

# Connect-U

## Requirement Specification

### Team 11

2017310605 김도현

2019311895 김연재

2016310401 이광호

2017314655 이인수

2017312222 최영우

# Table of Contents

1. Preface.....	8
1.1. Readership.....	8
A. User Requirements Readership.....	8
B. System Requirements Readership.....	8
1.2 Document Structure.....	8
A. Introduction.....	8
B. Glossary.....	8
C. User Requirements Definition.....	8
D. System Architecture.....	9
E. System Requirements.....	9
F. System Models.....	9
G. System Evolution.....	9
H. Appendices.....	9
I. Index.....	9
J. Reference.....	9
2. Introduction.....	10
2.1. Needs.....	10
2.2. System overview.....	11
2.3. Expected Effects.....	12
A. Professor.....	12
B. School Staff.....	12
C. Student.....	12
3. Glossary.....	13

4. User Requirements Definition.....	14
4.1 Functional Requirements.....	14
A. Sign Up.....	14
B. Sign In & Certification of Belonging.....	14
C. My Page.....	14
D. Recruit Enrollment.....	14
I . Professor’s Laboratory.....	15
II. Project / Study Recruit.....	15
III. Announcement.....	15
E. Job Search.....	15
F. Recruit Recommendation.....	16
G. Workforce Matching.....	16
H. Notification.....	16
I. User Evaluating System.....	16
J. Declaration System.....	17
K. Blacklist and Suspension.....	17
4.2 Non-functional Requirements.....	17
A. Product Requirements.....	17
1) Usability.....	17
2) Security.....	17
3) Reliability.....	18
4) Performance.....	18
B. Organizational Requirements.....	18
1) Development Requirements.....	18

2) Environment Requirements.....	18
C. External Requirements.....	19
1) Regulatory Requirements.....	19
2) Safety Requirements.....	19
5. System Architecture.....	20
5.1 System Overview.....	20
5.2 Front-end Architecture.....	21
5.3 Back-end Architecture.....	22
5.4 Job Recommendation System.....	23
6. System Requirements specification.....	24
6.1 Functional Requirements.....	24
A. Sign Up.....	24
B. Sign In & Certification of Belonging.....	25
C. My Page.....	26
D. Recruit Enrollment.....	27
E. Job Search.....	28
F. Recruit Recommendation.....	29
G. Workforce Matching.....	29
H. Notification.....	30
I. User Evaluating System.....	31
J. Declaration System.....	31
K. Blacklist and Suspension.....	32
6.2 Non-functional Requirements.....	33
A. Product Requirements.....	33

1) Usability.....	33
2) Security.....	33
3) Reliability.....	33
4) Performance.....	34
B. Organizational Requirements.....	34
1) Development Requirements.....	34
2) Environment Requirements.....	34
C. External Requirements.....	34
1) Regulatory Requirements.....	34
2) Safety Requirements.....	35
6.3 Scenario Examples.....	35
A. Login Scenario.....	35
1) Initial Assumption.....	35
2) Normal flow of events.....	35
3) What can go wrong.....	35
4) System state on completion.....	35
B. Search Scenario.....	36
1) Initial Assumption.....	36
2) Normal flow of events.....	36
3) What can go wrong.....	36
4) System state on completion.....	36
C. Registration Scenario.....	36
1) Initial Assumption.....	36
2) Normal flow of events.....	37

3) What can go wrong.....	37
4) System state on completion.....	37
D. Recommendation Scenario.....	37
1) Initial Assumption.....	37
2) Normal flow of events.....	37
3) What can go wrong.....	37
4) System state on completion.....	38
E. Matching Scenario.....	38
1) Initial Assumption.....	38
2) Normal flow of events.....	38
3) What can go wrong.....	38
4) System state on completion.....	38
F. Rating Scenario.....	39
1) Initial Assumption.....	39
2) Normal flow of events.....	39
3) What can go wrong.....	39
4) System state on completion.....	39
7. System Models.....	39
7.1 Context Models.....	40
A. Context Diagram.....	40
B. Process Diagram.....	40
7.2 Interaction Models.....	41
A. Use case Diagram.....	41
B. Tabular Description of Use case Diagram.....	41

1) Login.....	41
2) Register and Authorization.....	41
3) Profile Update.....	42
4) Post Project.....	42
5) Participate Project.....	43
6) Project Review.....	43
7.3 Behavioral Models.....	44
8. System Evolution.....	45
8.1. Job Offer System.....	45
8.2. Recommendation system by machine learning.....	45
8.3. Sanction against malicious evaluation.....	45
8.4. Extension to Web.....	46
9. Appendices.....	46
10. Index.....	47

# 1. Preface

Preface 에서는 본 문서의 예상 독자를 정의하고 문서의 전체적인 구조에 대해 간략히 설명한다. 문서에 변경 사항이 생겼을 때 변경 내용과 이유를 포함한 version history를 각 version 마다 기록한다.

## 1.1 Readership

Readership에서는 본 문서를 읽을 독자층에 따라 user requirements와 system requirements로 구분하여 각각 어떤 내용으로 구성되어 있는지 설명한다.

### A. User Requirements Readership

User requirements의 예상 독자는 system을 사용할 user이다. 이 문서는 user 관점에서 requirement를 이해하기 쉽도록 natural language로 작성되어 있으며 diagram과 table을 이용하여 설명한다.

### B. System Requirements Readership

System requirements의 예상 독자는 system을 개발하는 개발자이다. 또한 client와 contractor의 계약 문서로 사용될 수 있게 자세하게 기술한다. 이 문서는 system의 functional requirements와 non-functional requirements를 자세하게 설명한다.

## 1.2 Document Structure

### A. Introduction

Introduction에서는 system의 필요성을 제시하고 system의 구조 및 기능을 간략하게 설명한다. 그리고 system의 기대효과에 대해서 설명한다.

### B. Glossary

문서에서 사용된 기술 용어를 설명한다.

### C. User Requirements Definition

User requirements definition에서는 시스템이 제공하는 서비스에 대해 설명한다.



Functional requirement과 non-functional requirement로 분류하여 설명한다. 사용자의 관점에서 이해하기 쉽게 자연어로 기술하며 다이어그램과 표를 같이 사용한다.

#### **D. System Architecture**

System architecture에서는 시스템의 전체적인 구조에 대해 개략적으로 기술하고 서브시스템의 구성, 서브시스템 간의 관계를 설명한다.

