

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



ĐỒ ÁN TỔNG HỢP

**Nền tảng kết nối nhà tuyển dụng và người
tìm việc - BKJOB**

Giảng viên hướng dẫn: Thầy Lê Đình Thuận

Sinh viên: Nguyễn Hữu Khang – 2311449
Đặng Phạm Gia Long – 2311892
Đương Khả Vân – 2313866
Nguyễn Văn Công Thành – 2313133
Trần Thế Đại Phát – 2212537

Học kỳ: 251

TP. HỒ CHÍ MINH, THÁNG 12/2025

Mục lục

1 Requirement elicitation	4
1.1 Perform general analysis of related works	5
1.2 Functional requirements	5
1.3 Non-functional requirements	6
1.4 Use-case diagram	8
1.5 Use-case senarios	9
1.5.1 Đăng ký	9
1.5.2 Đăng nhập	10
1.5.3 Quản lý thông tin Dashboard (Recruiter)	11
1.5.4 Đăng tin tuyển dụng	12
1.5.5 Xem danh sách Jobseeker được gợi ý	12
1.5.6 Xem chi tiết hồ sơ Jobseeker	13
1.5.7 Quản lý danh sách Jobseeker đã ứng tuyển	13
1.5.8 Quản lý tin tuyển dụng (My Jobs)	14
1.5.9 Xem và chỉnh sửa hồ sơ cá nhân	15
1.5.10 Tìm kiếm công việc	15
1.5.11 Lưu công việc	16
1.5.12 Xem chi tiết công việc	16
1.5.13 Ứng tuyển công việc	17
1.5.14 Xem danh sách các công việc đã lưu	17
1.5.15 Xem danh sách công việc được đề xuất	18
1.5.16 Xem danh sách các công ty	18
1.5.17 Tìm kiếm công ty	19
1.5.18 Xem danh sách công ty theo thứ hạng có người ứng tuyển từ nhiều đến ít	19
1.5.19 Xem chi tiết công ty	20
1.5.20 Chatting	20
1.5.21 Nhận thông báo theo dõi về công việc đã ứng tuyển	21
1.5.22 Báo cáo công ty	21
2 System modeling	22
2.1 Activity diagrams	22
2.2 Sequence diagrams	22
2.2.1 Luồng đăng nhập (Login)	22
2.2.2 Luồng đăng ký (Register)	23
2.2.3 Luồng xác thực email (Verify Email)	25
2.2.4 Luồng tạo người dùng theo vai trò (Create User With Type)	27
2.2.5 Luồng quên mật khẩu (Forgot Password)	28
2.2.6 Luồng đặt lại mật khẩu (Reset Password)	29
2.2.7 Luồng cập nhật hồ sơ JobSeeker (Update Profile JobSeeker)	31
2.2.8 Luồng tìm kiếm công việc (Search Job)	32
2.2.9 Luồng đăng nhập bằng Google (Login with Google)	33
2.2.10 Luồng ứng tuyển công việc (Apply Job)	35
2.2.11 Luồng đăng tin tuyển dụng (Post Job)	37
2.3 Class diagrams	38



3	Architecture design	38
3.1	Database design	38
3.2	Software architecture	38
4	Implementation	38



Danh sách thành viên & Phân chia nhiệm vụ



1 Requirement elicitation

Trong thời đại số hóa hiện nay, việc tìm kiếm việc làm không chỉ dừng lại ở việc nộp hồ sơ truyền thống mà đã trở thành một quá trình phức tạp, đòi hỏi sự kết nối nhanh chóng và chính xác giữa người tìm việc và nhà tuyển dụng. Sự đa dạng của thị trường lao động cùng với mức độ cạnh tranh ngày càng cao khiến nhiều ứng viên gặp khó khăn trong việc tìm kiếm công việc phù hợp với kỹ năng, kinh nghiệm và định hướng cá nhân. Ở chiều ngược lại, các doanh nghiệp và nhà tuyển dụng cũng phải đổi mới với thách thức trong việc tiếp cận, sàng lọc và đánh giá số lượng lớn ứng viên để tìm ra những nhân sự đáp ứng đúng yêu cầu công việc.

Xuất phát từ thực tiễn đó, nhóm chúng tôi đề xuất xây dựng BKJOB, một nền tảng tuyển dụng trực tuyến nhằm kết nối hiệu quả giữa người tìm việc và nhà tuyển dụng, đồng thời tối ưu hóa toàn bộ quy trình tuyển dụng. BKJOB hướng đến việc tạo ra một môi trường thuận tiện, minh bạch và thân thiện, hỗ trợ cả hai phía trong việc tiếp cận thông tin, tương tác và đưa ra quyết định nhanh chóng, chính xác.

Đối với người tìm việc, hệ thống cho phép người dùng dễ dàng tạo và quản lý hồ sơ cá nhân, bao gồm thông tin cá nhân, kỹ năng, kinh nghiệm và lĩnh vực quan tâm. Người dùng có thể đăng ký và đăng nhập thông qua nhiều nền tảng khác nhau như email hoặc các dịch vụ xác thực bên thứ ba như Google, Github, giúp quá trình sử dụng trở nên nhanh chóng và linh hoạt. BKJOB hỗ trợ tìm kiếm việc làm theo nhiều tiêu chí khác nhau như từ khóa, địa điểm, ngành nghề, mức lương và yêu cầu kinh nghiệm, giúp ứng viên tiếp cận các cơ hội phù hợp với nhu cầu cá nhân. Trong quá trình ứng tuyển, người tìm việc có thể theo dõi trạng thái hồ sơ, lưu lại các công việc quan tâm và nhận thông báo kịp thời khi có phản hồi từ nhà tuyển dụng. Bên cạnh đó, hệ thống còn tích hợp cơ chế gợi ý thông minh, dựa trên hồ sơ cá nhân, lịch sử tìm kiếm, lịch sử ứng tuyển và các công việc đã lưu, nhằm đề xuất những vị trí phù hợp nhất cho từng người dùng.

Đối với nhà tuyển dụng, BKJOB hỗ trợ xây dựng và quản lý hồ sơ công ty, đăng tải các tin tuyển dụng với thông tin chi tiết về vị trí, yêu cầu và quyền lợi. Nhà tuyển dụng có thể theo dõi danh sách ứng viên ứng tuyển, lọc và tìm kiếm hồ sơ theo các tiêu chí cụ thể, từ đó đánh giá và đưa ra quyết định tuyển chọn phù hợp. Ngoài ra, hệ thống còn cho phép nhà tuyển dụng chủ động tìm kiếm các ứng viên tiềm năng, nâng cao khả năng tiếp cận nguồn nhân lực chất lượng. Việc phản hồi kết quả ứng tuyển được thực hiện trực tiếp trên hệ thống, giúp rút ngắn thời gian tuyển dụng và tăng hiệu quả kết nối giữa hai bên.

Bên cạnh các chức năng dành cho người dùng cuối, hệ thống BKJOB còn cung cấp các công cụ quản trị cho quản trị viên nhằm giám sát và kiểm soát toàn bộ hoạt động của nền tảng. Quản trị viên có thể quản lý người dùng, tin tuyển dụng và xử lý các báo cáo được gửi từ hệ thống, đảm bảo dữ liệu luôn chính xác và môi trường sử dụng lành mạnh. Hệ thống cũng hỗ trợ thống kê và báo cáo theo thời gian, giúp theo dõi số lượng công việc, lượt ứng tuyển và mức độ hoạt động của người dùng theo từng giai đoạn.

Về mặt kỹ thuật, BKJOB được xây dựng với định hướng đảm bảo hiệu năng, tính bảo mật và khả năng mở rộng. Thông tin nhạy cảm của người dùng, đặc biệt là mật khẩu, được mã hóa và kiểm soát truy cập chặt chẽ. Hệ thống hỗ trợ thông báo theo thời gian thực cũng như qua email, giúp người dùng không bỏ lỡ các thông tin quan trọng. Giao diện được thiết kế thân thiện, dễ sử dụng và tương thích với nhiều thiết bị, đặc biệt là trên nền tảng di động, nhằm nâng cao trải nghiệm người dùng. Ngoài ra, hệ thống hướng đến việc đáp ứng các yêu cầu về hiệu năng, khả năng bảo trì và chất lượng mã nguồn, đảm bảo quá trình phát triển và mở rộng trong tương lai diễn ra thuận lợi.

Trong giai đoạn mở rộng, BKJOB có thể tích hợp các tính năng nâng cao như lưu trữ tài



liệu trên nền tảng điện toán đám mây như AWS, với kiến trúc hướng dịch vụ và API chuẩn RESTful, BKJOB có khả năng tích hợp linh hoạt với các hệ thống khác, đáp ứng nhu cầu phát triển lâu dài và thích ứng với sự thay đổi của thị trường lao động.

1.1 Perform general analysis of related works

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của công nghệ thông tin và Internet, nhiều nền tảng tuyển dụng trực tuyến đã được xây dựng nhằm hỗ trợ kết nối giữa người tìm việc và nhà tuyển dụng. Các hệ thống này đóng vai trò quan trọng trong việc số hóa quy trình tuyển dụng, giảm thiểu chi phí và thời gian cho cả hai phía, đồng thời mở rộng phạm vi tiếp cận của thị trường lao động.

Một trong những nền tảng tuyển dụng phổ biến nhất hiện nay là LinkedIn Jobs. Hệ thống này tận dụng mạng xã hội nghề nghiệp để kết nối ứng viên và nhà tuyển dụng dựa trên hồ sơ cá nhân, kinh nghiệm làm việc và mạng lưới quan hệ. LinkedIn Jobs cho phép nhà tuyển dụng đăng tin tuyển dụng, tìm kiếm ứng viên chủ động và đánh giá hồ sơ một cách linh hoạt. Tuy nhiên, hệ thống này có nhược điểm là nhiều tính năng nâng cao chỉ thông qua các gói trả phí, gây hạn chế đối với người dùng phổ thông và doanh nghiệp nhỏ.

Bên cạnh đó, Indeed là một nền tảng tuyển dụng toàn cầu khác, nổi bật với khả năng tổng hợp tin tuyển dụng từ nhiều nguồn khác nhau. Indeed hỗ trợ tìm kiếm việc làm theo từ khóa, địa điểm và mức lương, giúp người tìm việc tiếp cận nhanh chóng với nhiều cơ hội. Tuy nhiên, hệ thống chủ yếu tập trung vào việc đăng và tìm kiếm tin tuyển dụng, trong khi các chức năng quản lý hồ sơ cá nhân, theo dõi trạng thái ứng tuyển và tương tác trực tiếp giữa hai bên còn tương đối hạn chế.

Tại thị trường Việt Nam, VietnamWorks và TopCV là hai hệ thống tuyển dụng trực tuyến được sử dụng rộng rãi. Các nền tảng này cung cấp chức năng đăng tin tuyển dụng, tạo hồ sơ ứng viên và gợi ý việc làm dựa trên thông tin người dùng. VietnamWorks có thể mạnh về số lượng doanh nghiệp lớn và dữ liệu thị trường lao động, trong khi TopCV tập trung vào việc hỗ trợ tạo CV và gợi ý công việc cho sinh viên và người mới ra trường. Tuy nhiên, phần lớn các hệ thống này vẫn còn hạn chế trong việc cá nhân hóa trải nghiệm người dùng và chưa tích hợp sâu các cơ chế gợi ý thông minh dựa trên hành vi sử dụng.

Từ việc phân tích các hệ thống tuyển dụng hiện có, có thể nhận thấy rằng mỗi nền tảng đều có những ưu điểm riêng, song vẫn tồn tại những hạn chế nhất định về trải nghiệm người dùng, khả năng tương tác hai chiều và mức độ tự động hóa trong quá trình tuyển dụng. Dựa trên những quan sát đó, hệ thống BKJOB được đề xuất nhằm kế thừa các ưu điểm của các nền tảng hiện có, đồng thời khắc phục một số hạn chế bằng cách tập trung vào việc quản lý hồ sơ tập trung, theo dõi trạng thái ứng tuyển, tích hợp cơ chế gợi ý thông minh và hỗ trợ thông báo theo thời gian thực. BKJOB hướng đến việc xây dựng một nền tảng tuyển dụng linh hoạt, dễ sử dụng, phù hợp với cả người tìm việc lẫn nhà tuyển dụng trong bối cảnh thị trường lao động hiện nay.

