High-level Architecture UML Diagrams Document

팀명: UOS Works		프로젝트명: 알바시대
2014920047 임진우	2020920031 서제임스	2020920058 조성채
2022920010 김동하	2022920024 박동찬	2022920045 이승목

Revision History

version	Author	Description	Date
v01	전체	전체 로직 작성	24.11.18
v02	서제임스, 박동찬, 임진우	비밀번호 찾기 로직 변경에 따른 각 diagram에 verification, reset password 추가	24. 12. 08

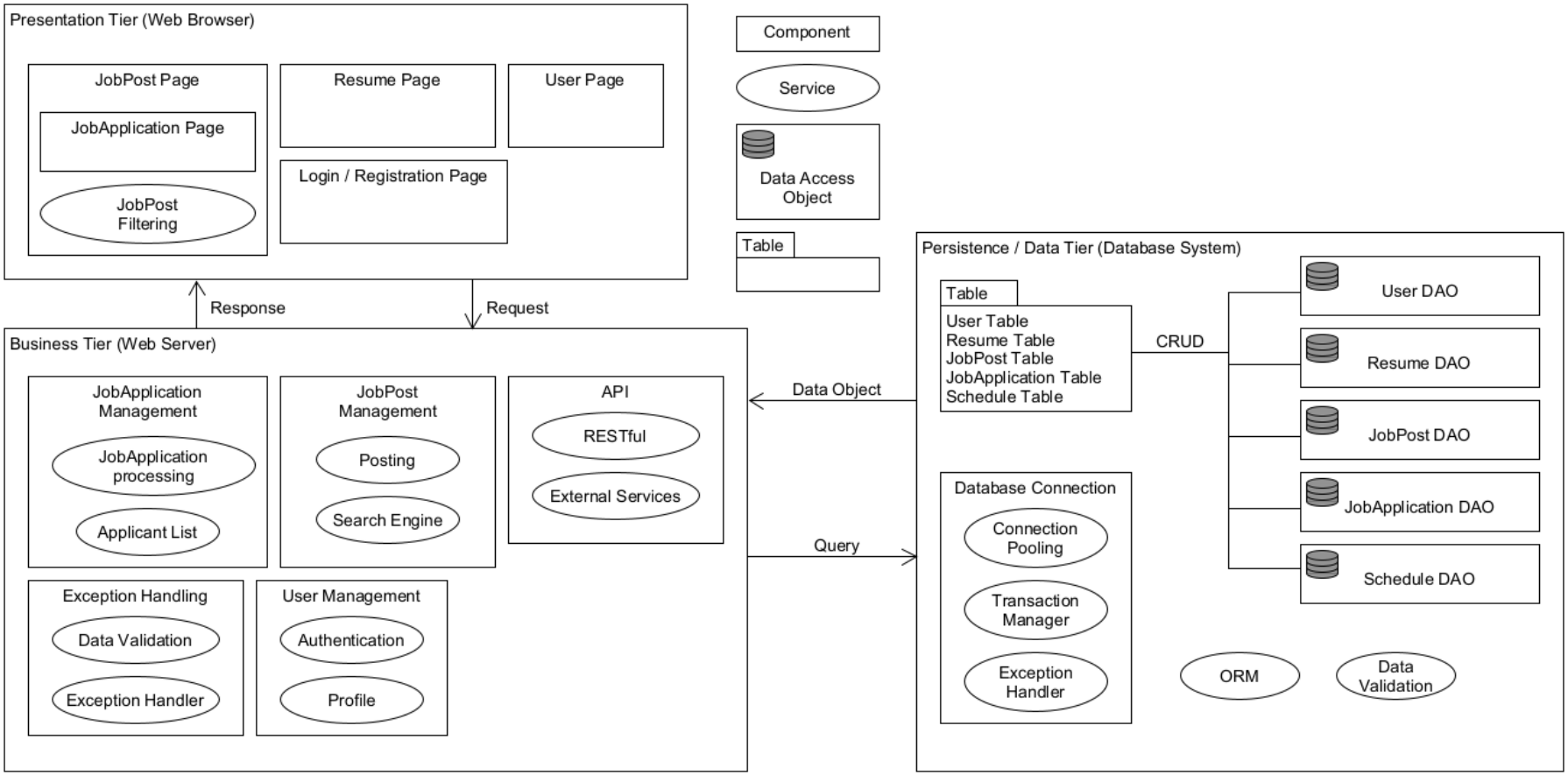
Table of Contents

1. Client-Server (3-Tier-style) ---------- 5
2. Model-View-Controller ---------- 8
3. Class Diagram ---------- 11
4. Sequence Diagaram ---------- 16