#### **E. System Requirements**

Specification System Requirements specification에서는 user requirement definition을 바탕으로 functional requirements와 non-functional requirements를 개발자가 이해할 수 있도록 자세하게 기술한다.

#### **F. System Models**

System models에서는 하나의 관점으로 시스템을 추상인 모델로 나타내는 system modeling에 대해 기술한다. 여기서 사용되는 관점은 External Perspective, Interaction Perspective, Structural Perspective, Behavioral Perspective이다. 각 시스템의 component와 system의 관계, component간의 관계, system을 둘러싼 환경 사이의 관계를 modeling하여 UML diagram으로 나타낸다.

#### **G. System Evolution**

System evolution에서는 추후 시스템에 요청될 수 있는 변경 사항에는 어떤 것이 있는지 가정하고 설명한다. 시스템의 유지보수에 영향을 미칠 수 있는 시장의 변화, 사용자의 요구사항의 변화를 살펴본다.

#### **H. Appendices**

#### **I. Index**

#### **J. Reference**

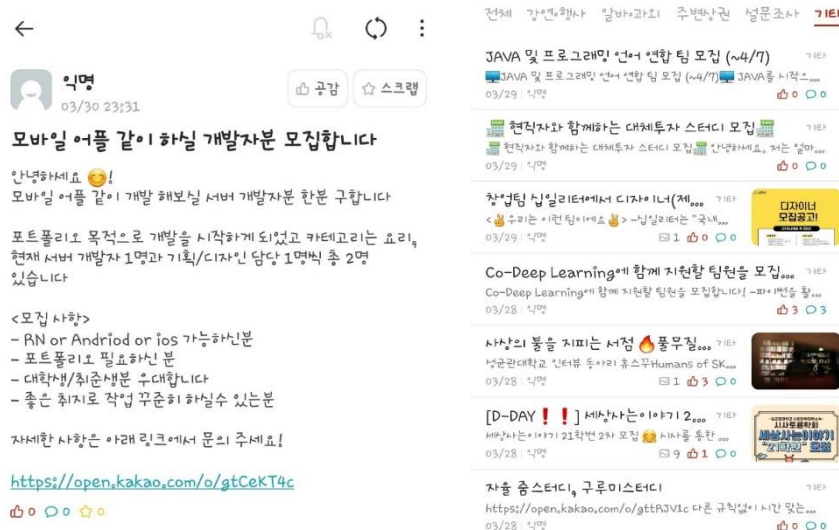
## 2. Introduction

이 챕터에서는 Connect-U 시스템이 왜 필요한지에 대한 배경을 설명하고 시스템의 기능을 간략하게 서술하며, 시스템이 사람들에게 끼칠 영향을 예측하여 설명한다.

### 2.1. Needs

최근 정부나 기업이 주도하는 공모전 형태의 참여형 프로젝트가 늘어나고 있다. 특히 공모전이 경험과 스펙을 쌓을 수 있는 기회이고 취업특전을 제공하는 경우도 있다는 점에서 대학생의 참여가 두드러지게 나타나고 있다. 이러한 참여형 프로젝트의 경우 기획, 디자인, 개발 등 복합적인 요소가 필요한 주제가 주어지기 때문에 대부분 혼자 모두 도맡아서 진행하는 형태보다 팀을 이루어서 참여를 하는 경우가 많다. 팀을 이루기 위해서 친구, 선후배 등의 지인을 활용하는 전통적 방법을 사용할 수도 있지만 해당 방법을 통해 팀원을 구하는 것은 한계가 존재하기 때문에 일반적으로 웹상의 플랫폼을 사용하는 경우가 많을 것이다. 하지만 공모전의 정보를 모아서 볼 수 있는 플랫폼은 많으나 공모전을 같이 진행할 팀원을 모집할 플랫폼은 상대적으로 부족하다.

현재 대다수의 학생들은 팀원들을 모으기 위해서 '에브리타임'이나 '캠퍼스픽'을 활용하고 있다. 그러나 '에브리타임'과 '캠퍼스픽'은 서비스의 목적이 기본적으로 커뮤니티이기 때문에 팀원을 모을 때 적합하지 않은 점이 존재한다. 우선 모집 양식이 정해져 있지 않기 때문에 공모전에 참여하려는 사람이 자신에게 적합한 모집 공고를 찾아내는 것이 상당히 힘들다. 또한 해당 커뮤니티가 익명을 전제로 하기 때문에 상대방의 역량을 확인할 방법이 없고 프로젝트를 진행하는 도중에 멋대로 그만두는 등의 문제를 상습적으로 일으키는 사람인지를 확인할 방법이 없다. 즉 현재 존재하는 플랫폼들은 같이 진행할 팀원의 신뢰도를 확인할 수단이 부족하다.



## Everytime Application Recruitment example

경험을 얻을 수 있는 기회로 공모전뿐만 아니라 학부연구생에 대한 대학생들의 관심도 점점 높아지고 있다. 하지만 학부연구생을 신청하고자 때 교수님께 직접 찾아가는 방법을 제외하고 홈페이지 등의 플랫폼을 통해 할 수 있는 경우는 거의 없다. 연구분야나 랩실에 대한 정보를 얻을 수 있는 수단도 '김박사넷'이나 랩실의 홈페이지 등 상당히 제한적이다.



PhD.Kimnet example

대학생활 중에는 공모전, 학부연구생뿐만 아니라 산학협력프로젝트, 스터디, 설문 등 불특정 다수를 대상으로 사람들을 모집할 필요가 있는 경우가 많이 존재한다. 그러나 이러한 활동들이 공모전의 경우와 마찬가지로 사람들을 모집할 때 적합한 플랫폼이 없다. 그렇기 때문에 인원을 모집하는 일에 도움을 주기 위해서 모집 공고를 분야에 따라 검색할 수 있고 참여자의 신뢰도가 보장되는 인력 중개 시스템이 필요하다.

## 2.2. System overview

Connect-U(가제)는 모집자가 모집 카테고리(공모전, 스터디 등)를 설정하고 모집을 원하는 사람들의 조건(분야, 기간 등)을 설정해서 시스템이 제공하는 양식에 맞게 인원 모집 공고를 작성하고 올리면 참여를 원하는 지원자가 공고를 보고 신청해서 매칭을 시켜주는 인력 중개 시스템이다.