1.2 Functional requirements

- Người dùng:

- Đăng ký tài khoản
- Đăng nhập/Dăng xuất
- Quên mật khẩu
- Xem báo cáo tổng quan

- Người tìm việc:



- Quản lý hồ sơ cá nhân
 - Tìm kiếm công việc theo từ khóa, địa điểm, ngành nghề, mức lương, kinh nghiệm.
 - Ứng tuyển việc làm
 - Theo dõi trạng thái ứng tuyển
 - Lưu công việc yêu thích
 - Nhận thông báo về việc làm đã ứng tuyển
 - Nhận gợi ý việc làm
 - Báo cáo một công việc, nhà tuyển dụng không phù hợp
 - Tìm kiếm nhà tuyển dụng
- Nhà tuyển dụng:
 - Quản lý hồ sơ công ty
 - Đăng tin tuyển dụng
 - Quản lý tin tuyển dụng
 - Theo dõi danh sách ứng viên
 - Lọc và tìm kiếm ứng viên
 - Tìm kiếm ứng viên chủ động
 - Phản hồi kết quả ứng tuyển
 - Báo cáo một ứng viên, nhà tuyển dụng, công việc không phù hợp

- Quản trị viên:
 - Quản lý người dùng (vô hiệu hóa, cập nhật)
 - Quản lý tin tuyển dụng (vô hiệu hóa, cập nhật)
 - Xử lý báo cáo
 - Quản lý hành động của bản thân

1.3 Non-functional requirements

- Hiệu năng (Performance):
 - Hệ thống cần đảm bảo thời gian phản hồi (response time) ở mức chấp nhận được đối với các thao tác cơ bản của người dùng như đăng nhập, tìm kiếm việc làm, ứng tuyển và đăng tin tuyển dụng.
 - Thông lượng xử lý (throughput) của hệ thống cần đáp ứng được số lượng lớn người dùng truy cập đồng thời, đặc biệt trong các thời điểm cao điểm. Việc đánh giá và tối ưu các chỉ số hiệu năng sẽ được thực hiện sau khi hoàn thiện đầy đủ các chức năng chính của hệ thống.
- Toàn vẹn dữ liệu (Data Integrity)
 - Tất cả dữ liệu được nhập vào hệ thống phải được kiểm tra tính hợp lệ trước khi lưu trữ, bao gồm định dạng, giá trị và các ràng buộc liên quan.
 - Hệ thống cần đảm bảo dữ liệu được lưu trữ nhất quán, tránh trùng lặp hoặc sai lệch trong quá trình xử lý và cập nhật.



- Bảo mật dữ liệu (Data Security)

- Thông tin nhạy cảm của người dùng, đặc biệt là mật khẩu, phải được mã hóa trước khi lưu trữ trong cơ sở dữ liệu.
- Hệ thống cần kiểm soát quyền truy cập thông tin dựa trên vai trò người dùng, đảm bảo mỗi đối tượng chỉ có thể truy cập các chức năng và dữ liệu được cho phép.

- Khả năng sử dụng và giao diện (Usability UI)

- Giao diện hệ thống cần được thiết kế thân thiện, trực quan và dễ sử dụng đối với người dùng ở nhiều độ tuổi và trình độ khác nhau.
- Hệ thống phải hỗ trợ thiết kế responsive, đảm bảo hiển thị và sử dụng tốt trên các thiết bị di động và máy tính bảng.

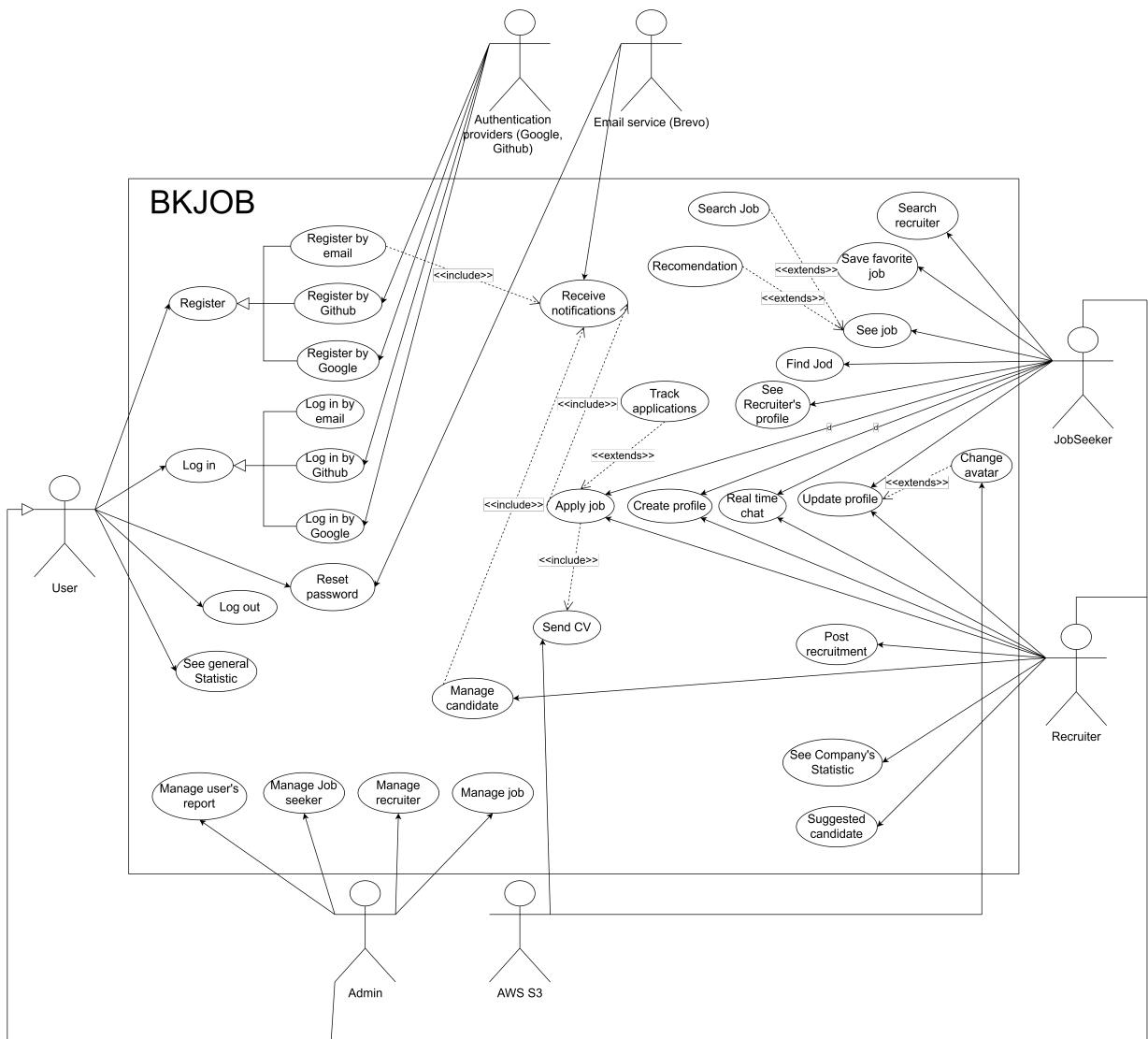
- Trải nghiệm người dùng (User Experience)

- Thời gian thực hiện thao tác đăng tin tuyển dụng của nhà tuyển dụng không vượt quá 10 phút trong điều kiện sử dụng bình thường.
- Người dùng mới có thể nắm bắt và sử dụng các chức năng cơ bản của hệ thống trong vòng không quá 5 phút.
- Hệ thống hỗ trợ đăng nhập thông qua nhiều nền tảng khác nhau nhằm tăng sự thuận tiện cho người dùng.

- Khả năng bảo trì và mở rộng (Maintainability)

- Kiến trúc hệ thống cần được thiết kế rõ ràng, dễ bảo trì và mở rộng trong tương lai.
- Mã nguồn phía backend cần tuân thủ các nguyên tắc thiết kế SOLID để đảm bảo tính rõ ràng, tái sử dụng và dễ dàng nâng cấp.

1.4 Use-case diagram



Hình 1: Use-case diagram của hệ thống BKJOB



1.5 Use-case scenarios

1.5.1 Đăng ký

Use-case ID	uc-01
Use-case name	Dăng ký tài khoản
Actor	Jobseeker, Recruiter
Description	Người dùng tạo tài khoản mới thông qua email hoặc đăng ký nhanh bằng Google/GitHub.
Trigger	Người dùng chọn “Sign up”.
Precondition	Người dùng chưa có tài khoản trong hệ thống.
Postcondition	Tài khoản được tạo thành công và thông tin cá nhân được hoàn thiện.
Normal flow	<ol style="list-style-type: none">Người dùng chọn “Sign up”.Nhập thông tin: email, số điện thoại, mật khẩu, nhập lại mật khẩu.Người dùng nhấn “Sign up”.Hệ thống gửi mã xác thực (OTP) đến email.Người dùng nhập mã xác thực.Hệ thống xác thực thành công và chuyển đến form “Complete Information”.Người dùng nhập các thông tin hồ sơ cần thiết.Hệ thống lưu thông tin và hoàn tất đăng ký.
Alternative flows	A1: Đăng ký bằng Google/GitHub A1.1 Người dùng chọn “Continue with Google/GitHub”. A1.2 Hệ thống nhận thông tin xác thực từ Google/GitHub. A1.3 Nếu lần đầu sử dụng, hệ thống hiển thị form “Complete Information”. A1.4 Người dùng nhập thông tin bổ sung và hệ thống lưu lại.
Exception flows	E1: Email đã tồn tại – Hệ thống hiển thị thông báo "User already existed". E2: Mã xác thực không hợp lệ hoặc hết hạn – Hệ thống thông báo lỗi và cho phép người dùng nhập lại hoặc gửi mã mới.



1.5.2 Đăng nhập

Use-case ID	uc-02
Use-case name	Dăng nhập
Actor	Tất cả người dùng (Jobseeker, Recruiter, Admin)
Description	Người dùng đăng nhập vào hệ thống BKJOB bằng email và mật khẩu hoặc đăng nhập nhanh qua Google/GitHub.
Trigger	Người dùng chọn “Log in” trên giao diện.
Precondition	Người dùng chưa đăng nhập.
Postcondition	Người dùng được xác thực thành công và chuyển vào trang phù hợp với vai trò.
Normal flow	<ol style="list-style-type: none">Người dùng mở giao diện đăng nhập.Nhập email và mật khẩu.Nhấn “Log in”.Hệ thống xác thực thông tin đăng nhập.Đăng nhập thành công và chuyển hướng theo vai trò (Jobseeker/Recruiter/Admin).
Alternative flows	<p>A1: Đăng nhập bằng Google/GitHub</p> <p>A1.1 Người dùng chọn “Continue with Google/GitHub”.</p> <p>A1.2 Hệ thống nhận thông tin từ dịch vụ xác thực.</p> <p>A1.3 Nếu người dùng đã có tài khoản → chuyển hướng vào hệ thống.</p> <p>A1.4 Nếu người dùng chưa có tài khoản → hệ thống chuyển đến form “Complete Information”.</p> <p>A1.5 Người dùng điền thông tin bổ sung và hoàn tất đăng nhập.</p> <p>A2: Người dùng chọn “Quên mật khẩu”</p> <p>A2.1 Người dùng nhập email đã đăng ký.</p> <p>A2.2 Nhấn "Forgot password"</p> <p>A2.3 Hệ thống gửi link đặt lại mật khẩu.</p> <p>A2.4 Người dùng đặt lại mật khẩu và quay lại màn hình đăng nhập.</p>
Exception flows	<p>E1: Thiếu email hoặc mật khẩu</p> <ul style="list-style-type: none">Hệ thống yêu cầu nhập đầy đủ thông tin. <p>E2: Email không đúng</p> <ul style="list-style-type: none">Hệ thống thông báo “User not found”. <p>E3: Email đúng và mật khẩu không đúng</p> <ul style="list-style-type: none">Hệ thống thông báo “Password incorrect”.



1.5.3 Quản lý thông tin Dashboard (Recruiter)

Use-case ID	uc-03
Use-case name	Quản lý thông tin Dashboard (Recruiter)
Actor	Recruiter
Description	Recruiter xem bảng điều khiển (dashboard) để quản lý các số liệu tổng quan như tổng số Job đã đăng, tổng số Application, tổng số User liên quan, cùng các biểu đồ thống kê theo tuần hoặc tháng.
Trigger	Recruiter chọn mục “Dashboard” từ giao diện chính.
Precondition	Recruiter đã đăng nhập vào hệ thống và có quyền truy cập dashboard của công ty.
Postcondition	Dashboard hiển thị đầy đủ dữ liệu tổng quan và biểu đồ thống kê phù hợp với bộ lọc được chọn.
Normal flow	<ol style="list-style-type: none">Recruiter nhấp vào mục “Dashboard”.Hệ thống tải dữ liệu tổng quan, bao gồm: Total Applications, Total Jobs, Total Companies (nếu áp dụng), Total Users liên quan.Hệ thống hiển thị các biểu đồ:<ul style="list-style-type: none">– Job Postings Trend.– Application Status Distribution.Recruiter lựa chọn bộ lọc thời gian (Weekly / Monthly).Hệ thống cập nhật dashboard theo bộ lọc được chọn.
Alternative flows	<p>A1: Chọn phạm vi dữ liệu của công ty A1.1 Recruiter chọn “My Company”. A1.2 Dashboard chỉ hiển thị dữ liệu thuộc công ty của Recruiter.</p> <p>A2: Recruiter chọn xem số liệu toàn hệ thống A2.1 Recruiter chọn “General” . A2.2 Hệ thống hiển thị tổng quan toàn bộ dữ liệu bên ngoài công ty.</p>
Exception flows	None



1.5.4 Đăng tin tuyển dụng

Use-case ID	uc-04
Use-case name	Dăng tin tuyển dụng
Actor	Recruiter
Description	Nhà tuyển dụng đăng bài tuyển dụng để tìm ứng viên phù hợp.
Trigger	Người dùng chọn chức năng “Post Job” trên giao diện hệ thống.
Precondition	Người dùng đã đăng nhập và thuộc vai trò Recruiter.
Postcondition	Tin tuyển dụng được lưu và hiển thị trên hệ thống.
Normal flow	<ol style="list-style-type: none">Recruiter chọn chức năng “Post Job”.Nhập thông tin job (Job Title, Job Description, Requirements, Skills Required, Salary Range, Experience Level, Work Location, Deadline, Industry, Education Required...).Nhấn “Post”.Hệ thống lưu tin thành công và công bố trên trang tuyển dụng.
Alternative flows	<p>A1: Thiếu thông tin bắt buộc</p> <ul style="list-style-type: none">Hệ thống yêu cầu nhập đầy đủ các trường bắt buộc (Job Title, Job Description, Job Type, Work Location, Industry, Province...). <p>A2: Min Salary > Max Salary</p> <ul style="list-style-type: none">Hệ thống hiển thị lỗi: “Max salary must be greater than or equal to Min salary.” <p>A3: Application deadline không hợp lệ</p> <ul style="list-style-type: none">Người dùng nhập deadline nhỏ hơn ngày hiện tại.Hệ thống hiển thị lỗi: “Application deadline must be today or in the future.”Hệ thống yêu cầu chọn lại ngày hợp lệ.
Exception flows	Mất kết nối hoặc lỗi hệ thống → tin không được tạo.

