy cập vào tài nguyên yêu cầu, API sẽ trả về lỗi 401 Unauthorized.

Mã trạng thái HTTP: 401 Unauthorized

{

"detail": "Invalid credentials"

}

2.2.2. Đăng ký người dùng

Mô Tả API

API Đăng Ký Người Dùng cho phép khách truy cập tự tạo tài khoản người dùng trong hệ thống. Khi đăng ký thành công, người dùng có thể đăng nhập và nhận JWT để sử dụng các chức năng bảo mật của hệ thống.

Endpoint: /nguoi-dung/

Phương thức HTTP: POST

Yêu Cầu Đầu Vào (Request Payload)

Để tạo tài khoản mới, khách truy cập cần cung cấp các thông tin sau:

{

"username": "username123", // Tên người dùng (phải là duy nhất)

"email": "email@example.com", // Địa chỉ email (phải là duy nhất)

"password": "yourpassword", // Mật khẩu của người dùng

"first\_name": "John", // Tên người dùng (không bắt buộc)

"last\_name": "Doe", // Họ người dùng (không bắt buộc)

"hinh\_dai\_dien": "image\_url" // (Tùy chọn) URL ảnh đại diện từ Cloudinary

}

Trả Lời Thành Công (201 Created)

Nếu tài khoản người dùng được tạo thành công, hệ thống sẽ trả về thông tin người dùng cùng URL ảnh đại diện (nếu có).

Mã trạng thái HTTP: 201 Created

Trả về thông tin người dùng bao gồm các trường thông tin như `username`, `email`, `first\_name`, `last\_name`, và `hinh\_dai\_dien` (nếu có).

{

"username": "username123",

"email": "email@example.com",

"first\_name": "John",

"last\_name": "Doe",

"hinh\_dai\_dien": "https://res.cloudinary.com/your\_cloud\_name/image/upload/v1744970332/b4wzj4zkmbbzab2gilzp.jpg"

}

Trả Lời Lỗi (400 Bad Request)

Nếu thiếu các trường bắt buộc hoặc có lỗi trong dữ liệu, API sẽ trả về lỗi 400 với thông báo chi tiết.

Mã trạng thái HTTP: 400 Bad Request

{

"detail": "Username, email, and password are required."

}

Trả Lời Lỗi (409 Conflict)

Nếu tên người dùng hoặc email đã tồn tại trong hệ thống, API sẽ trả về lỗi 409.

Mã trạng thái HTTP: 409 Conflict

{

"detail": "Username or email already exists."

}

## Kiểm thử API

# **PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG REACT NATIVE**

## Cấu trúc ứng dụng

## Xây dựng giao diện

## Tích hợp API

## Tối ưu hiệu suất

# HỆ THỐNG <TÊN ĐỀ TÀI>

## Kết quả đạt được của đề tài

## Các chức năng hệ thống

### Chức năng 1

Giới thiệu chức năng 1

Hình HỆ THỐNG <TÊN ĐỀ TÀI>.2: Hình chức năng 1 (chụp từ kết quả hệ thống)

### Chức năng 2

Giới thiệu chức năng 2

Hình HỆ THỐNG <TÊN ĐỀ TÀI>.3: Hình chức năng 2 (chụp từ kết quả hệ thống)