Client-Server(3-tier-style), MVC architecture 선정 이유

- 다수의 사용자에게 서비스를 제공할 수 있도록 서비스를 확장하는 것이 용이함
- 모듈화를 통해 각 컴포넌트들의 책임을 분리하여 효율적으로 개발하고 테스트할 수 있음

Client-Server (3-Tier-style)



Client-Server (3-Tier-style)

전체적인 시나리오

1. 사용자가 알바시대 메인 페이지에 접속하여 채용정보 페이지에서 필터링 기능을 사용하여 원하는 조건에 맞는 알바 정보를 검색
2. 비즈니스 계층은 퍼시스턴스/데이터 계층에 필요한 데이터를 요청
3. 퍼시스턴스/데이터 계층은 동적 쿼리를 생성하여 데이터베이스에서 데이터를 조회하고 결과를 비즈니스 계층에게 반환
4. 비즈니스 계층은 퍼시스턴스/데이터 계층에서 받은 데이터를 가공하거나 추가 로직을 적용한 후 프레젠테이션 계층에 전달
5. 프레젠테이션 계층은 필터링 된 채용정보를 사용자에게 표시

Client-Server (3-Tier-style)

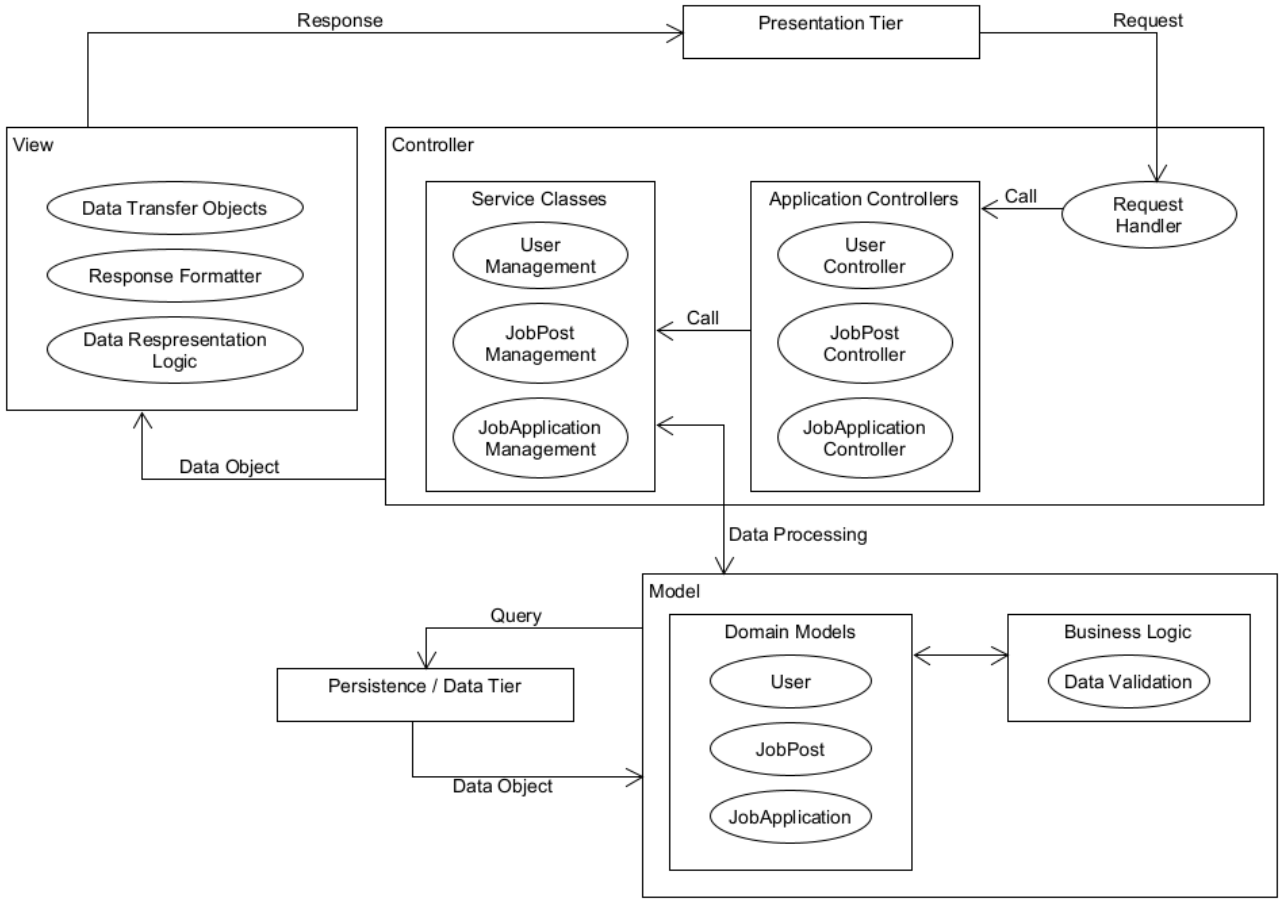
장점

- 계층 별로 개발을 분담할 수 있어 효율적인 개발이 가능합니다.(FE: 프레젠테이션 계층, BE: 비즈니스, 퍼시스턴스/데이터 계층)