1.5.5 Xem danh sách Jobseeker được gợi ý

Use-case ID	uc-05
Use-case name	Xem danh sách Jobseeker được gợi ý
Actor	Recruiter
Description	Recruiter xem danh sách các Jobseeker được hệ thống đề xuất dựa trên mức độ phù hợp với bài tuyển dụng.
Trigger	Recruiter nhấn nút “Suggested CV” trên giao diện hệ thống.
Precondition	Recruiter đã đăng nhập và có ít nhất một bài tuyển dụng hợp lệ.
Postcondition	Danh sách Jobseeker được gợi ý hiển thị thành công.
Normal flow	<ol style="list-style-type: none">Recruiter nhấn nút “Suggested CV”.Hệ thống tìm Jobseeker phù hợp theo kỹ năng, kinh nghiệm, vị trí...Hệ thống hiển thị danh sách các Jobseeker được đề xuất.
Alternative flows	<p>A1: Không có ứng viên phù hợp</p> <ul style="list-style-type: none">Hệ thống thông báo: “0 candidates found” và không hiện danh sách Jobseeker
Exception flows	None.



1.5.6 Xem chi tiết hồ sơ Jobseeker

Use-case ID	uc-06
Use-case name	Xem chi tiết hồ sơ Jobseeker
Actor	Recruiter
Description	Recruiter xem thông tin chi tiết của một Jobseeker được đề xuất.
Trigger	Recruiter nhấn nút “View full profile” trên thẻ Jobseeker trong danh sách gợi ý.
Precondition	Recruiter đã xem danh sách Jobseeker được gợi ý.
Postcondition	Hồ sơ Jobseeker được hiển thị đầy đủ.
Normal flow	1. Recruiter chọn nút “View Full Profile” trên thẻ Jobseeker. 2. Hệ thống điều hướng đến trang hồ sơ Jobseeker. 3. Hệ thống hiển thị toàn bộ thông tin của Jobseeker.
Alternative flows	None.
Exception flows	E1: Lỗi khi tải danh sách Jobseeker – Hệ thống không hiển thị danh sách.

1.5.7 Quản lý danh sách Jobseeker đã ứng tuyển

Use-case ID	uc-07
Use-case name	Quản lý danh sách Jobseeker đã ứng tuyển
Actor	Recruiter
Description	Recruiter xem danh sách các Jobseeker đã ứng tuyển vào Job đã đăng và thực hiện thao tác <i>Accept</i> hoặc <i>Reject</i> ứng viên.
Trigger	Người dùng chọn nút “Application” trong trang quản lý Job.
Precondition	Recruiter đã đăng nhập và là chủ bài tuyển dụng.
Postcondition	Trạng thái ứng tuyển của Jobseeker được cập nhật (Accepted / Rejected).
Normal flow	1. Recruiter chọn nút “Applications”. 2. Hệ thống hiển thị danh sách Jobseeker đã ứng tuyển (kèm trạng thái hiện tại : Pending, Accepted, Rejected). 3. Recruiter chọn một ứng viên để xem chi tiết hồ sơ (CV, kỹ năng, kinh nghiệm...). 4. Recruiter chọn “Accept” hoặc “Reject”. 5. Hệ thống cập nhật trạng thái ứng tuyển của ứng viên.
Alternative flows	A1: Không có ứng viên – Hệ thống hiển thị: “No applications found.”
Exception flows	None



1.5.8 Quản lý tin tuyển dụng (My Jobs)

Use-case ID	uc-08
Use-case name	Quản lý tin tuyển dụng (My Jobs)
Actor	Recruiter
Description	Recruiter xem và quản lý danh sách các tin tuyển dụng mà họ đã đăng, bao gồm lọc, tìm kiếm, truy cập chi tiết từng tin.
Trigger	Recruiter chọn mục “My Jobs” trong ứng dụng.
Precondition	Recruiter đã đăng nhập.
Postcondition	Danh sách tin tuyển dụng được hiển thị theo điều kiện tìm kiếm đã chọn.
Normal flow	<ol style="list-style-type: none">1. Recruiter mở trang My Jobs.2. Hệ thống hiển thị danh sách tin tuyển dụng hiện có.3. Recruiter sử dụng tìm kiếm hoặc bộ lọc cơ bản để thu hẹp danh sách.4. Hệ thống cập nhật danh sách theo điều kiện tìm kiếm.5. Recruiter chuyển trang để xem thêm các tin khác.6. Recruiter chọn một tin để xem chi tiết.
Alternative flows	A1. Danh sách rỗng → hệ thống hiển thị thông báo “No jobs found.”
Exception flows	E1. Lỗi khi tải danh sách → hệ thống thông báo lỗi và không hiển thị dữ liệu.



1.5.9 Xem và chỉnh sửa hồ sơ cá nhân

Use-case ID	uc-09
Use-case name	Xem và chỉnh sửa hồ sơ cá nhân
Actor	Jobseeker, Recruiter
Description	Người dùng xem và chỉnh sửa toàn bộ thông tin cá nhân của mình trên hệ thống (họ tên, giới tính, ảnh đại diện, kỹ năng, kinh nghiệm, CV...).
Trigger	Người dùng chọn chức năng “Profile” hoặc "See my profile".
Precondition	Người dùng đã đăng nhập vào hệ thống.
Postcondition	Thông tin hồ sơ được cập nhật thành công.
Normal flow	<ol style="list-style-type: none">Người dùng mở trang “Profile”.Hệ thống hiển thị toàn bộ thông tin hồ sơ hiện tại.Người dùng chọn chức năng “Edit Profile”.Người dùng chỉnh sửa các trường thông tin được phép (họ tên, giới tính, mô tả...).Người dùng có thể cập nhật các mục bổ sung: ảnh đại diện, kỹ năng, kinh nghiệm, CV...Hệ thống hiển thị preview đối với file được upload (ảnh đại diện, CV).Người dùng nhấn “Update Jobseeker”.Hệ thống kiểm tra dữ liệu và lưu lại toàn bộ thay đổi.
Alternative flows	None.
Exception flows	<p>E1: Thiếu thông tin bắt buộc – Hệ thống hiển thị thông báo lỗi.</p> <p>E2: Sai định dạng email/phone – Hệ thống yêu cầu sửa lại.</p> <p>E3: Upload file không hợp lệ – File CV hoặc ảnh sai định dạng → hệ thống từ chối.</p>

1.5.10 Tìm kiếm công việc

Use-case ID	uc-10
Use-case name	Tìm kiếm công việc
Actor	Jobseeker
Description	Jobseeker tìm kiếm các công việc thông qua từ khoá hoặc bộ lọc như vị trí, lương, kinh nghiệm, kỹ năng.
Trigger	Người dùng truy cập trang “Jobs”.
Precondition	None.
Postcondition	Danh sách Job phù hợp được hiển thị.
Normal flow	<ol style="list-style-type: none">Người dùng mở trang “Jobs”.Nhập từ khoá hoặc chọn bộ lọc.Hệ thống tìm kiếm và hiển thị danh sách Job phù hợp.
Alternative flows	<p>A1: Không có kết quả – Hệ thống hiển thị: “No jobs found.”</p>
Exception flows	None.



1.5.11 Lưu công việc

Use-case ID	uc-11
Use-case name	Lưu công việc
Actor	Jobseeker
Description	Jobseeker lưu lại một công việc để xem sau.
Trigger	Người dùng nhấn nút “Save” trên thẻ Job hoặc trong trang Job Detail.
Precondition	Jobseeker đã đăng nhập.
Postcondition	Job được lưu vào danh sách Saved Jobs của Jobseeker.
Normal flow	1. Người dùng xem danh sách Job hoặc trang chi tiết. 2. Nhấn nút “Save”. 3. Hệ thống lưu Job vào Saved List.
Alternative flows	A1: Job đã được lưu trước đó – Hệ thống thông báo: “This job is already saved.”
Exception flows	E1: Chưa đăng nhập – Hệ thống yêu cầu đăng nhập để lưu Job.

1.5.12 Xem chi tiết công việc

Use-case ID	uc-12
Use-case name	Xem chi tiết công việc
Actor	Jobseeker
Description	Jobseeker xem thông tin chi tiết về một công việc, bao gồm mô tả, yêu cầu, quyền lợi, công ty, và nút Apply hoặc Save.
Trigger	Người dùng nhấn vào một thẻ Job trong trang Jobs.
Precondition	None.
Postcondition	Thông tin chi tiết Job hiển thị thành công.
Normal flow	1. Người dùng chọn một Job trong danh sách. 2. Hệ thống tải thông tin và hiển thị trang Job Detail.
Alternative flows	None.
Exception flows	None.



1.5.13 Ứng tuyển công việc

Use-case ID	uc-13
Use-case name	Ứng tuyển công việc
Actor	Jobseeker
Description	Jobseeker nộp đơn ứng tuyển vào một công việc.
Trigger	Jobseeker chọn nút “Apply Job” tại trang Job Detail.
Precondition	Jobseeker đã đăng nhập.
Postcondition	Hệ thống ghi nhận một Application mới với trạng thái ban đầu <i>Pending</i> .
Normal flow	<ol style="list-style-type: none">1. Jobseeker truy cập trang chi tiết công việc.2. Jobseeker chọn nút "Apply Job"3. Jobseeker nhập CV với định dạng hợp lệ (pdf,doc,docx) và nhập cover letter (nếu có).4. Jobseeker nhấn “Apply”.5. Hệ thống ghi nhận thông tin ứng tuyển.6. Hệ thống hiển thị thông báo ứng tuyển thành công.
Alternative flows	A1: Jobseeker đã apply trước đó – Hệ thống hiển thị: “Applied” tại trang chi tiết công việc
Exception flows	E1: Thiếu CV – Hệ thống yêu cầu hoàn thiện CV trước khi ứng tuyển.

1.5.14 Xem danh sách các công việc đã lưu

Use-case ID	uc-14
Use-case name	Xem danh sách các công việc đã lưu
Actor	Jobseeker
Description	Jobseeker xem danh sách các công việc đã lưu để theo dõi hoặc apply sau.
Trigger	Jobseeker chọn mục “Saved Jobs” từ menu "Jobs" hoặc mục "My saved jobs" từ profile cá nhân.
Precondition	Jobseeker đã đăng nhập.
Postcondition	Danh sách Saved Jobs được hiển thị.
Normal flow	<ol style="list-style-type: none">1. Jobseeker mở trang “Saved Jobs”.2. Hệ thống truy xuất danh sách công việc đã lưu.
Alternative flows	A1: Chưa lưu job nào – Hệ thống hiển thị: “You have saved 0 jobs”
Exception flows	None



1.5.15 Xem danh sách công việc được đề xuất

Use-case ID	uc-15
Use-case name	Xem danh sách công việc được đề xuất
Actor	Jobseeker
Description	Jobseeker xem danh sách các công việc được hệ thống gợi ý dựa trên hồ sơ và hoạt động của người dùng.
Trigger	Jobseeker chọn mục “Recommended Jobs”.
Precondition	Jobseeker đã đăng nhập và có hồ sơ cá nhân.
Postcondition	Các recommended jobs được hiển thị.
Normal flow	<ol style="list-style-type: none">1. Jobseeker mở trang “Recommended Jobs”.2. Hệ thống phân tích hồ sơ và hoạt động người dùng.3. Hệ thống trả về danh sách job phù hợp.4. Jobseeker có thể mở xem chi tiết, ứng tuyển hoặc lưu job.
Alternative flows	A1: Không tìm thấy job phù hợp – Hệ thống hiển thị: “No recommended jobs available.”
Exception flows	None

1.5.16 Xem danh sách các công ty

Use-case ID	uc-16
Use-case name	Xem danh sách các công ty
Actor	Jobseeker
Description	Jobseeker xem danh sách các công ty đang hoạt động trên hệ thống.
Trigger	Người dùng truy cập trang “Companies”.
Precondition	Jobseeker đã đăng nhập.
Postcondition	Danh sách công ty được hiển thị.
Normal flow	<ol style="list-style-type: none">1. Jobseeker mở trang Companies.2. Hệ thống hiển thị danh sách công ty (avatar, background, mô tả...).3. Jobseeker có thể chọn 1 công ty để xem chi tiết.
Alternative flows	None
Exception flows	E1: Không có công ty nào – Hệ thống hiển thị: “No companies found.”