Hình 3.3: Kết quả

* **Đặc tả các use case đăng ký**

|  |  |
| --- | --- |
| **Use case id** | UC001\_DangKyTaiKhoan |
| **Tên use case** | Đăng ký tài khoản |
| **Mô tả vấn tắt** | Người dùng cung cấp thông tin đăng ký, xác thực email/SĐT để tạo tài khoản mới trên hệ thống. |
| **Actor chính** | Người chưa đăng nhập |
| **Actor phụ** | Hệ thống xác thực email/SMS |
| **Tiền điều kiện** | Người dùng chưa có tài khoản hợp lệ trên hệ thống. |
| **Hậu điều kiện** | Nếu thông tin hợp lệ, tài khoản của người dùng được kích hoạt và họ có thể đăng nhập vào hệ thống.  Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi. |
| **Luồng hoạt động** | B1. Người dùng mở giao diện đăng ký.  B2. Nhập thông tin cá nhân (Tên, Email/SĐT, Mật khẩu...).  B3. Hệ thống kiểm tra tính hợp lệ của thông tin.  B4. Nếu thông tin hợp lệ, hệ thống gửi mã xác thực (OTP) qua email/SMS.  B5. Người dùng nhập mã xác thực OTP.  B6. Hệ thống kiểm tra mã OTP.  B7. Nếu OTP hợp lệ, tài khoản được kích hoạt.  B8. Người dùng nhận thông báo đăng ký thành công và có thể đăng nhập. |
| **Luồng thay thế** | *B2.1 Người dùng đã có tài khoản*  B2.1.1 Hệ thống hiển thị thông báo “Tài khoản đã tồn tại”.  B2.1.2 Gợi ý người dùng đăng nhập thay vì đăng ký mới.  *B5.1 Người dùng không nhận được mã OTP*  B5.1.1 Hệ thống cung cấp tùy chọn “Gửi lại OTP”.  B5.1.2 Người dùng yêu cầu gửi lại mã OTP qua email/SMS.  B5.1.3 Hệ thống gửi lại mã mới. |
| **Luồng ngoại lệ** | *B3.1 Nhập sai thông tin đăng ký*  B3.1.1 Email/SĐT không hợp lệ → Hiển thị thông báo lỗi.  B3.1.2 Mật khẩu quá yếu → Yêu cầu người dùng nhập mật khẩu mạnh hơn.  *B6.1 Nhập sai mã OTP nhiều lần*  B6.1.1 Nếu nhập sai OTP quá 5 lần, hệ thống tạm khóa xác thực 5 phút.  B6.1.2 Người dùng phải đợi hoặc liên hệ hỗ trợ. |

*Bảng 1.1: Bảng thể hiện use case đăng ký tài khoản*

* **Đặc tả use case đăng nhập**

|  |  |
| --- | --- |
| **Use case id** | UC002\_DangNhapTaiKhoan |
| **Tên use case** | Đăng ký tài khoản |
| **Mô tả vấn tắt** | Người dùng nhập thông tin username và password hoặc sử dụng đăng nhập bằng mạng xã hội (MXH) để truy cập vào hệ thống. |
| **Actor chính** | Người dùng (Người tìm việc / Nhà tuyển dụng) |
| **Actor phụ** | Hệ thống xác thực email/SMS |
| **Tiền điều kiện** | Người dùng đã có tài khoản hợp lệ trên hệ thống. |
| **Hậu điều kiện** | Nếu thông tin hợp lệ, người dùng được đăng nhập vào hệ thống.  Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi. |
| **Luồng hoạt động** | B1. Người dùng mở giao diện đăng nhập.  B2. Nhập email/SĐT và mật khẩu hoặc chọn đăng nhập bằng MXH.  B3. Hệ thống kiểm tra thông tin đăng nhập.  B4. Nếu thông tin hợp lệ, hệ thống cho phép người dùng truy cập vào hệ thống.  B5. Người dùng có thể sử dụng các chức năng của hệ thống. |
| **Luồng thay thế** | *B2.1 Người dùng quên mật khẩu*  B2.1.1 Người dùng nhấn vào "Quên mật khẩu".  B2.1.2 Hệ thống yêu cầu nhập email hoặc số điện thoại đã đăng ký.  B2.1.3 Hệ thống gửi mã OTP hoặc đường dẫn đặt lại mật khẩu.  B2.1.4 Người dùng đặt lại mật khẩu mới.  B2.1.5 Quay lại bước 2 của luồng chính.  *B2.2 Đăng nhập bằng MXH*  B2.2.1 Người dùng chọn đăng nhập bằng Google/Facebook.  B2.2.2 Hệ thống chuyển hướng đến trang xác thực của MXH.  B2.2.3 Nếu xác thực thành công, người dùng được đăng nhập vào hệ thống. |
| **Luồng ngoại lệ** | *B3.1 Đăng nhập thất bại do nhập sai thông tin*  B3.1.1 Hệ thống hiển thị thông báo “Tên đăng nhập hoặc mật khẩu không đúng”.  B3.1.2 Người dùng có thể thử lại.  B3.1.3 Nếu nhập sai quá 5 lần, tài khoản bị tạm khóa.  *B3.2 Hệ thống lỗi hoặc mất kết nối*  B3.2.1 Hệ thống hiển thị thông báo “Lỗi hệ thống, vui lòng thử lại sau”.  B3.2.2 Người dùng không thể đăng nhập cho đến khi hệ thống hoạt động ổn định. |