사용자가 분야와 기간 등 모집 공고에서 원하는 조건을 설정하여 검색할 수 있도록 기능을 지원해 자신에게 적합한 프로젝트에 손쉽게 참여가 가능하다. 추가적으로 관심분야를 설정할 수 있어 시스템에서 모집 공고를 추천받는 것도 가능하다.

또한 회원가입 시 사용자의 대학교 이메일을 통해 소속 인증 절차를 거쳐 가입하고 프로젝트 종료 후 함께했던 팀원을 평가하는 기능이 존재하여 사용자의 신뢰도를 확보한다. 그리고 마이페이지에 참여했던 프로젝트 내역과 팀원에게서 받은 평가지표가 있기 때문에 역량을 객관적으로 평가할 수 있다.

## **2.3. Expected Effects**

### **A. Professor**

학부연구생을 모집하는 것이 좀 더 용이해지며 공고를 통해 참여의사를 밝히는 학생이 플랫폼을 통해 사전에 연구실이나 랩실에 대한 정보를 취득할 수 있기 때문에 연구실에 더 적합한 학생을 채용하는 것이 가능해진다.

### **B. School Staff**

학생을 대상으로 설문을 하는 것이 더 쉬워지고 행정 보조 업무 등 대학생 인력을 모집하는 것이 용이해지기 때문에 효율성이 높아진다.

### **C. Student**

모집자의 입장에서 공모전, 스터디 등 원하는 조건에 맞는 인원을 모으는 것이 쉬워지고 평가 시스템이나 프로젝트 참여 이력 기능으로 기존보다 더 신뢰도가 있는 참여자와 진행을 하는 것이 가능해진다. 참여자의 입장에서는 검색을 통해 원하는 활동을 찾아내는 것이 가능해지기 때문에 공고의 가독성과 접근성이 좋아진다.

### 3. Glossary

Glossary에서는 본 문서에 사용된 기술적 용어들을 정의하며 일반 사용자도 이해하기 쉽게 설명한다. 또한 문서를 이해하기 위해 필요한 배경지식과 관련된 용어도 같이 설명한다.

Term	Description
User	Connect-U 서비스를 이용하는 사용자이다. 여기서는 주로 교수, 연구원, 학생, 교직원 등이 있다.
Sign up	사용자가 서비스를 이용하기 위하여 회원 가입을 하는 것을 의미한다. 이 과정이 끝나면 Login을 할 수 있다.
Login	사용자가 서비스에 접근하는 방법으로, 이 과정이 끝나면 사용자에게 다양한 권한을 제공한다.
My page	사용자 개인의 공간으로 개인 정보를 수정하고 여러 설정 조건들을 조정할 수 있는 공간이다.
일자리	Connect-U 서비스에서 recruiter가 recruitee를 모집하기 위해 공지하는 프로젝트, 스터디, 프로그램, 연구원 모집 등을 의미한다.
Recruiter(모집자)	Connect-U 서비스 이용자 중 프로젝트, 스터디 등의 일자리 정보를 제공하고 인원 모집 공고를 게시하는 유저이다.
Recruitee(지원자)	Connect-U 서비스 이용자 중 recruiter의 게시글을 확인하고 본인이 원하는 게시글에 참여를 신청하는 유저이다.
Authentication(인증)	Connect-U 서비스를 사용하기 위하여 사용자의 소속을 확인하는 것을 의미한다.
Database	대량 데이터의 집합을 의미한다.

## 4. User Requirements Definition

User requirements definition에서는 시스템이 제공하는 서비스를 바탕으로 functional requirement와 non-functional requirement로 분류하여 설명한다. 사용자의 관점에서 이해하기 쉽게 자연어로 기술한다.

### 4.1 Functional Requirements

#### A. Sign Up

학교 내부에서 일자리를 중개하는 시스템 특성에 따라 보안을 위하여 회원 가입을 요구한다. 가입 및 로그인 시 사용자는 이메일 주소와 비밀번호를 입력해야 한다. 가입 절차에는 추가 이메일 확인 절차가 필요하다. Firebase 인증을 사용하면 사용자 확인이 단순화되고 사용자 정보가 안전하게 보호된다.

#### B. Sign In & Certification of Belonging

가입 절차를 거친 인원만 시스템을 이용할 수 있도록 로그인을 요구한다. 회원가입 시 등록했던 이메일과 비밀번호를 맞게 입력해야만 이후의 서비스들을 이용할 수 있다. 회원 가입 이후 최초 한 번은 학교 내부 인원임을 인증하는 인증 절차를 거쳐야 한다.

#### C. My Page

마이 페이지는 개인정보 수정 메뉴와 나의 정보 확인 메뉴, 환경 설정 메뉴로 총 3개의 메뉴로 구성된다. 먼저 개인정보 수정 메뉴를 통해 회원 가입 시 입력했던 닉네임, 학과 등의 정보를 수정하고 관심 분야와 원하는 일자리의 조건 등의 추가 정보를 등록할 수 있다. 나의 정보 확인 메뉴로 지금까지 참여한 활동, 지원한 활동 및 모집, 본인의 랭킹 점수와 별점을 확인할 수 있다. 환경 설정 메뉴에서는 알림 설정, 권한 설정 등 시스템의 기본적인 설정을 할 수 있다. 알림 설정에는 이전에 설정한 관심분야 키워드 알림, 타인의 평가 알림, 관심있는 프로젝트의 지원 마감 임박 알림, 블랙리스트 알림, 진동 설정, 알림음 설정이 있다.

#### D. Recruit Enrollment

## **I . Professor's Laboratory**

Connect-U 서비스의 사용자인 교수가 연구생 모집을 위한 홍보를 할 수 있는 기능이 있다. 모집글을 등록하기 전 등록자의 간단한 신분 정보와 랭킹 점수 제공을 위한 정보 제공 동의 여부를 선택한다. 모집글을 작성할 때에는 양식에 맞춰 연구분야, 모집인원, 모집연구생 학위(undergraduate, MS, Ph.D), 타 학교 연구인력 선발 여부, 지원자에게 요구하는 최소한의 조건, 급여 등의 조건들을 모두 기입한 후에 공지할 수 있다. 이렇게 작성한 항목들은 검색기능에 사용된다.