1.5.17 Tìm kiếm công ty

Use-case ID	uc-17
Use-case name	Tìm kiếm công ty
Actor	Jobseeker
Description	Jobseeker tìm kiếm công ty theo tên, email.
Trigger	Jobseeker nhập từ khóa vào ô tìm kiếm tại trang Companies.
Precondition	Jobseeker đang ở trang Companies và có dữ liệu công ty trong hệ thống.
Postcondition	Danh sách công ty được lọc và hiển thị theo từ khóa.
Normal flow	1. Jobseeker nhập từ khóa vào ô Search. 2. Hệ thống tìm kiếm các công ty phù hợp với keyword. 3. Hệ thống hiển thị danh sách công ty trùng khớp.
Alternative flows	None
Exception flows	E1: Không có kết quả phù hợp – Hệ thống không hiển thị công ty nào.

1.5.18 Xem danh sách công ty theo thứ hạng có người ứng tuyển từ nhiều đến ít

Use-case ID	uc-18
Use-case name	Xem danh sách công ty theo thứ hạng có người ứng tuyển từ nhiều đến ít
Actor	Jobseeker
Description	Jobseeker xem bảng xếp hạng các công ty dựa trên số lượng ứng tuyển mà công ty nhận được.
Trigger	Jobseeker nhấn nút “Ranking companies by number of applications”.
Precondition	Jobseeker đang ở trang Companies và hệ thống có dữ liệu ứng tuyển.
Postcondition	Danh sách công ty được hiển thị theo thứ hạng dựa trên số lượng ứng tuyển.
Normal flow	1. Jobseeker nhấn nút “Ranking companies by number of applications” để bật chế độ Ranking . 2. Hệ thống truy vấn và tính toán thứ hạng công ty theo số lượng ứng tuyển. 3. Hệ thống hiển thị danh sách công ty kèm số lượng ứng tuyển và vị trí xếp hạng.
Alternative flows	A1: Jobseeker tắt chế độ Ranking - Jobseeker nhấn nút “Ranking companies by number of applications” 1 lần nữa để tắt. – Hệ thống trở về danh sách công ty hiển thị theo chế độ thông thường.
Exception flows	None



1.5.19 Xem chi tiết công ty

Use-case ID	uc-19
Use-case name	Xem chi tiết công ty
Actor	Jobseeker
Description	Jobseeker xem thông tin chi tiết của một công ty, bao gồm profile, mô tả công ty và danh sách job đang tuyển.
Trigger	Người dùng click vào một Company Card trong danh sách công ty.
Precondition	Jobseeker đã truy cập vào trang Companies và công ty đó đang hoạt động.
Postcondition	Chi tiết công ty được hiển thị; Jobseeker có thể xem các job của công ty.
Normal flow	<ol style="list-style-type: none">1. Jobseeker click vào một Company Card.2. Hệ thống điều hướng đến trang thông tin chi tiết của công ty.3. Hệ thống hiển thị profile công ty, mô tả, thông tin liên hệ và danh sách job đang tuyển.
Alternative flows	None
Exception flows	E1: Công ty không tồn tại hoặc bị khóa – Hệ thống hiển thị lỗi “Company not found”.

1.5.20 Chatting

Use-case ID	uc-20
Use-case name	Chatting
Actor	Jobseeker, Recruiter
Description	Jobseeker và Recruiter có thể trao đổi tin nhắn trực tiếp với nhau để thảo luận về công việc, phỏng vấn hoặc các thông tin liên quan.
Trigger	Jobseeker hoặc Recruiter chọn chức năng Chat tại trang hồ sơ người dùng hoặc nhập tên người dùng vào ô tìm kiếm của khung chat.
Precondition	Người dùng đã đăng nhập.
Postcondition	Tin nhắn được gửi thành công và lưu lại trong hệ thống.
Normal flow	<ol style="list-style-type: none">1. Actor mở cửa sổ chat.2. Actor nhập nội dung tin nhắn.3. Actor nhấn nút Enter.4. Hệ thống gửi tin nhắn đến người nhận.5. Hệ thống hiển thị tin nhắn trong cuộc hội thoại.
Alternative flows	A1: Người nhận không online – Tin nhắn được lưu và hiển thị khi người nhận đăng nhập.
Exception flows	E1: Gửi tin nhắn thất bại – Hệ thống hiển thị thông báo lỗi.



1.5.21 Nhận thông báo theo dõi về công việc đã ứng tuyển

Use-case ID	uc-21
Use-case name	Nhận thông báo theo dõi về công việc đã ứng tuyển
Actor	Jobseeker
Description	Jobseeker nhận thông báo về trạng thái hồ sơ ứng tuyển như pending, được chấp nhận hoặc bị từ chối.
Trigger	Recruiter cập nhật trạng thái hồ sơ ứng tuyển của Jobseeker.
Precondition	Jobseeker đã ứng tuyển vào ít nhất một công việc.
Postcondition	Thông báo được gửi và hiển thị cho Jobseeker.
Normal flow	<ol style="list-style-type: none">1. Jobseeker ứng tuyển công việc.2. Hệ thống thông báo Jobseeker có thể theo dõi trạng thái ứng tuyển khi click vào thông báo đó.
Alternative flows	None
Exception flows	None

1.5.22 Báo cáo công ty

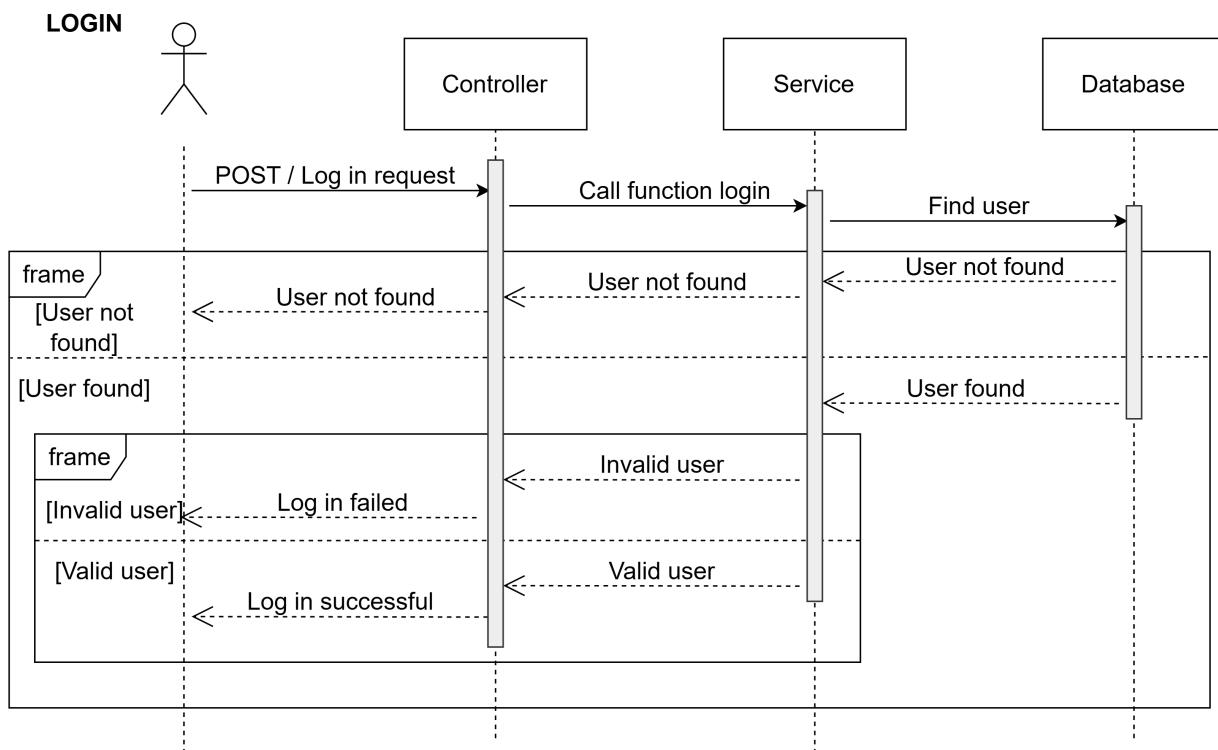
Use-case ID	uc-22
Use-case name	Báo cáo công ty
Actor	Jobseeker
Description	Jobseeker gửi báo cáo về công ty có hành vi gian lận, thông tin sai lệch hoặc vi phạm chính sách của hệ thống.
Trigger	Jobseeker nhấn nút "Report to us" tại trang chi tiết công ty.
Precondition	Jobseeker đã đăng nhập và đang xem thông tin công ty.
Postcondition	Báo cáo được lưu và gửi đến Admin để xử lý.
Normal flow	<ol style="list-style-type: none">1. Jobseeker chọn chức năng Report Company.2. Jobseeker nhập nội dung báo cáo.3. Jobseeker gửi báo cáo.4. Hệ thống lưu báo cáo.5. Hệ thống thông báo gửi báo cáo thành công.
Alternative flows	A1: Jobseeker hủy báo cáo – Hệ thống đóng form báo cáo và không lưu dữ liệu.
Exception flows	None

2 System modeling

2.1 Activity diagrams

2.2 Sequence diagrams

2.2.1 Luồng đăng nhập (Login)



Hình 2: Sequence diagram - Login

Hình trên mô tả luồng xử lý chức năng đăng nhập của hệ thống theo kiến trúc ba tầng Controller – Service – Database. Trình tự xử lý được mô tả như sau:

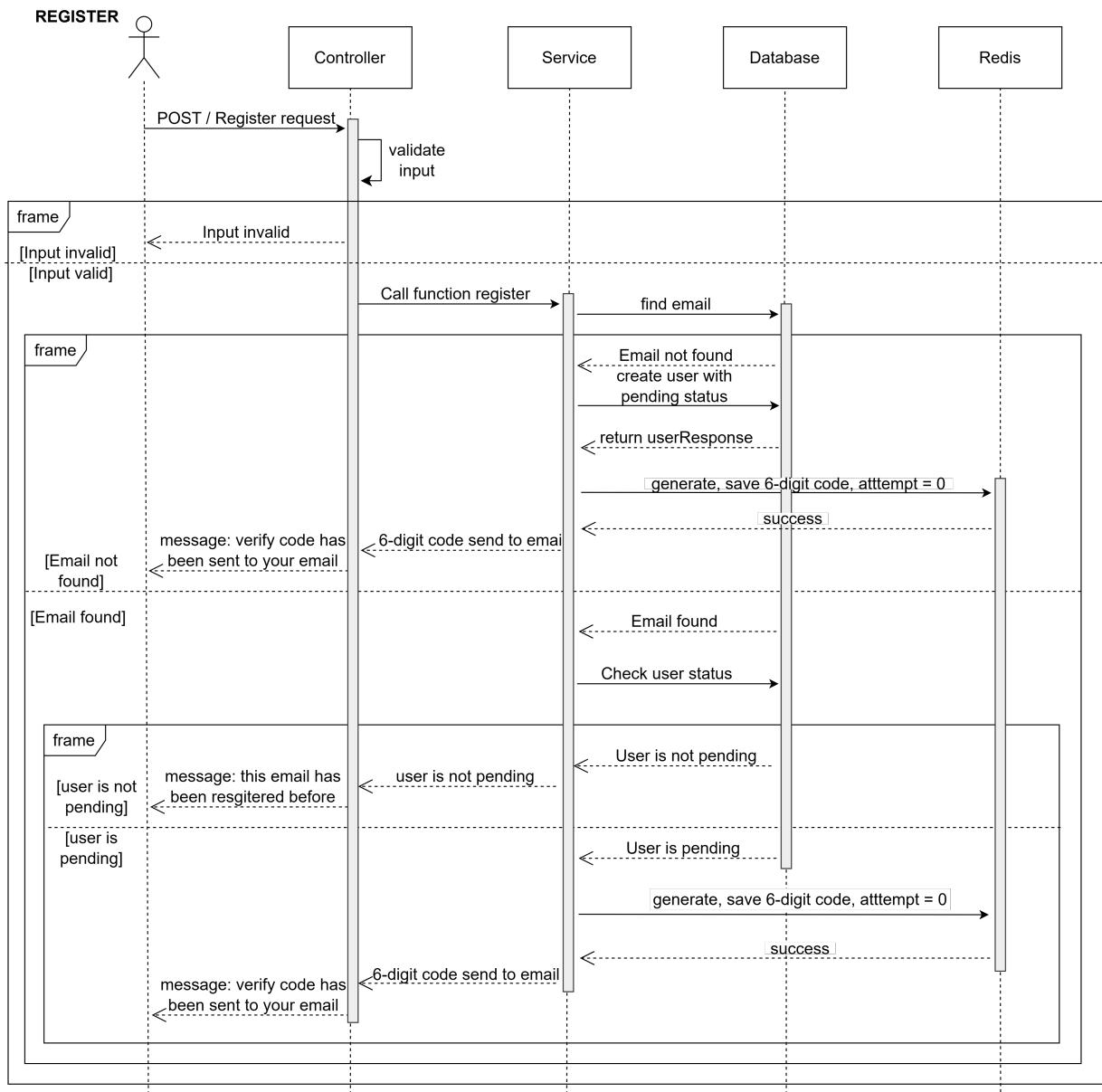
1. Người dùng gửi yêu cầu đăng nhập (POST /login) đến hệ thống, bao gồm thông tin đăng nhập như email/username và mật khẩu.
2. Controller tiếp nhận yêu cầu đăng nhập và gọi hàm xử lý đăng nhập tại tầng Service.
3. Service gửi yêu cầu truy vấn đến Database để tìm kiếm người dùng tương ứng.
4. Trường hợp người dùng không tồn tại:
 - Database trả về kết quả không tìm thấy người dùng.
 - Service trả thông báo *User not found* về cho Controller.
 - Controller phản hồi thông báo đăng nhập thất bại cho người dùng.
5. Trường hợp người dùng tồn tại:
 - Database trả về thông tin người dùng cho Service.
 - Service kiểm tra tính hợp lệ của thông tin đăng nhập.
6. Trường hợp thông tin đăng nhập không hợp lệ:

- Service xác định người dùng không hợp lệ (*Invalid user*).
- Controller phản hồi thông báo *Login failed* cho người dùng.

7. Trường hợp thông tin đăng nhập hợp lệ:

- Service xác nhận người dùng hợp lệ (*Valid user*).
- Controller phản hồi thông báo *Login successful* cho người dùng.

2.2.2 Luồng đăng ký (Register)



Hình 3: Sequence diagram - Register

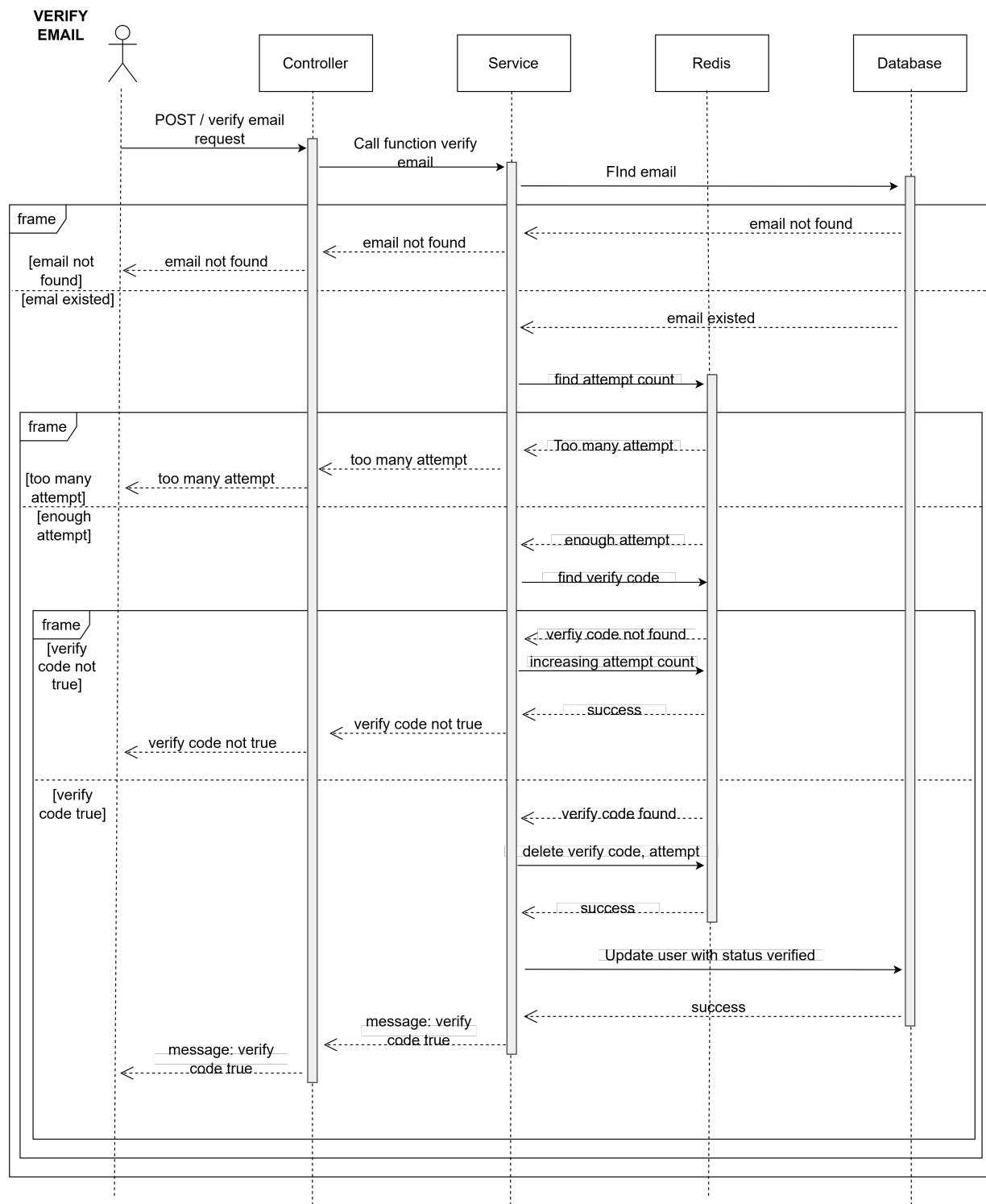
Hình trên mô tả luồng xử lý chức năng đăng ký tài khoản của hệ thống theo kiến trúc Controller – Service – Database – Redis. Trình tự xử lý được mô tả như sau:

1. Người dùng gửi yêu cầu đăng ký (POST /register) đến hệ thống với các thông tin cần thiết.



2. Controller tiếp nhận yêu cầu và kiểm tra tính hợp lệ của dữ liệu đầu vào.
3. Trường hợp dữ liệu đầu vào không hợp lệ:
 - Controller trả về thông báo lỗi *Input invalid* cho người dùng.
4. Trường hợp dữ liệu đầu vào hợp lệ:
 - Controller gọi hàm xử lý đăng ký tại tầng Service.
 - Service gửi yêu cầu đến Database để kiểm tra sự tồn tại của email.
5. Trường hợp email chưa tồn tại:
 - Database trả về kết quả *Email not found*.
 - Service tạo mới người dùng với trạng thái *pending*.
 - Service tạo mã xác thực 6 chữ số và lưu vào Redis với số lần thử bằng 0.
 - Controller thông báo người dùng kiểm tra email để xác thực tài khoản.
6. Trường hợp email đã tồn tại:
 - Service kiểm tra trạng thái người dùng.
7. Trường hợp người dùng không ở trạng thái pending:
 - Controller trả về thông báo *This email has been registered before*.
8. Trường hợp người dùng đang ở trạng thái pending:
 - Service tạo lại mã xác thực 6 chữ số và lưu vào Redis.
 - Controller thông báo người dùng kiểm tra email để xác thực tài khoản.

2.2.3 Luồng xác thực email (Verify Email)



Hình 4: Sequence diagram - Verify Email

Hình trên mô tả luồng xử lý chức năng xác thực email của hệ thống theo kiến trúc Controller – Service – Redis – Database. Trình tự xử lý được mô tả như sau:

1. Người dùng gửi yêu cầu xác thực email (POST /verify-email) đến hệ thống, kèm theo email và mã xác thực.



2. Controller tiếp nhận yêu cầu và gọi hàm xác thực email tại tầng Service.

3. Service gửi yêu cầu đến Database để kiểm tra sự tồn tại của email.

4. Trường hợp email không tồn tại trong hệ thống:

- Database trả về kết quả *email not found*.
- Service trả kết quả xác thực thất bại về cho Controller.
- Controller phản hồi thông báo email không tồn tại cho người dùng.

5. Trường hợp email tồn tại trong hệ thống:

- Service gửi yêu cầu đến Redis để kiểm tra số lần thử xác thực.

6. Trường hợp số lần thử vượt quá giới hạn cho phép:

- Redis trả về kết quả *too many attempt*.
- Service trả kết quả xác thực thất bại về cho Controller.
- Controller phản hồi thông báo vượt quá số lần xác thực cho người dùng.

7. Trường hợp số lần thử vẫn trong giới hạn cho phép:

- Service gửi yêu cầu đến Redis để lấy mã xác thực.

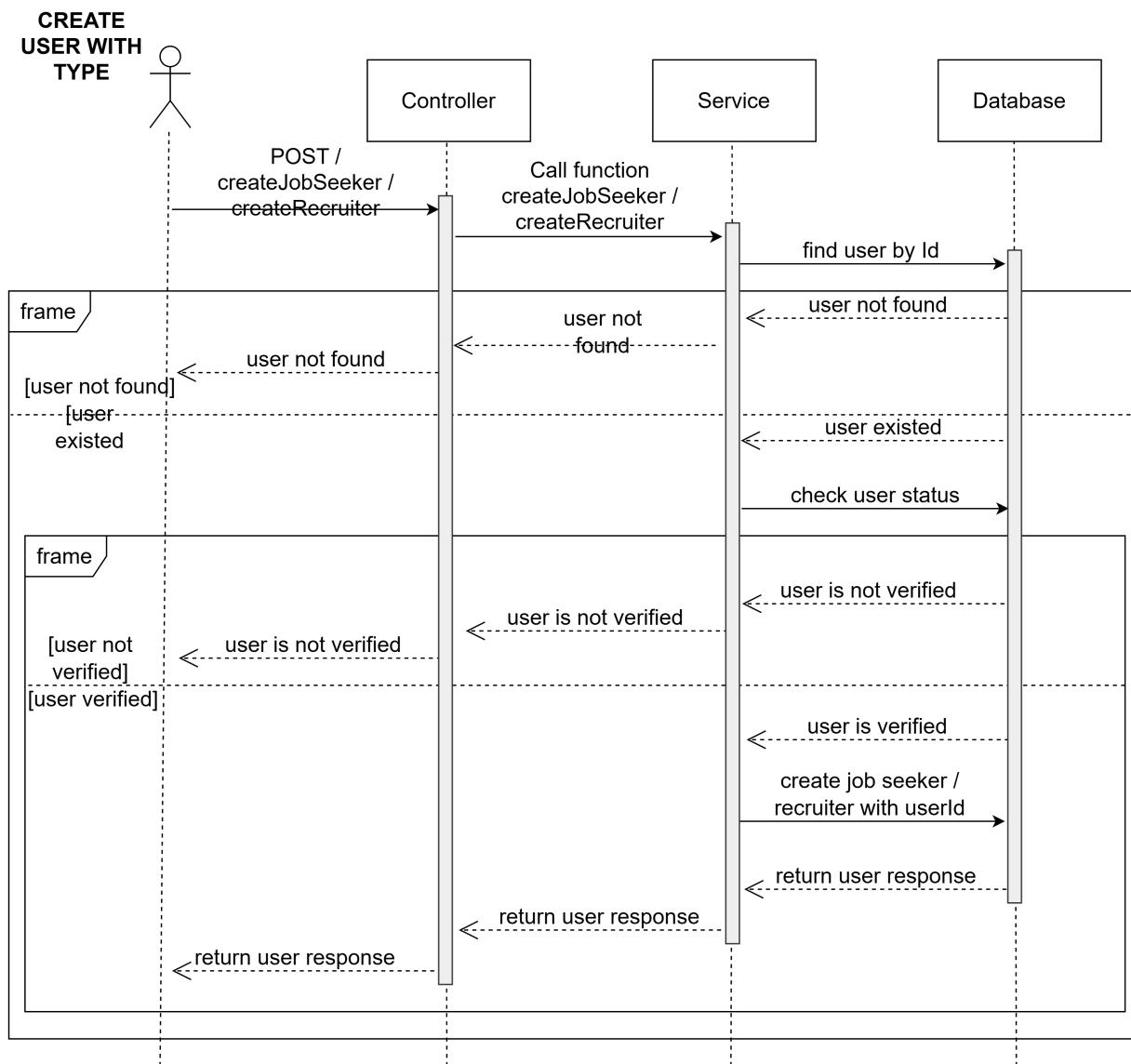
8. Trường hợp mã xác thực không đúng:

- Redis không tìm thấy hoặc trả về mã xác thực không hợp lệ.
- Service tăng số lần thử xác thực trong Redis.
- Service trả kết quả xác thực thất bại về cho Controller.
- Controller phản hồi thông báo mã xác thực không đúng cho người dùng.