*Bảng 1.2: Bảng thể hiện use case đăng nhập vào hệ thống*

* **Đặc tả use case đăng tin tuyển dụng**

|  |  |
| --- | --- |
| **Use case id** | UC003\_TimKiemViecLam |
| **Tên use case** | Tìm kiếm việc làm |
| **Mô tả vấn tắt** | Người dùng (cả đã đăng nhập và chưa đăng nhập) có thể tìm kiếm việc làm theo từ khóa, ngành nghề, địa điểm, mức lương và các tiêu chí khác. |
| **Actor chính** | Người tìm việc (đã đăng nhập hoặc chưa đăng nhập) |
| **Actor phụ** | Hệ thống gợi ý việc làm (Recommendation System) |
| **Tiền điều kiện** | Hệ thống đã có dữ liệu về các tin tuyển dụng. |
| **Hậu điều kiện** | Hiển thị danh sách công việc phù hợp với tiêu chí tìm kiếm.  Nếu không có kết quả phù hợp, hệ thống hiển thị thông báo "Không tìm thấy công việc nào phù hợp". |
| **Luồng hoạt động** | B1. Người dùng mở giao diện tìm kiếm việc làm.  B2. Nhập từ khóa tìm kiếm (ví dụ: tên công việc, công ty, kỹ năng).  B3. Chọn các tiêu chí lọc (ngành nghề, địa điểm, mức lương, kinh nghiệm, hình thức làm việc, v.v.).  B4. Hệ thống xử lý truy vấn, tìm kiếm công việc phù hợp trong cơ sở dữ liệu.  B5. Hiển thị danh sách công việc phù hợp.  B6. Người dùng xem chi tiết công việc hoặc thực hiện tìm kiếm mới. |
| **Luồng thay thế** | *B3.1 Người dùng không nhập tiêu chí tìm kiếm*  B3.1.1 Hệ thống hiển thị danh sách công việc mới nhất hoặc phổ biến.  *B3.2 Người dùng muốn sử dụng gợi ý công việc (AI Recommendation System)*  B3.2.1 Hệ thống yêu cầu nhập thông tin cá nhân (nếu chưa đăng nhập).  B3.2.2 Hệ thống phân tích hồ sơ (nếu có) và đề xuất công việc phù hợp.  B3.2.3 Người dùng chọn công việc từ danh sách gợi ý. |
| **Luồng ngoại lệ** | *B4.1 Không có công việc phù hợp*  B4.1.1 Hệ thống hiển thị thông báo: "Không tìm thấy công việc nào phù hợp".  B4.1.2 Đề xuất người dùng thử lại với tiêu chí khác.  *B4.2 Lỗi kết nối hệ thống*  B4.2.1 Hệ thống hiển thị thông báo: "Lỗi hệ thống, vui lòng thử lại sau". |

*Bảng 1.3: Bảng thể hiện use case đăng tin tuyển dụng*

* **Đặc tả use case đăng tin tuyển dụng**

|  |  |
| --- | --- |
| **Use case id** | UC004\_DangTinTuyenDung |
| **Tên use case** | Đăng tin tuyển dụng |
| **Mô tả vấn tắt** | Nhà tuyển dụng có thể tạo, chỉnh sửa, và đăng tải thông tin tuyển dụng lên hệ thống để tìm kiếm ứng viên. |
| **Actor chính** | Nhà tuyển dụng |
| **Actor phụ** | Hệ thống gợi ý việc làm (Recommendation System) |
| **Tiền điều kiện** | Nhà tuyển dụng đã đăng nhập vào hệ thống.  Có quyền đăng tin tuyển dụng. |
| **Hậu điều kiện** | Tin tuyển dụng được đăng tải và hiển thị trong danh sách việc làm.  Ứng viên có thể xem và ứng tuyển vào tin tuyển dụng. |
| **Luồng hoạt động** | B1. Nhà tuyển dụng mở giao diện đăng tin tuyển dụng.  B2. Nhập thông tin công việc (tên công việc, mô tả, yêu cầu, mức lương, địa điểm, v.v.).  B3. Chọn hình thức tuyển dụng (toàn thời gian, bán thời gian, thực tập, remote, v.v.).  B4. Hệ thống kiểm tra dữ liệu và xác nhận thông tin.  B5. Nhà tuyển dụng nhấn “Đăng tin”.  B6. Hệ thống lưu tin tuyển dụng và hiển thị công khai. |
| **Luồng thay thế** | *B4.1 Nếu thiếu thông tin bắt buộc, hệ thống thông báo lỗi và yêu cầu bổ sung.*  *B6.1 Nếu nhà tuyển dụng muốn lưu bản nháp, hệ thống cho phép lưu mà không đăng tải ngay.* |
| **Luồng ngoại lệ** | *B5.1 Nếu hệ thống lỗi khi lưu tin, thông báo “Hệ thống đang gặp sự cố, vui lòng thử lại sau”.* |