## **II. Project / Study Recruit**

프로젝트나 스터디를 함께 진행할 팀원을 구하는 공지글을 등록하는 기능이다. 글을 올리기 전에 이 글을 등록하는 사람의 간단한 신분 정보와 랭킹 점수 제공을 위한 정보 제공 동의 여부를 선택한다. 글을 올리는 사용자는 양식에 맞춰 프로젝트/스터디 주제, 기간, 모집인원, 지원 요건 등을 모두 기입한 후에 공지할 수 있다. 이렇게 작성한 항목들은 검색기능에 사용된다.

## **III. Announcement**

Connect-U 서비스의 사용자인 학교 교직원이 학교나 학과에서 진행중인 프로그램 및 공지사항 등을 공지할 수 있는 기능이 있다. 프로그램 인원 모집글을 작성할 때에는 프로그램 이름, 진행 기간, 지원 대상, 모집 인원, 지원 조건, 프로그램 진행 시 주어지는 혜택 등을 모두 기입한 후에 공지할 수 있다. 이렇게 작성한 항목들은 검색기능에 사용된다.

## **E. Job Search**

검색은 사용자가 특정 키워드를 사용하여 정보를 검색할 수 있도록 하고 사용자가 관심있는 모집글만 찾을 수 있도록 하여 시간을 절약할 수 있는 요구사항이다. 웹사이트의 모든 페이지에서 사용자는 특정 키워드를 입력하고 입력 키워드와 관련된 연구실, 프로젝트, 스터디, 진행중인 프로그램을 찾을 수 있다. 이때 사용한 키워드는 최근 검색 기록에 저장한다.

## **F. Recruit Recommendation**

사용자가 마이페이지에 등록한 관심분야와 사용자의 최근 검색 목록을 바탕으로 추천 알고리즘을 이용하여 사용자의 요구 사항과 비슷한 일자리를 추천한다. 사용자의 마음에 들지 않거나 원했던 모집글인 경우 '관심 없음'표시로 다음 번의 추천 알고리즘에 반영이 되도록 한다.

## **G. Workforce Matching**

사용자가 일자리 검색과 추천을 통해 일자리에 지원한다. 이때, 지원자는 본인의 간단한 신상정보와 랭킹정보를 제공한다는 정보 제공 동의를 해야 한다. 모집글을 등록한 등록자는 지원자들의 명단을 list의 형태로 확인할 수 있다. 이 list 중 한 사람을 클릭하면 지원자의 정보를 확인할 수 있고, 이를 확인 후 수락 여부를 결정한다. 등록자가 수락한다면 매칭이 완료되고, 지원자 모두에게 매칭 성공/실패 여부가 알려진다.

## **H. Notification**

마이페이지 환경 설정 메뉴의 알림 설정에는 이전에 설정한 관심분야 키워드 알림, 타인의 평가 알림, 관심있는 프로젝트의 지원 마감 임박 알림이 있다. 방해금지 시간 설정은 사용자가 직접 시간대를 설정하여 지정된 시간에는 알림이 오지 않도록 한다. 지원자가 어떤 모집글에 지원한 경우, 이 글을 등록한 사용자에게 알림 메시지에 보내진다. 그런데 이 지원자가 블랙리스트로 등록되어 있는 경우 모집글을 등록한 사용자에게 추가적으로 경고 메시지가 보내진다. 모집자가 지원자의 지원 여부를 확인하면 지원자에게도 모집자가 본인의 지원 여부를 확인하였다는 알림 메시지가 보내진다. 사용자에게 알림된 모든 메시지는 알림함에 보관된다.

## **I. User Evaluating System**

일자리 중개 시에 발생할 수 있는 위험을 줄이기 위해 이전에 진행했던 프로젝트나 프로그램을 통해 사용자들 사이에서 평가할 수 있는 시스템이다. 각 사용자는 같은 같이 일을 했던 다른 사용자들을 최대 5점 만점으로 평가하여 한 사용자의 평가 점수는 여태 다른 사용자들이 평가한 점수들의 평균으로 매겨진다. 모집글에 지원한 사용자와 모집을 하는 사용자는 서로 평가 점수를 확인할 수 있다.



## J. Declaration System

일자리 중개 시에 발생할 수 있는 위험을 줄이기 위해 시스템을 불건전하게 악용하거나 상습적으로 다른 사람에게 피해를 주는 사용자는 신고를 할 수 있다. 불건전하거나 서비스와는 어울리지 않는 모집글을 올리는 경우, 매칭 이후 상습적으로 연락이 두절되어 프로그램 진행에 차질이 생겨 다른 사용자들에게 피해를 주는 경우 신고 버튼을 통해 신고할 수 있다. 신고 버튼을 클릭하면 신고 유형을 체크하고 신고 사유를 작성한다. 이때 신고 사유는 합당한 사유여야 하며, 관리자는 이 내용을 검토하여 적절한 신고인지 판단한 후 적절하다고 판단하면 신고 당한 사용자를 블랙리스트에 등록한다.

## K. Blacklist and Suspension

일자리 중개 시에 발생할 수 있는 위험을 줄이기 위해 시스템을 불건전하게 악용하거나 상습적으로 다른 사람에게 피해를 주는 사용자를 신고하여 관리자의 판단 후 블랙리스트에 등록할 수 있다. 이 사용자는 신고 수위에 따라 적절한 기간의 이용 정지 처분을 내린다. 첫 정지 시 7일, 두번째 정지 시 30일, 세번째 정지 시 영구 정지 처분이 내려진다. 30일 이하의 이용 정지를 당한 사용자는 정지된 기간동안 게시글 검색 및 확인은 가능하지만 지원 및 등록이 불가하고 추천 시스템 또한 제공되지 않는다. 영구적으로 계정 정지 처분을 받은 사용자는 로그인 시 로그인할 수 없다는 내용의 팝업창이 뜨며 로그인이 불가능하여 서비스 이용이 불가능하다.

# 4.2 Non-functional Requirements

## A. Product Requirements

### 1) Usability

본 시스템의 이용 층은 대학생, 대학 교수, 교직원 등으로 다양한 연령대의 사용자가 존재한다. 따라서 누구나 쉽게 이용할 수 있고, 필요한 메뉴와 정보를 한눈에 알아볼 수 있을 만한 사용자 인터페이스를 제공해야 한다. 이에 따라 광고를 최대한 줄이고, 본 시스템에서 제공하는 기능인 일자리 등록 및 검색, 추천 기능 등 필요한 기능 위주의 깔끔한 인터페이스를 디자인한다.