9. Trường hợp mã xác thực đúng:

- Service xóa mã xác thực và số lần thử tương ứng trong Redis.
- Service cập nhật trạng thái người dùng thành *verified* trong Database.
- Database trả kết quả cập nhật thành công về cho Service.
- Service trả kết quả xác thực thành công về cho Controller.
- Controller phản hồi thông báo xác thực email thành công cho người dùng.

2.2.4 Luồng tạo người dùng theo vai trò (Create User With Type)



Hình 5: Sequence diagram - Create User With Type

Hình trên mô tả luồng xử lý chức năng tạo người dùng theo vai trò (JobSeeker hoặc Recruiter) của hệ thống theo kiến trúc Controller – Service – Database. Trình tự xử lý được mô tả như sau:

- Người dùng gửi yêu cầu tạo người dùng theo vai trò (POST /createJobSeeker hoặc POST /createRecruiter) đến hệ thống.
- Controller tiếp nhận yêu cầu và gọi hàm xử lý tương ứng tại tầng Service.
- Service gửi yêu cầu đến Database để tìm kiếm người dùng theo *userId*.
- Trường hợp người dùng không tồn tại:
 - Database trả về kết quả *user not found*.
 - Service trả kết quả thất bại về cho Controller.
 - Controller phản hồi thông báo người dùng không tồn tại cho phía người dùng.

5. Trường hợp người dùng tồn tại:

- Service tiến hành kiểm tra trạng thái xác thực của người dùng.

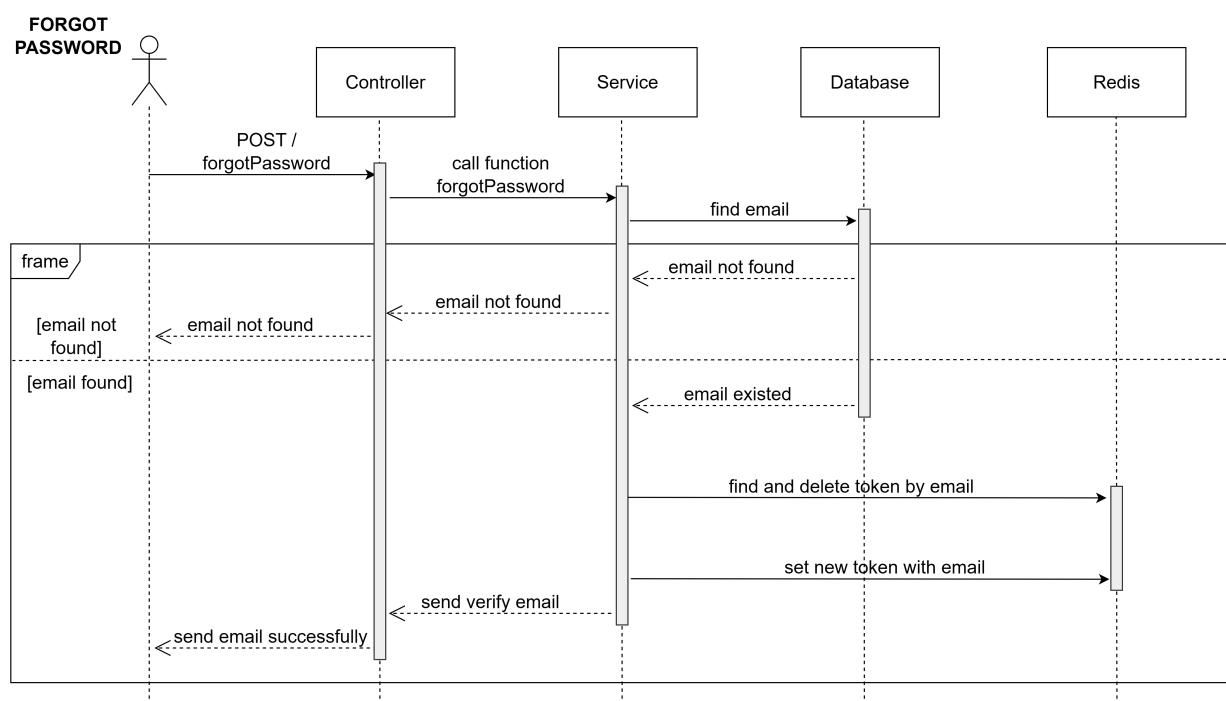
6. Trường hợp người dùng chưa được xác thực:

- Database trả về trạng thái *user is not verified*.
- Service trả kết quả thất bại về cho Controller.
- Controller phản hồi thông báo người dùng chưa được xác thực.

7. Trường hợp người dùng đã được xác thực:

- Service tạo mới thông tin JobSeeker hoặc Recruiter tương ứng với *userId*.
- Database lưu thông tin người dùng theo vai trò và trả về kết quả thành công.
- Service trả thông tin người dùng đã tạo về cho Controller.
- Controller phản hồi kết quả tạo người dùng thành công cho phía người dùng.

2.2.5 Luồng quên mật khẩu (Forgot Password)



Hình 6: Sequence diagram - Forgot Password

Hình trên mô tả luồng xử lý chức năng quên mật khẩu của hệ thống theo kiến trúc Controller – Service – Database – Redis. Trình tự xử lý được mô tả như sau:

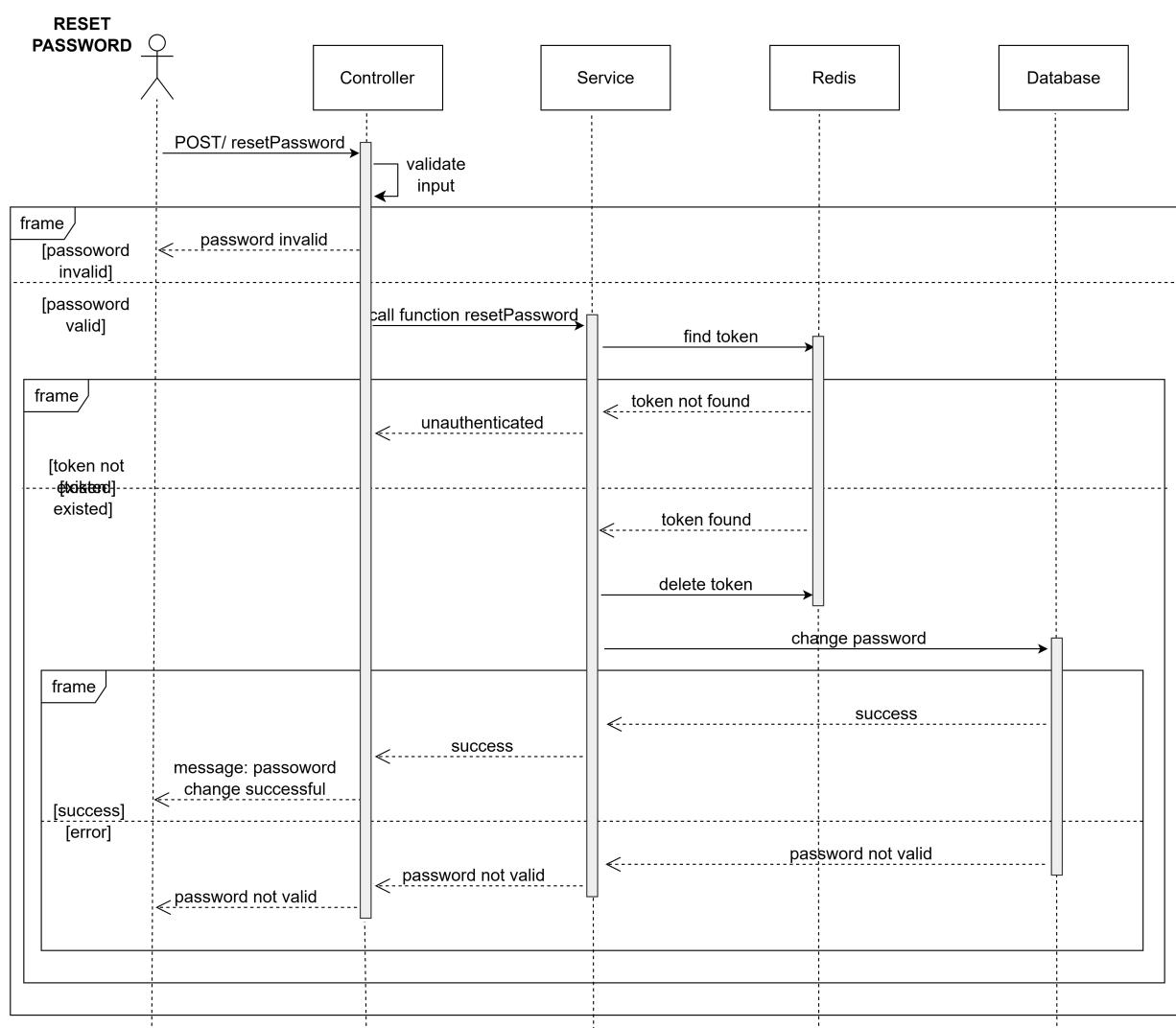
1. Người dùng gửi yêu cầu quên mật khẩu (POST /forgot-password) đến hệ thống, kèm theo địa chỉ email đã đăng ký.
2. Controller tiếp nhận yêu cầu và gọi hàm xử lý quên mật khẩu tại tầng Service.
3. Service gửi yêu cầu đến Database để kiểm tra sự tồn tại của email.
4. Trường hợp email không tồn tại trong hệ thống:

- Database trả về kết quả *email not found*.
- Service trả kết quả thất bại về cho Controller.
- Controller phản hồi thông báo email không tồn tại cho người dùng.

5. Trường hợp email tồn tại trong hệ thống:

- Service gửi yêu cầu đến Redis để tìm và xóa token đặt lại mật khẩu cũ (nếu có).
- Service tạo token đặt lại mật khẩu mới và lưu vào Redis tương ứng với email.
- Service gửi email xác thực đặt lại mật khẩu đến địa chỉ email của người dùng.
- Controller phản hồi thông báo gửi email thành công cho người dùng.

2.2.6 Luồng đặt lại mật khẩu (Reset Password)



Hình 7: Sequence diagram - Reset Password

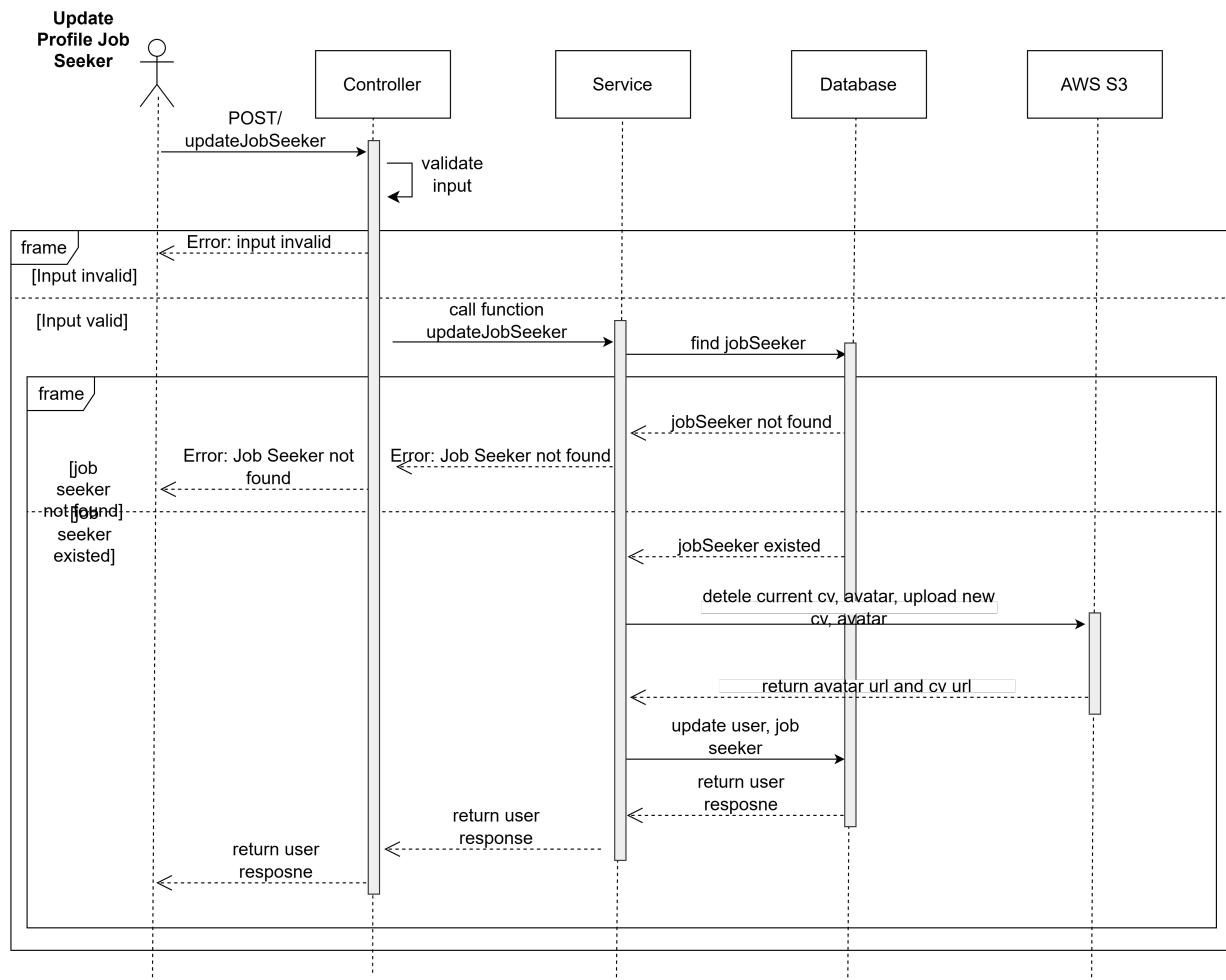
Hình trên mô tả luồng xử lý chức năng đặt lại mật khẩu của hệ thống theo kiến trúc Controller – Service – Redis – Database. Trình tự xử lý được mô tả như sau:

1. Người dùng gửi yêu cầu đặt lại mật khẩu (POST /reset-password) đến hệ thống, kèm theo token đặt lại mật khẩu và mật khẩu mới.