- 특정 계층의 부하가 증가할 경우 해당 계층만 독립적으로 확장이 가능합니다.
- 각 계층 별로 테스트를 수행할 수 있어 효율적으로 디버깅할 수 있습니다.

단점

- 여러 계층을 관리해야 하므로 프로젝트 관리의 부담이 증가합니다.
- 계층마다 데이터를 전달하고 처리하는 시간이 추가되어 응답 시간이 늘어날 수 있습니다.

Model-View-Controller



Model-View-Controller

전체적인 시나리오

1. 프레젠테이션 계층에서 컨트롤러로 요청을 전달
2. 컨트롤러는 해당 요청을 처리하기 위해 서비스 클래스를 호출
3. 서비스 클래스는 필요한 모델을 이용하여 비즈니스 로직을 수행
4. 처리 결과를 DTO 형태로 변환하고, 뷰를 통해 프레젠테이션 계층에 전달
5. 프레젠테이션 계층은 전달받은 데이터를 사용자에게 표시

Model-View-Controller

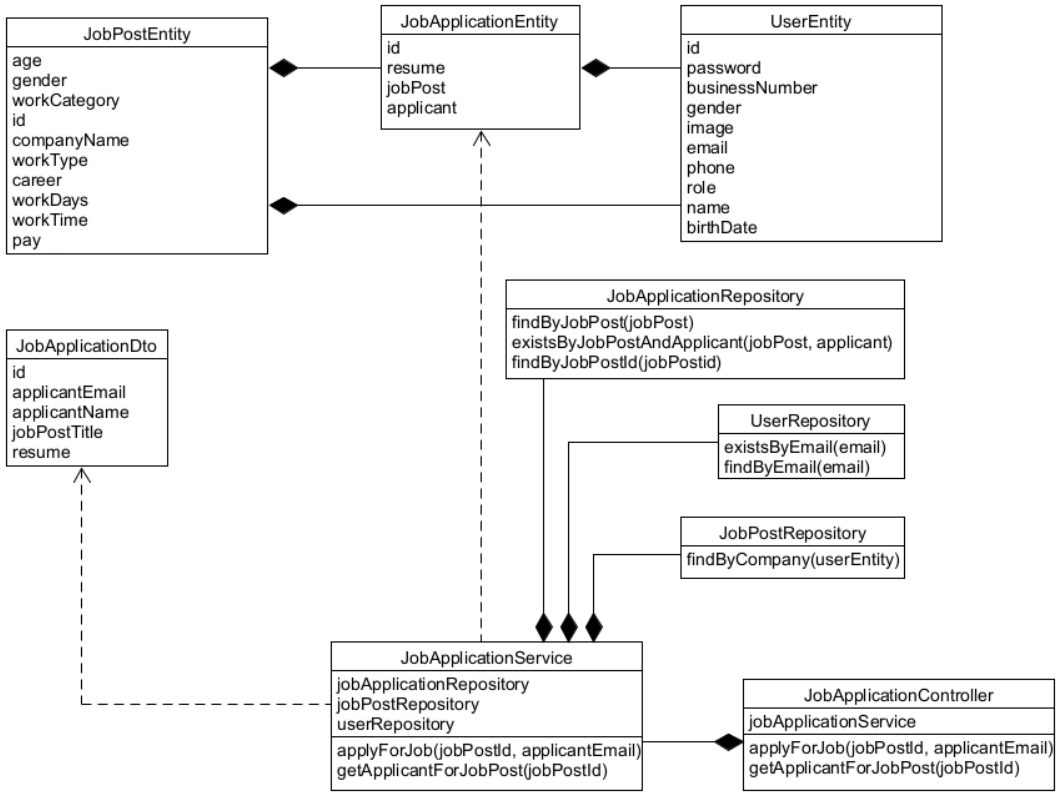
장점

- UI 변경이 비즈니스 로직에 영향을 주지 않아 다양한 인터페이스를 적용할 수 있습니다.
- 모델, 뷰, 컨트롤러 각각의 책임이 분리되어 있어 문제가 발생했을 때 원인을 신속하게 파악할 수 있습니다.
- 모델 계층을 확장하여 기존의 흐름을 크게 변경하지 않고도 새로운 기능을 확장할 수 있습니다.