### 2) Security

본 시스템에서 수집한 개인 정보를 외부에 유출되지 않도록 철저히 관리해야 한다. 또한 사용자들 사이에도 일자리 등록 및 지원 시에 서로에게 제공되는 개인 정보는 사용자가 동의한 정보만을 제공해야 한다.

### 3) Reliability

지원자가 시간 낭비를 하지 않도록 이미 마감된 일자리에 대해 빠르고 정확한 변경이 있어야 한다. 또한 사용자가 최근에 검색했던 조건을 가진 일자리가 새로 등록된다면 사용자에게 빠르게 일자리 추천 메뉴를 통해 정보를 전달한다.

### 4) Performance

본 시스템에서 사용되는 일자리 추천 기능은 추천의 정확도가 낮으면 사용자의 이용률에 영향을 줄 수 있기 때문에 최대한 사용자의 요구 사항과 일치하는 데이터를 파악해야 한다. 또한 수많은 이용자 데이터와 일자리 데이터를 효율적으로 관리하여 데이터를 나타내는데 문제가 없어야 한다.

## B. Organizational Requirements

### 1) Development Requirements

본 시스템의 빠른 개발과 배포를 위하여 parallel development를 도입한다. 이에 따라 일자리 추천 등의 알고리즘, 프론트엔드, 백엔드로 역할을 크게 나누어 서로 독립적으로 동시에 개발이 가능하도록 한다. 또한 개발에 있어서 모든 과정과 소스 코드는 github를 사용하여 관리한다.

### 2) Environment Requirements

사용자가 시스템을 이용하기 위해서는 데이터베이스 서버에 접근하여 사용자 정보와 일자리 정보를 받아와야 하므로 모바일 디바이스는 네트워크에 연결되어 있어야 한다. 또한 데이터베이스를 관리하고 디바이스와 연결할 서버는 항상 구동되어 있어야 한다.

## C. External Requirements

### 1) Regulatory Requirements

본 시스템은 회원 가입을 진행하거나 일자리 등록 및 지원 과정에서 개인정보를 수집하고 제공함에 대한 동의를 구해야 하며, 취급 방침에 대한 알리를 제공해야 한다. 또한 관련 약관을 미리 고지하고, 가입 이후에도 약관을 확인할 수 있도록 제공한다.

### 2) Safety Requirements

학교 관계자가 아닌 외부 인원이 회원 가입을 할 경우 인력 중개라는 목적을 가진 본 시스템에서 큰 문제가 발생할 수 있다. 따라서 학교 소속 인증 절차에서 허점이 없도록 철저히 진행해야 한다. 또한 학교 내부 인원이라 하더라도 불건전한 목적으로 시스템을 악용하거나 타인에게 피해를 주는 이용자는 블랙 리스트로 등록하여 시스템을 관리한다.

## 5. System Architecture

이 챕터에서는 시스템의 전체적인 아키텍처를 highly abstract level에서 설명합니다. 또한 시스템 구성 요소 간의 상호 작용을 나타내고, 각 상호 작용은 다이어그램으로 설명되어 있습니다.

### 5.1 System Overview

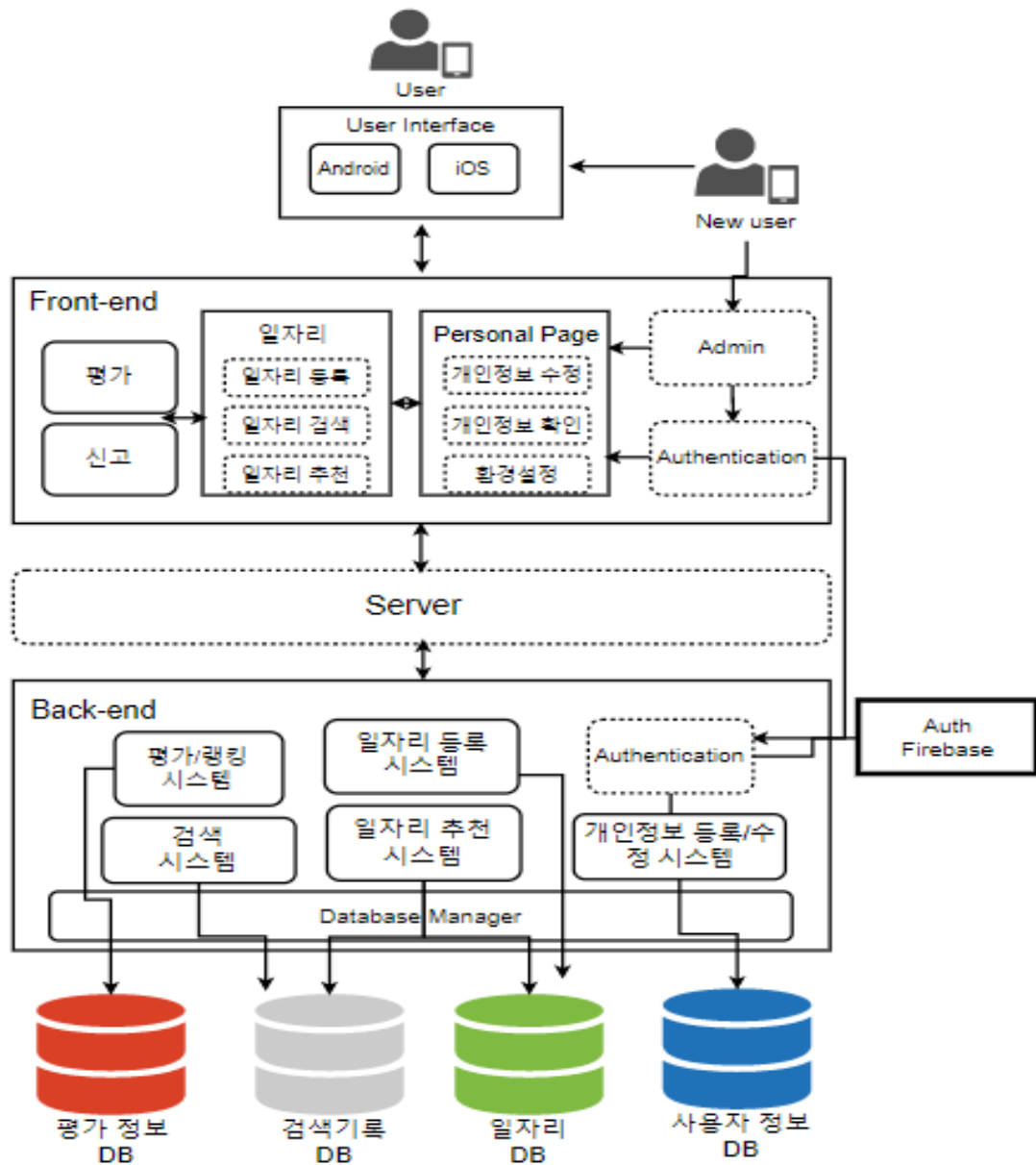


Diagram1: System Overview Diagram

전반적인 시스템 구조를 살펴보면 Front-end에서 다양한 사용자의 요청에 따라 Back-end에서 그에 따른 요청을 수행하고 그에 따른 결과를 데이터베이스에 저장한다.