*Bảng 1.4: Bảng thể hiện use case đăng tin tuyển dụng*

* **Đặc tả use case xem thống kê hồ sơ ứng tuyển**

|  |  |
| --- | --- |
| **Use case id** | UC005\_XemThongKeHoSo |
| **Tên use case** | Xem thống kê hồ sơ ứng tuyển |
| **Mô tả vấn tắt** | Nhà tuyển dụng có thể xem thống kê về số lượng ứng viên đã ứng tuyển vào tin tuyển dụng của họ. |
| **Actor chính** | Nhà tuyển dụng |
| **Actor phụ** | Không có |
| **Tiền điều kiện** | Nhà tuyển dụng đã đăng nhập vào hệ thống.  Có ít nhất một tin tuyển dụng đã đăng. |
| **Hậu điều kiện** | Hệ thống hiển thị thống kê số lượng hồ sơ ứng tuyển theo từng vị trí.  Nhà tuyển dụng có thể lọc và phân loại hồ sơ theo tiêu chí. |
| **Luồng hoạt động** | B1. Nhà tuyển dụng mở trang quản lý tin tuyển dụng.  B2. Chọn một tin tuyển dụng để xem thống kê.  B3. Hệ thống hiển thị số lượng ứng viên đã ứng tuyển.  B4. Nhà tuyển dụng có thể lọc danh sách theo tiêu chí (kinh nghiệm, kỹ năng, mức lương mong muốn, v.v.). |
| **Luồng thay thế** | *B3.1 Nếu không có hồ sơ ứng tuyển nào, hiển thị thông báo “Chưa có ứng viên nào ứng tuyển vào vị trí này”.* |
| **Luồng ngoại lệ** | *B3.2 Nếu hệ thống lỗi, hiển thị “Không thể tải dữ liệu, vui lòng thử lại sau”.* |

*Bảng 1.5: Bảng thể hiện use case xem thống kê hồ sơ ứng tuyển*

* **Đặc tả use case liên hệ ứng viên**

|  |  |
| --- | --- |
| **Use case id** | UC005\_LienHeUngVien |
| **Tên use case** | Liên hệ ứng viên |
| **Mô tả vấn tắt** | Nhà tuyển dụng có thể gửi tin nhắn hoặc email để liên hệ trực tiếp với ứng viên. |
| **Actor chính** | Nhà tuyển dụng |
| **Actor phụ** | Không có |
| **Tiền điều kiện** | Nhà tuyển dụng đã đăng nhập vào hệ thống.  Nhà tuyển dụng có quyền xem danh sách ứng viên. |
| **Hậu điều kiện** | Ứng viên nhận được tin nhắn hoặc email từ nhà tuyển dụng. |
| **Luồng hoạt động** | B1. Nhà tuyển dụng mở danh sách ứng viên đã ứng tuyển.  B2. Chọn một ứng viên để liên hệ.  B3. Nhập nội dung tin nhắn hoặc email.  B4. Nhấn gửi.  B5. Hệ thống xác nhận và gửi tin nhắn/email đến ứng viên. |
| **Luồng thay thế** | *B3.1 Nếu nhà tuyển dụng muốn đặt lịch phỏng vấn, có thể chọn “Gửi lời mời phỏng vấn”.* |
| **Luồng ngoại lệ** | *B5.1 Nếu ứng viên không cho phép nhận tin nhắn, hiển thị “Ứng viên này đã ẩn thông tin liên hệ”.*  *B5.2 Nếu hệ thống gặp lỗi, hiển thị “Không thể gửi tin nhắn, vui lòng thử lại sau”.* |