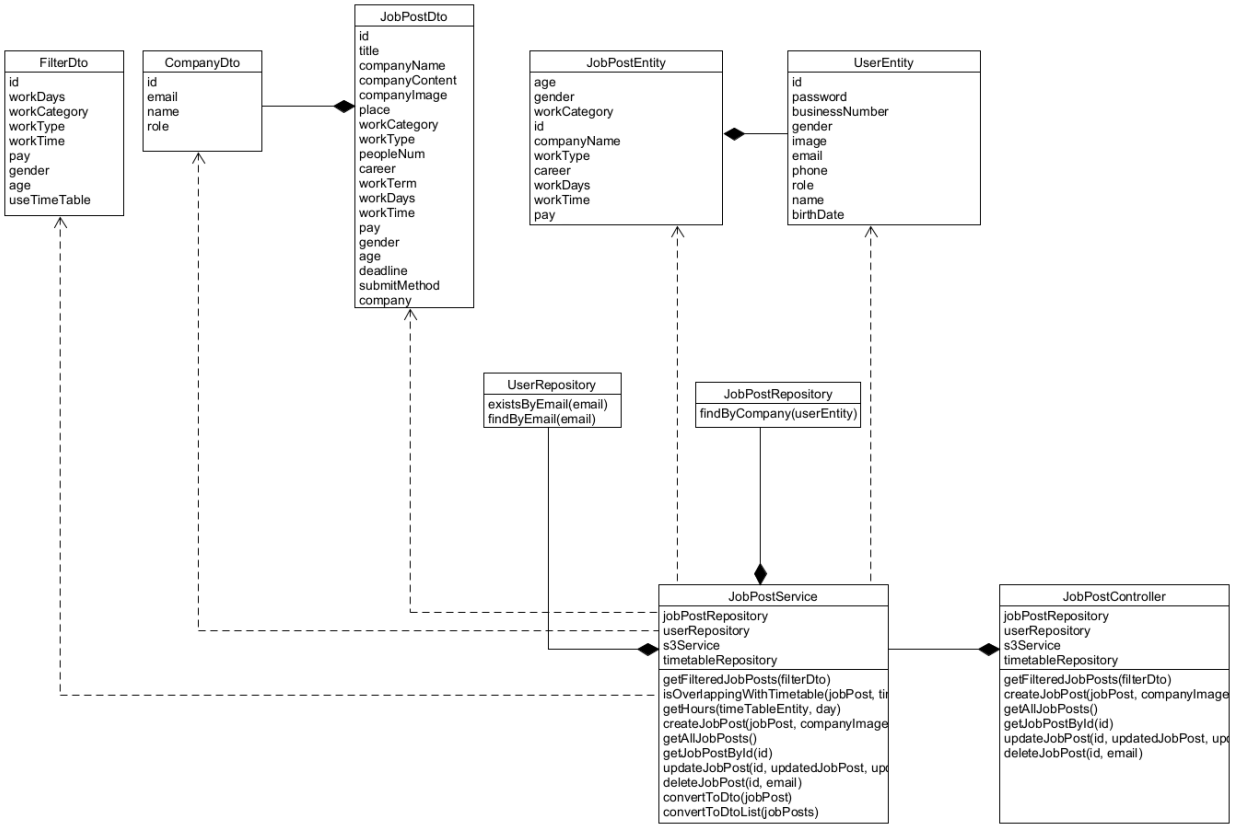
단점

- MVC 패턴의 엄격한 구조로 인해 유연한 설계가 어려울 수 있습니다.
- 코드의 양이 증가하여 개발 생산성이 저하될 수 있습니다.