## 5.2 Front-end Architecture

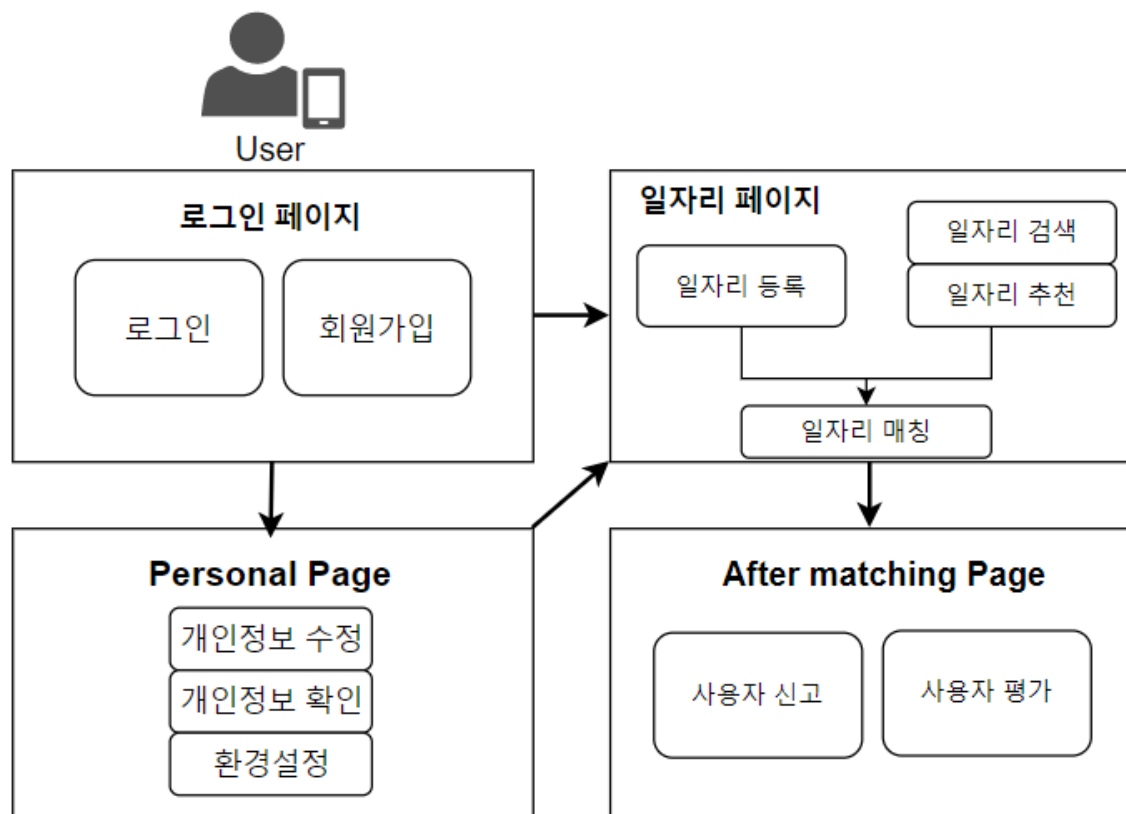
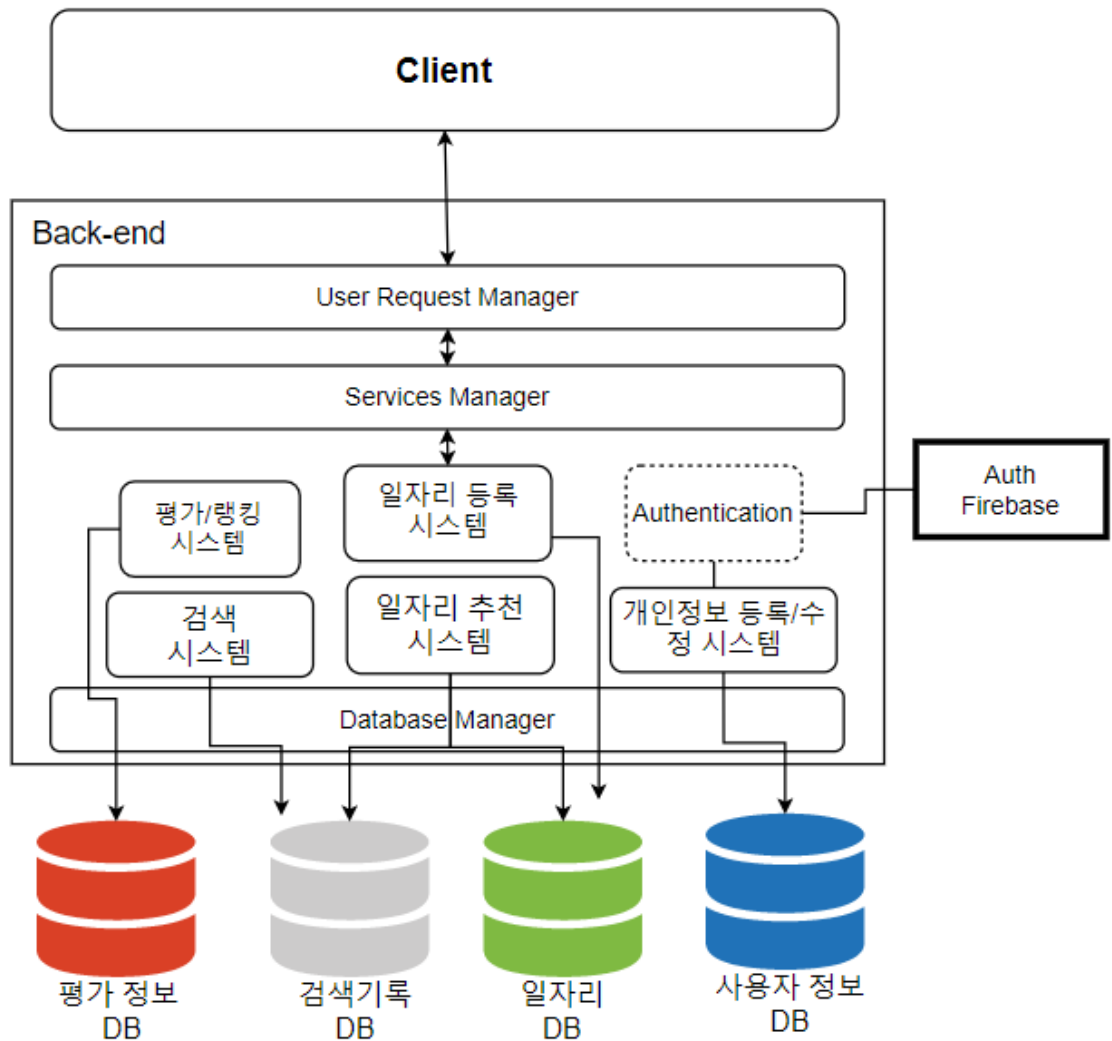