2. Controller tiếp nhận yêu cầu và kiểm tra tính hợp lệ của mật khẩu mới.
3. Trường hợp mật khẩu mới không hợp lệ:
 - Controller trả về thông báo *password invalid* cho người dùng.
4. Trường hợp mật khẩu mới hợp lệ:
 - Controller gọi hàm xử lý đặt lại mật khẩu tại tầng Service.
 - Service gửi yêu cầu đến Redis để kiểm tra sự tồn tại của token.
5. Trường hợp token không tồn tại hoặc không hợp lệ:
 - Redis trả về kết quả *token not found*.
 - Service trả kết quả *unauthenticated* về cho Controller.
 - Controller phản hồi thông báo lỗi xác thực cho người dùng.
6. Trường hợp token hợp lệ:
 - Service yêu cầu Redis xóa token đặt lại mật khẩu.
 - Service gửi yêu cầu đến Database để cập nhật mật khẩu mới cho người dùng.
7. Trường hợp cập nhật mật khẩu thành công:
 - Database trả về kết quả *success*.
 - Service trả kết quả thành công về cho Controller.
 - Controller phản hồi thông báo *password change successful* cho người dùng.
8. Trường hợp cập nhật mật khẩu thất bại:
 - Database trả về kết quả *password not valid*.
 - Service trả kết quả lỗi về cho Controller.
 - Controller phản hồi thông báo lỗi đặt lại mật khẩu cho người dùng.

2.2.7 Luồng cập nhật hồ sơ JobSeeker (Update Profile JobSeeker)



Hình 8: Sequence diagram - Update Profile JobSeeker

Hình trên mô tả luồng xử lý chức năng cập nhật hồ sơ JobSeeker của hệ thống theo kiến trúc Controller – Service – Database – AWS S3. Trình tự xử lý được mô tả như sau:

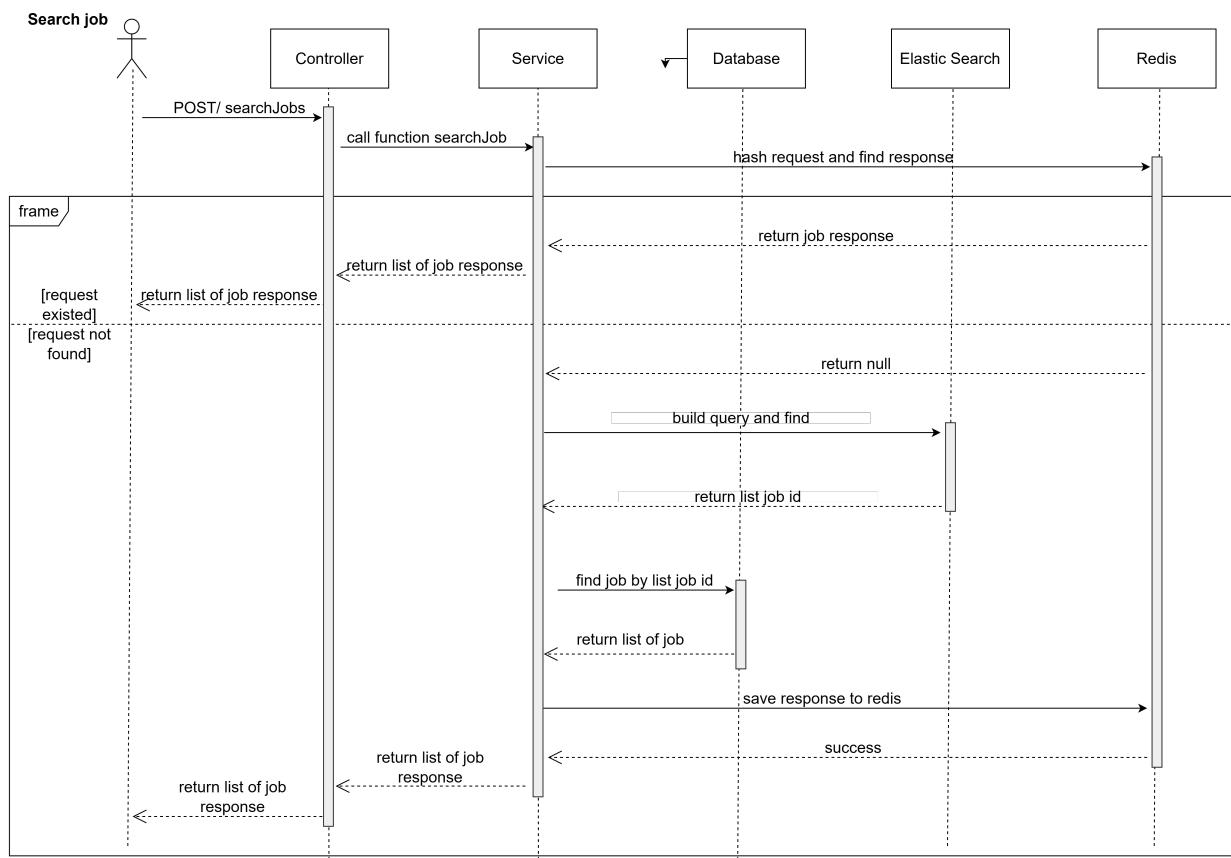
1. JobSeeker gửi yêu cầu cập nhật hồ sơ (POST /updateJobSeeker) đến hệ thống, kèm theo các thông tin hồ sơ mới, bao gồm dữ liệu cá nhân và tệp tin (CV, avatar).
2. Controller tiếp nhận yêu cầu và thực hiện kiểm tra tính hợp lệ của dữ liệu đầu vào.
3. Trường hợp dữ liệu đầu vào không hợp lệ:
 - Controller trả về thông báo lỗi *input invalid* cho JobSeeker.
4. Trường hợp dữ liệu đầu vào hợp lệ:
 - Controller gọi hàm xử lý cập nhật hồ sơ tại tầng Service.
 - Service gửi yêu cầu đến Database để tìm kiếm thông tin JobSeeker.
5. Trường hợp JobSeeker không tồn tại:
 - Database trả về kết quả *jobSeeker not found*.
 - Service trả kết quả thất bại về cho Controller.

- Controller phản hồi thông báo JobSeeker không tồn tại cho người dùng.

6. Trường hợp JobSeeker tồn tại:

- Service yêu cầu AWS S3 xóa các tệp tin hiện tại (CV và avatar) của JobSeeker.
- Service tải các tệp tin mới (CV và avatar) lên AWS S3.
- AWS S3 trả về đường dẫn (URL) của các tệp tin đã được tải lên.
- Service cập nhật thông tin hồ sơ JobSeeker và người dùng tương ứng trong Database.
- Database trả về thông tin người dùng đã được cập nhật.
- Service trả kết quả cập nhật thành công về cho Controller.
- Controller phản hồi thông tin hồ sơ JobSeeker đã được cập nhật cho người dùng.

2.2.8 Luồng tìm kiếm công việc (Search Job)



Hình 9: Sequence diagram - Search Jobs

Hình trên mô tả luồng xử lý chức năng tìm kiếm công việc của hệ thống theo kiến trúc Controller – Service – Database – Elasticsearch – Redis. Trình tự xử lý được mô tả như sau:

1. Người dùng gửi yêu cầu tìm kiếm công việc (POST /searchJobs) đến hệ thống với các tiêu chí tìm kiếm.
2. Controller tiếp nhận yêu cầu và gọi hàm xử lý tìm kiếm công việc tại tầng Service.
3. Service tạo khóa băm (hash key) từ nội dung yêu cầu và gửi yêu cầu đến Redis để kiểm tra dữ liệu tìm kiếm đã được lưu trong bộ nhớ đệm hay chưa.

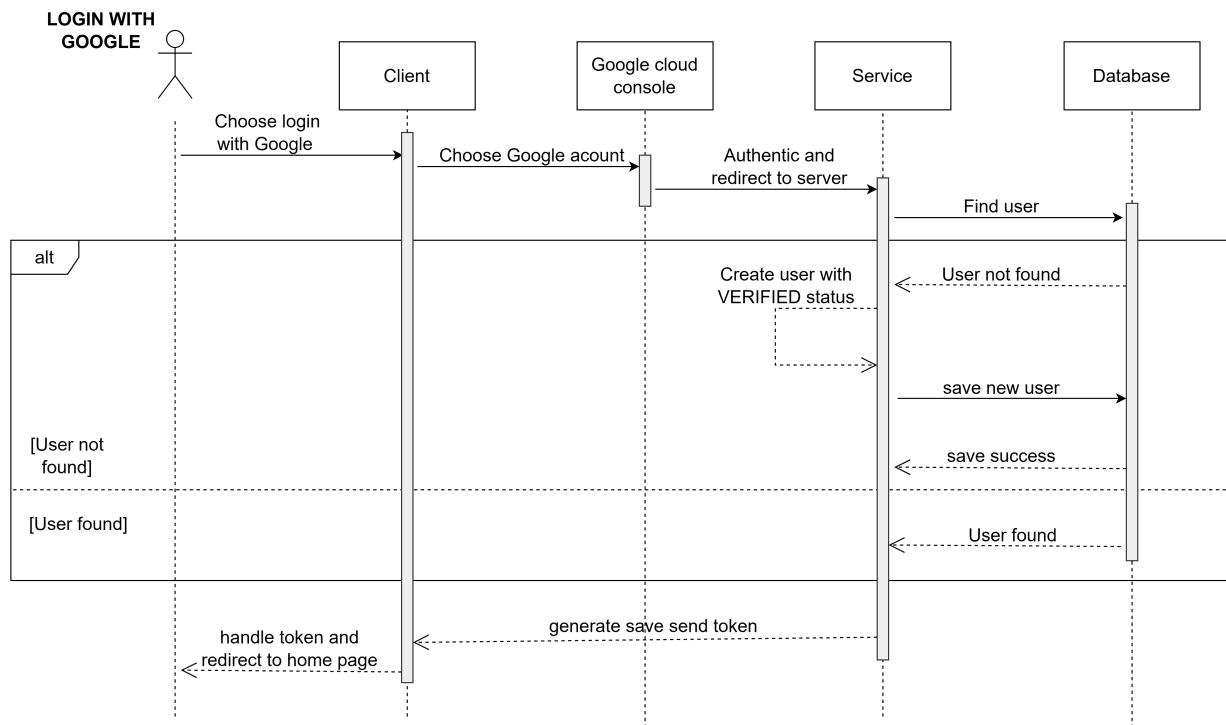
4. Trường hợp dữ liệu tìm kiếm đã tồn tại trong Redis:

- Redis trả về danh sách công việc tương ứng.
- Service trả danh sách công việc về cho Controller.
- Controller phản hồi danh sách công việc cho người dùng.

5. Trường hợp dữ liệu tìm kiếm chưa tồn tại trong Redis:

- Service xây dựng truy vấn tìm kiếm dựa trên các tiêu chí và gửi yêu cầu đến ElasticSearch.
- ElasticSearch thực hiện tìm kiếm và trả về danh sách các *jobId* phù hợp.
- Service gửi yêu cầu đến Database để lấy thông tin chi tiết công việc theo danh sách *jobId*.
- Database trả về danh sách công việc tương ứng cho Service.
- Service lưu kết quả tìm kiếm vào Redis để phục vụ các lần truy vấn sau.
- Service trả danh sách công việc về cho Controller.
- Controller phản hồi danh sách công việc cho người dùng.