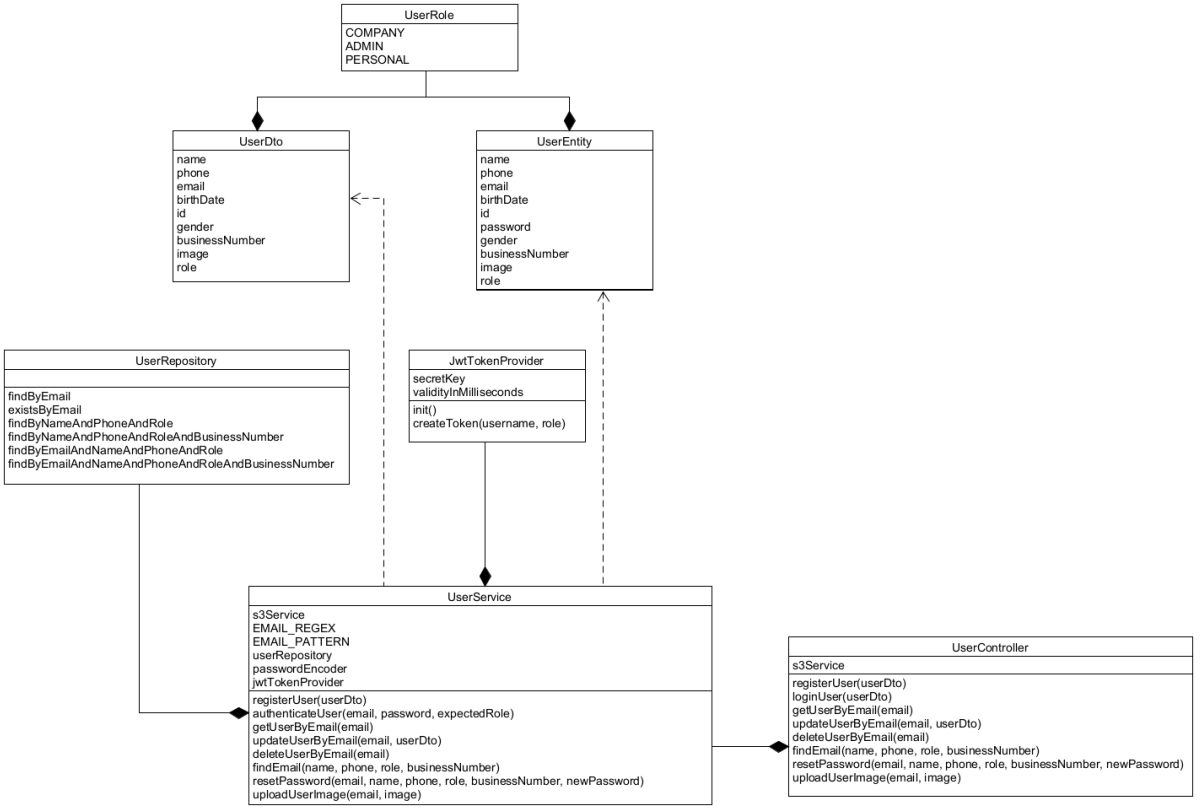
Class Diagram - JobApplication



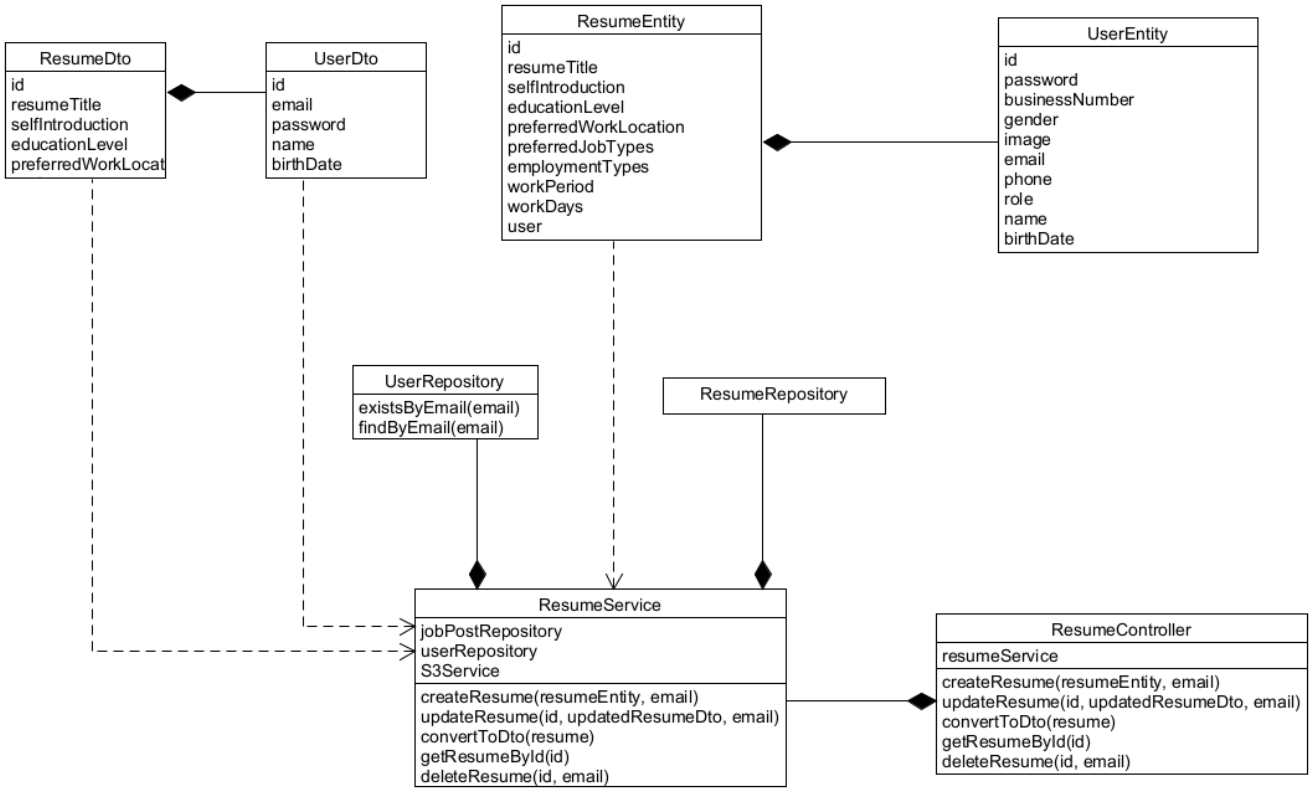