Diagram2: Front-end Architecture

Front-end에서 사용자는 user interface를 통해 시스템과 상호작용을 할 수 있다. 사용자가 어플을 들어가면, 로그인과 회원가입 화면이 나타나고, 로그인을 하면 Personal Page 또는 일자리 페이지로 들어 갈 수 있다. 일자리 페이지에서 일자리를 등록하거나 검색 또는 추천을 통해 일자리가 매칭이 되면, 그 후에는 등록자와 지원자 간의 상호 평가가 가능하고, 특정 사용자에 대한 신고가 가능하다.

### 5.3 Back-end Architecture



**Diagram 3: Back-end Architecture**

Back-end에서는 사용자의 다양한 요청에 맞는 일을 수행한다. 소속 인증 요청이 주어지면 Auth Firebase를 통해 e-mail로 인증을 하고, 새로운 사용자 정보가 생성되거나 기존 사용자의 정보가 수정되면, 개인정보 등록/수정 시스템을 통하여 변경되고 변경된 데이터는 사용자 정보 DB에 업데이트 된다. 사용자가 검색 요청을 하면 검색 시스템이 조건에 맞는 다양한 일자리를 검색해 화면에 나타내 주고, 검색 기록 DB를 업데이트 한다. 새로운 일자리가 등록되면 일자리 등록 시스템은 일자리 DB에 새로운 정보를 등록하고 평가/랭킹 시스템이 가능하게 된다. 사용자가 한 평가에 대한 정보는 평가/랭킹 시스템에 의하여 평가 정보 DB에 저장되고, 시스템은 요청 시

사용자의 랭킹을 계산한다.

## 5.4 Job Recommendation System

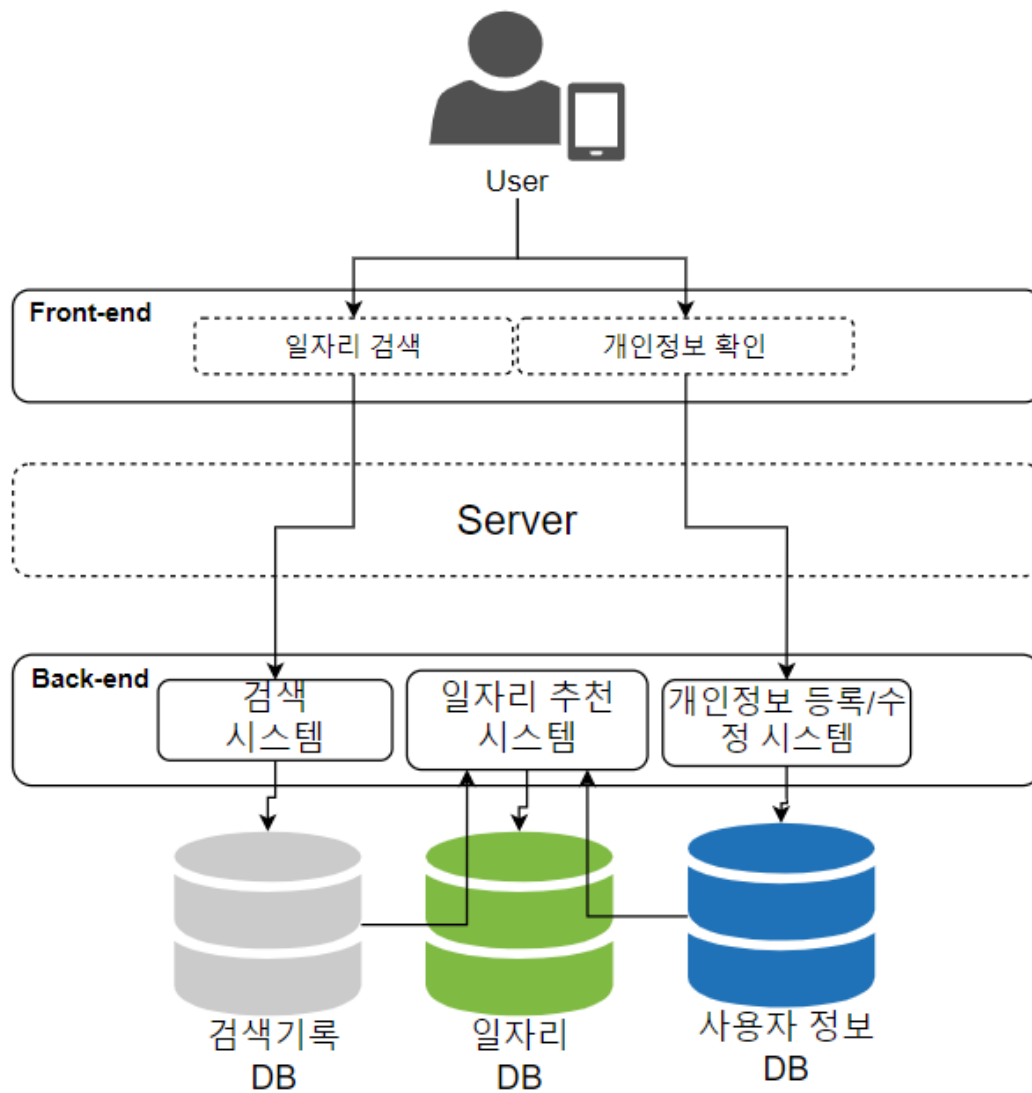


Diagram 4: Job Recommendation System Architecture

일자리 추천 시스템은 특정 사용자의 개인 정보와 일자리 검색 기록을 사용한다. 일자리 검색기록과 개인 정보 기록이 DB에 저장되면, 일자리 추천 시스템은 그 DB를 이용하여 사용자에게

게 알맞은 일자리를 추천한다.