2.2.9 Luồng đăng nhập bằng Google (Login with Google)



Hình 10: Sequence diagram - Login with Google

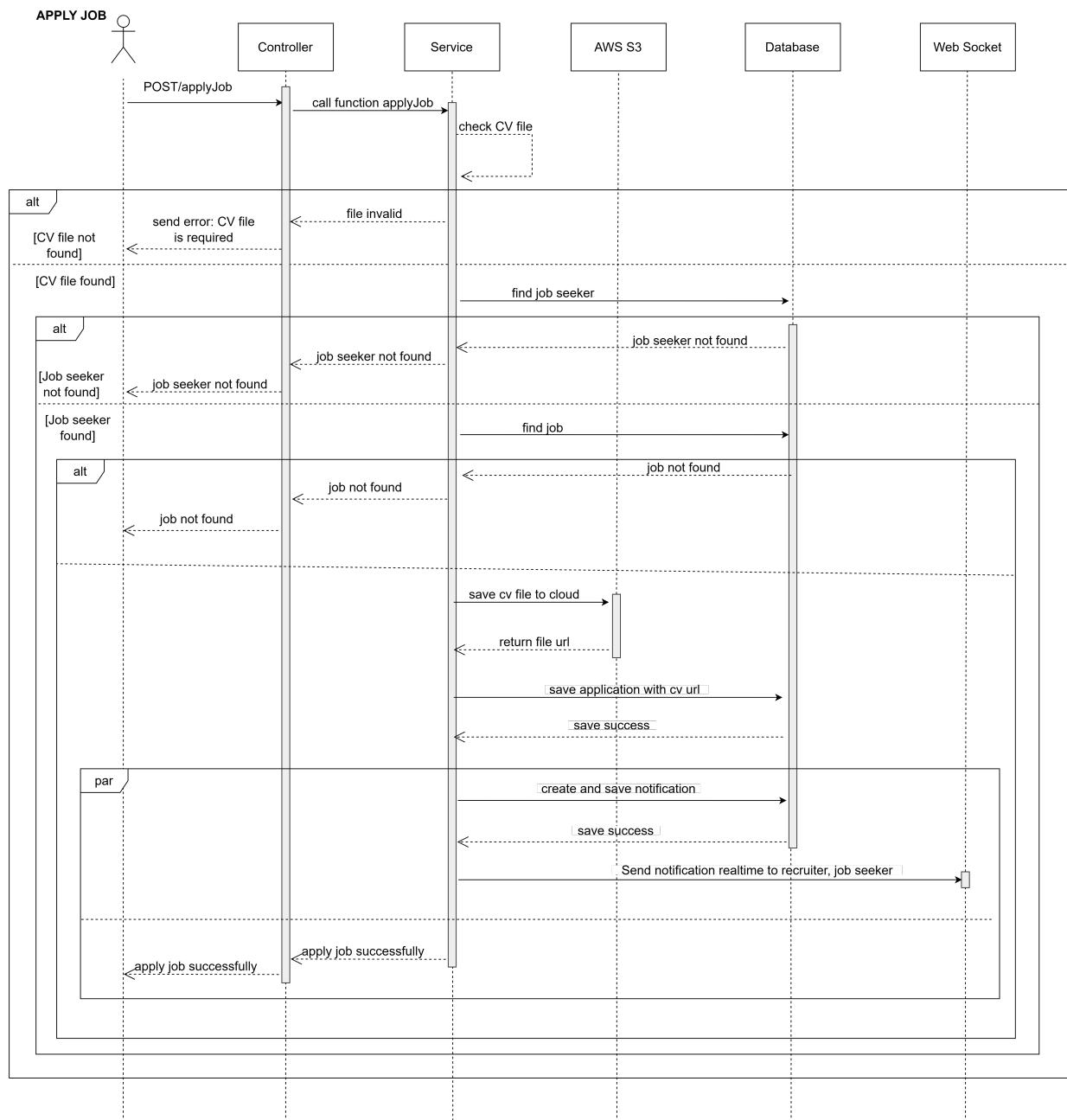
Hình trên mô tả luồng xử lý chức năng đăng nhập bằng Google của hệ thống thông qua dịch vụ xác thực của Google theo kiến trúc Client – Google Cloud – Service – Database. Trình tự xử lý được mô tả như sau:

1. Người dùng chọn chức năng đăng nhập bằng Google trên giao diện Client.
2. Client chuyển người dùng đến trang chọn tài khoản Google.



3. Người dùng chọn tài khoản Google và thực hiện xác thực với hệ thống của Google.
4. Sau khi xác thực thành công, Google Cloud Console chuyển hướng (redirect) người dùng kèm theo thông tin xác thực về server của hệ thống.
5. Service tiếp nhận thông tin xác thực từ Google và gửi yêu cầu đến Database để kiểm tra sự tồn tại của người dùng.
6. Trường hợp người dùng chưa tồn tại trong hệ thống:
 - Database trả về kết quả *User not found*.
 - Service tạo mới người dùng với trạng thái *VERIFIED*.
 - Service lưu thông tin người dùng mới vào Database.
 - Database trả về kết quả lưu thành công.
7. Trường hợp người dùng đã tồn tại trong hệ thống:
 - Database trả về kết quả *User found*.
8. Service tạo token xác thực cho người dùng và gửi token về Client.
9. Client xử lý token xác thực và chuyển hướng người dùng đến trang chủ của hệ thống.

2.2.10 Luồng ứng tuyển công việc (Apply Job)



Hình 11: Sequence diagram - Apply Job

Hình trên mô tả luồng xử lý chức năng ứng tuyển công việc của hệ thống theo kiến trúc Controller – Service – AWS S3 – Database – WebSocket. Trình tự xử lý được mô tả như sau:

1. JobSeeker gửi yêu cầu ứng tuyển công việc (POST /applyJob) đến hệ thống, kèm theo thông tin công việc và tệp tin CV.
2. Controller tiếp nhận yêu cầu và gọi hàm xử lý ứng tuyển công việc tại tầng Service.
3. Service kiểm tra sự tồn tại và tính hợp lệ của tệp tin CV.
4. Trường hợp không có tệp tin CV:
 - Service xác định tệp tin không hợp lệ.



- Controller phản hồi lỗi *CV file is required* cho JobSeeker.

5. Trường hợp tệp tin CV hợp lệ:

- Service gửi yêu cầu đến Database để tìm kiếm thông tin JobSeeker.

6. Trường hợp JobSeeker không tồn tại:

- Database trả về kết quả *job seeker not found*.
- Service trả lỗi về cho Controller.
- Controller phản hồi thông báo JobSeeker không tồn tại.

7. Trường hợp JobSeeker tồn tại:

- Service gửi yêu cầu đến Database để tìm kiếm thông tin công việc.

8. Trường hợp công việc không tồn tại:

- Database trả về kết quả *job not found*.
- Service trả lỗi về cho Controller.
- Controller phản hồi thông báo công việc không tồn tại.

9. Trường hợp JobSeeker và công việc đều tồn tại:

- Service tải tệp tin CV lên AWS S3.
- AWS S3 trả về đường dẫn (URL) của tệp tin CV.
- Service lưu thông tin ứng tuyển kèm URL CV vào Database.

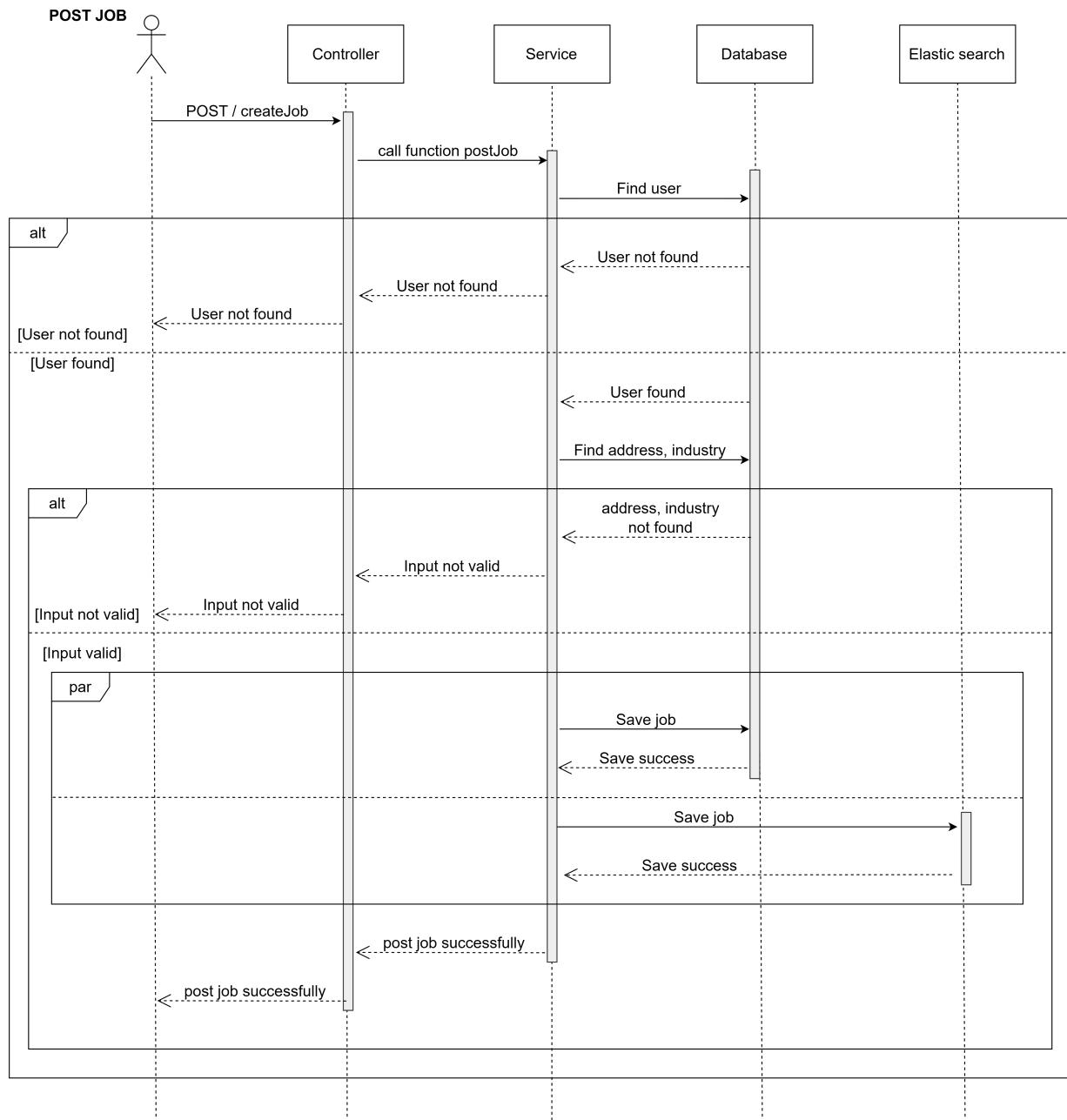
10. Song song với việc lưu đơn ứng tuyển:

- Service tạo và lưu thông báo ứng tuyển vào Database.
- Service gửi thông báo real-time đến Recruiter và JobSeeker thông qua WebSocket.

11. Sau khi tất cả các bước xử lý hoàn tất:

- Service trả kết quả ứng tuyển thành công về cho Controller.
- Controller phản hồi thông báo *apply job successfully* cho JobSeeker.

2.2.11 Luồng đăng tin tuyển dụng (Post Job)



Hình 12: Sequence diagram - Post Job

Hình trên mô tả luồng xử lý chức năng đăng tin tuyển dụng của hệ thống theo kiến trúc Controller – Service – Database – Elasticsearch. Trình tự xử lý được mô tả như sau:

1. Recruiter gửi yêu cầu đăng tin tuyển dụng (POST /createJob) đến hệ thống, kèm theo các thông tin mô tả công việc.
2. Controller tiếp nhận yêu cầu và gọi hàm xử lý đăng tin tuyển dụng tại tầng Service.
3. Service gửi yêu cầu đến Database để kiểm tra sự tồn tại của người dùng.
4. Trường hợp người dùng không tồn tại:
 - Database trả về kết quả *User not found*.



- Service trả kết quả thất bại về cho Controller.
- Controller phản hồi thông báo người dùng không tồn tại cho Recruiter.

5. Trường hợp người dùng tồn tại:

- Service tiếp tục kiểm tra các thông tin liên quan như địa chỉ và lĩnh vực ngành nghề.

6. Trường hợp dữ liệu đầu vào không hợp lệ (địa chỉ hoặc ngành nghề không tồn tại):

- Database trả về kết quả *address, industry not found*.
- Service xác định dữ liệu không hợp lệ.
- Controller phản hồi thông báo *Input not valid* cho Recruiter.

7. Trường hợp dữ liệu đầu vào hợp lệ:

- Service thực hiện lưu thông tin công việc vào Database.
- Song song với việc lưu vào Database, Service lưu thông tin công việc vào Elastic-Search để phục vụ chức năng tìm kiếm.

8. Sau khi lưu dữ liệu thành công:

- Database và ElasticSearch trả về kết quả *Save success*.
- Service trả kết quả đăng tin thành công về cho Controller.
- Controller phản hồi thông báo *post job successfully* cho Recruiter.

2.3 Class diagrams

3 Architecture design

3.1 Database design

3.2 Software architecture

4 Implementation