Class Diagram - JobPost



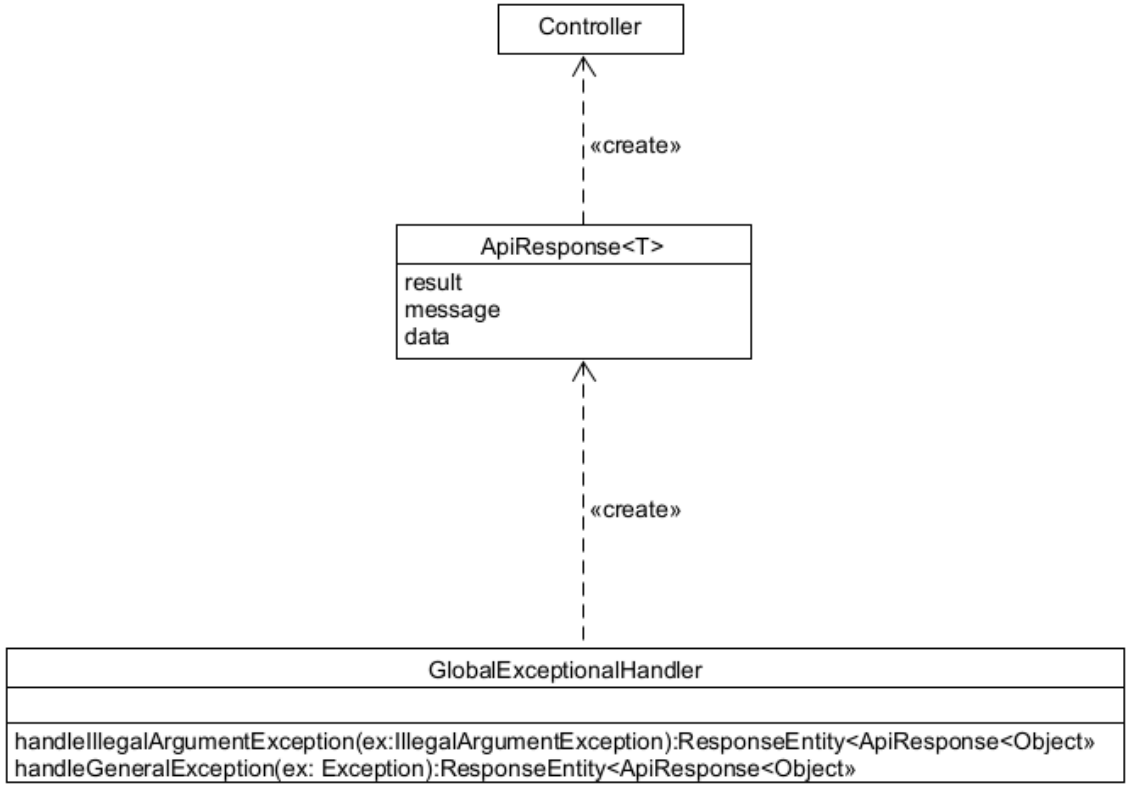
Class Diagram - User



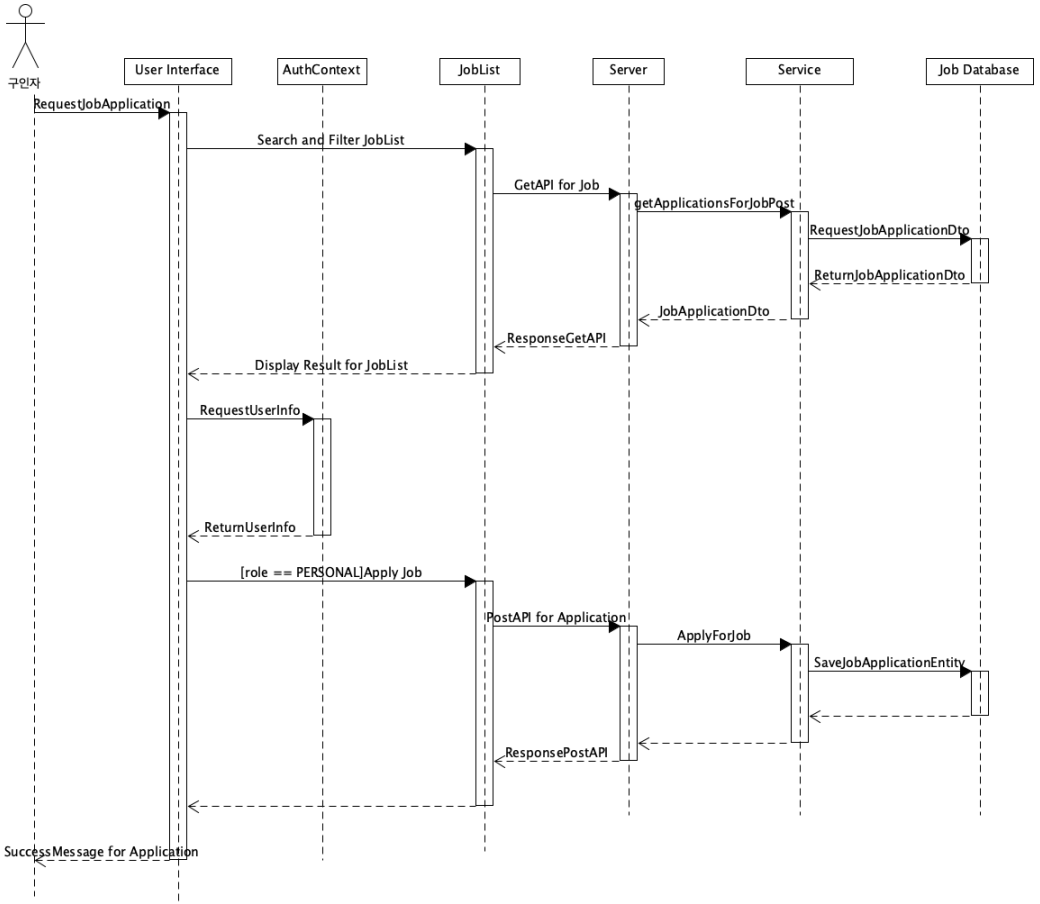