*Bảng 1.6: Bảng thể hiện use case liên hệ ứng viên*

* **Đặc tả use case quản lý ứng viên**

|  |  |
| --- | --- |
| **Use case id** | UC007\_QuanLyUngVien |
| **Tên use case** | Quản lý ứng viên |
| **Mô tả vấn tắt** | Nhà tuyển dụng có thể xem, phân loại, lưu trữ và đánh giá hồ sơ ứng viên. |
| **Actor chính** | Nhà tuyển dụng |
| **Actor phụ** | Không có |
| **Tiền điều kiện** | Nhà tuyển dụng đã đăng nhập vào hệ thống.  Có ít nhất một tin tuyển dụng có ứng viên ứng tuyển. |
| **Hậu điều kiện** | Nhà tuyển dụng có danh sách ứng viên được phân loại theo trạng thái.  Nhà tuyển dụng có thể lưu lại những ứng viên tiềm năng |
| **Luồng hoạt động** | B1. Nhà tuyển dụng mở trang quản lý ứng viên.  B2. Chọn tin tuyển dụng cần xem danh sách ứng viên.  B3. Hệ thống hiển thị danh sách ứng viên kèm hồ sơ.  B4. Nhà tuyển dụng có thể đánh giá, phân loại ứng viên theo trạng thái (Đang xem xét, Mời phỏng vấn, Đã tuyển, Loại bỏ).  B5. Lưu lại thông tin ứng viên. |
| **Luồng thay thế** | *B3.1 Nhà tuyển dụng có thể tải danh sách ứng viên dưới dạng file Excel.* |
| **Luồng ngoại lệ** | *B3.2 Nếu không có ứng viên nào, hiển thị “Chưa có hồ sơ ứng viên nào cho vị trí này”.*  *B5.1 Nếu hệ thống lỗi, hiển thị “Không thể lưu thông tin, vui lòng thử lại sau”.* |

*Bảng 1.7: Bảng thể hiện use case quản lý ứng viên*

* **Đặc tả use case chat và phỏng vấn với nhà tuyển dụng**

|  |  |
| --- | --- |
| **Use case id** | UC008\_PhongVan |
| **Tên use case** | Chat và phỏng vấn với nhà tuyển dụng |
| **Mô tả vấn tắt** | Ứng viên có thể trao đổi với nhà tuyển dụng qua hệ thống chat hoặc tham gia phỏng vấn trực tuyến qua video call. |
| **Actor chính** | Người tìm việc, Nhà tuyển dụng |
| **Actor phụ** | Không có |
| **Tiền điều kiện** | Ứng viên và nhà tuyển dụng đã đăng ký tài khoản và đăng nhập vào hệ thống.  Nhà tuyển dụng đã tạo tin tuyển dụng và ứng viên đã nộp hồ sơ.  Hệ thống hỗ trợ chức năng nhắn tin và video call. |
| **Hậu điều kiện** | Cuộc trò chuyện/phỏng vấn được lưu lại trên hệ thống.  Nhà tuyển dụng có thể đánh giá ứng viên hoặc tiếp tục quá trình tuyển dụng. |
| **Luồng hoạt động** | B1. Ứng viên đăng nhập vào hệ thống.  B2. Ứng viên truy cập vào danh sách công việc đã ứng tuyển.  B3. Ứng viên chọn một nhà tuyển dụng để chat hoặc phỏng vấn.  B4. Hệ thống hiển thị khung chat, cho phép nhắn tin trực tiếp.  B5. Nếu nhà tuyển dụng yêu cầu phỏng vấn, hệ thống hiển thị tùy chọn gọi video.  B6. Nếu ứng viên đồng ý, hệ thống kết nối cuộc gọi video.  B7. Cuộc trò chuyện/phỏng vấn diễn ra giữa hai bên.  B8. Sau khi kết thúc, hệ thống lưu lại lịch sử cuộc trò chuyện. |
| **Luồng thay thế** | *B3.1: Nếu ứng viên chưa có cuộc trò chuyện nào, hệ thống hiển thị thông báo "Không có tin nhắn nào".*  *B5.1: Nếu ứng viên muốn đổi từ chat sang gọi video, hệ thống cung cấp nút "Bắt đầu phỏng vấn".* |
| **Luồng ngoại lệ** | *B4.1: Nếu nhà tuyển dụng không phản hồi trong 24 giờ, hệ thống gửi nhắc nhở.*  *B6.1: Nếu kết nối mạng kém, hệ thống hiển thị cảnh báo và cho phép thử lại.*  *B7.1: Nếu cuộc phỏng vấn bị gián đoạn, hệ thống tự động lưu trạng thái và cho phép kết nối lại sau.* |

*Bảng 1.8: Bảng thể hiện use case chat và phỏng vấn với nhà tuyển dụng*

* **Đặc tả use case ứng tuyển việc làm**
* **Đặc tả use case tạo và cật nhật hồ sơ ứng tuyển**
* **Đặc tả use case nhận thông báo tuyển dụng**
* **Đặc tả use case xem thống kê hồ sơ cá nhân**
* **Đặc tả use case quản lý hệ thống**
* **Đặc tả use case xét duyệt tin tuyển dụng**