Class Diagram - Resume



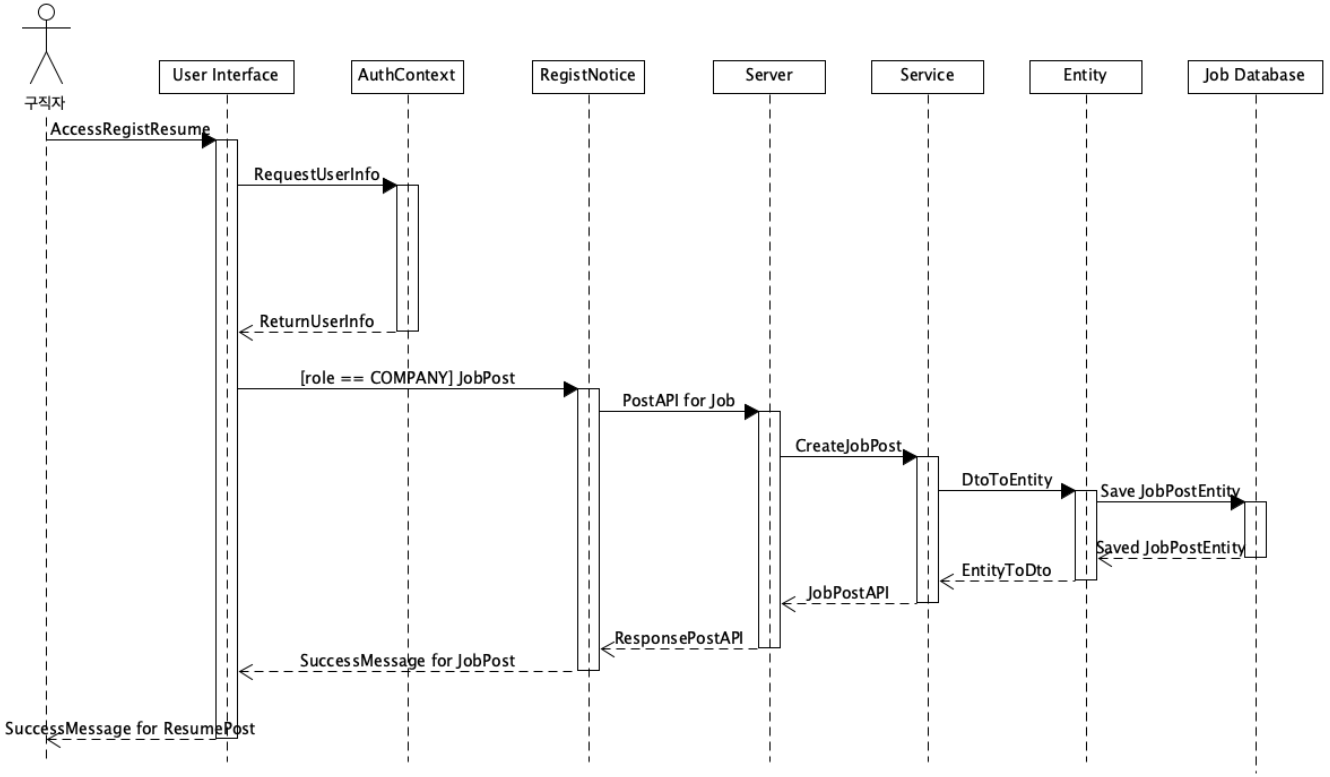
Class Diagram - etc



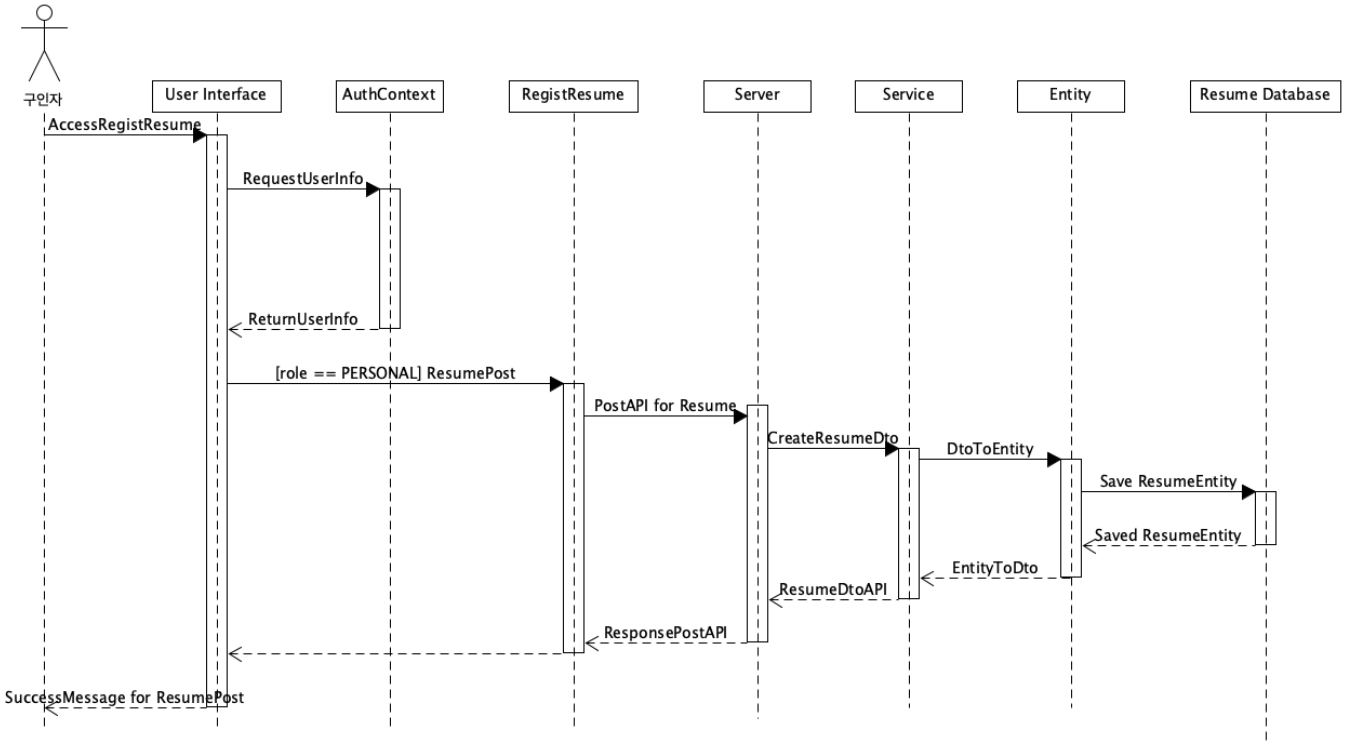
Sequence Diagram - JobApplication



Sequence Diagram - JobPost



Sequence Diagram - ResumePost



Sequence Diagram - Auth