## 6. System Requirements specification

이 챕터에서는 User Requirements Definition에 간략히 서술되어 있는 요구 사항을 바탕으로, 기능적 요구 사항(Functional Requirements)과 비기능적 요구 사항(Non-functional Requirements), 기타 요구 사항을 상세히 서술한다. 또한 시스템의Scenario를 제시한다.

### 6.1 Functional Requirements

#### A. Sign Up

Name	회원 가입
Description	학교 내부에서 일자리를 중개하는 시스템 특성에 따라 보안을 위하여 회원 가입을 요구한다. 최초 이용 시에 한 번만 진행한다.
Inputs	User Input : *이름, *ID, *password, *소속 대학, 소속 학과, *이메일, *연락처, 주소 ( *필수 항목)
Action	1. 사용자의 기본적인 정보를 입력 받은 후 DB 에 저장한다. 2. 사이트에서 입력 받은 이메일로 인증 메일을 보낸다. 3. 이메일 확인을 통한 본인 인증이 성공하면 회원 가입이 완료되며 입력했던 ID 와 password 를 통해 로그인 할 수 있게 된다.
Outputs	Database 에 계정 정보가 저장된다. 로그인 화면을 띄운다.
Destination	사용자의 화면, Database



Requirements	본인 인증 메일 전송 후 7 일 이내에 본인 인증을 완료하지 않는다면 Database 에서 계정 정보를 삭제한다.
--------------	---

## B. Sign In & Certification of Belonging

Name	로그인 및 소속 인증
Description	학교 내부에서 일자리를 중개하는 시스템 특성에 따라 보안을 위하여 회원 가입 절차를 거친 인원만 시스템을 이용할 수 있도록 로그인을 요구한다. 회원 가입 이후 최초 한 번은 학교 내부 인원임을 인증하는 인증 절차를 거쳐야 한다.
Inputs	User Input : ID / password Database : ID / password
Action	1. ID와 password를 입력한다. 2. Database에 저장되어있는 계정 정보와 동일한지 비교 후 동일할 경우에만 로그인이 성공하고 다를 경우에는 계정 정보와 다른 입력을 받았다는 메시지를 출력한다. 3. 회원 가입 이후 최초 한 번은 소속 인증 절차를 거쳐야 한다. 인증 배너를 통하여 학생증 등 소속을 인증할 수 있는 자료를 업로드한다. 4. 관리자가 자료를 확인한 후 인증이 완료되면 시스템의 모든 기능을 이용할 수 있게 된다.
Outputs	로그인에 성공하면 사용자를 위한 페이지를 보여준다. 회원 가입 이후 소속 인증이 되지 않은 회원에게는 인증 배너를 띄운다.
Destination	사용자의 화면 , Database
Requirements	회원 가입 이후 3 개월 이내에 소속 인증을 완료하지 않는다면 휴면 계정으로 등록한다. 이후 계정 활성화를 위해서는 사용자가

	본인 인증 절차를 다시 진행해야 한다.
--	-----------------------

### C. My Page

Name	마이 페이지
Description	<p>마이 페이지는 개인정보 수정 메뉴와 나의 정보 확인 메뉴, 환경 설정 메뉴로 총 3 개의 메뉴로 구성된다.</p> <p>먼저 개인정보 수정 메뉴를 통해 회원 가입 시 입력했던 정보를 수정하고, 추가 정보를 입력 할 수 있다. 또한 관심 분야와 원하는 일자리의 조건을 등록 할 수 있다.</p> <p>나의 정보 확인 메뉴로 지금까지 참여한 일자리, 등록된 일자리, 본인의 랭킹 점수를 확인 할 수 있다.</p> <p>환경 설정 메뉴에서는 알림 설정, 권한 설정 등 시스템의 기본적인 설정을 할 수 있다.</p>
Inputs	User Input : 본인 확인을 위한 이메일 인증, 수정할 개인정보, 추가적으로 등록 할 정보 등.
Action	<p>&lt;개인 정보 수정&gt;</p> <ol style="list-style-type: none"> <li>1. 먼저 개인정보 수정을 위해서는 본인 확인 절차를 거쳐야 한다.</li> <li>2. 이메일을 통한 본인 확인이 완료되면 수정할 정보를 입력할 수 있다.</li> <li>3. 관심 분야와 원하는 일자리 조건을 추가적으로 등록 할 수 있다.</li> </ol> <p>&lt;나의 정보 확인&gt;</p> <ol style="list-style-type: none"> <li>1. 지금까지 참여한 일자리, 등록된 일자리 등을 확인할 수 있다.</li> <li>2. 어플리케이션을 통하여 참여한 일자리가 있다면, 일자리를 등록한 사용자가 평가한 나의 랭킹을 확인 할 수 있다.</li> </ol> <p>&lt;환경 설정&gt;</p> <ol style="list-style-type: none"> <li>1. 알림 설정, 테마, 언어, 권한 설정, 메뉴 바 설정 등의 기본 설정을 변경 할</li> </ol>

	<p>수 있다.</p> <p>2. 시스템의 약관 정보를 확인 할 수 있다.</p>
Outputs	<p>사용자가 변경한 개인 정보를 Database 에 업데이트한다.</p> <p>환경 설정을 변경했다면 변경된 환경 설정을 반영한다.</p>
Destination	<p>사용자의 화면 , Database</p>
Requirements	<p>없음.</p>

#### D. Recruit Enrollment

Name	일자리 등록
Description	사용자가 일자리를 등록한다.
Inputs	User Input : 등록할 일자리의 카테고리( 연구실, 프로젝트, 아르바이트, 설문 등), 일자리를 등록하는 사람의 간단한 신분 정보, 일자리 조건( 시급, 기간 등 ), 타 학교 게시판에 게시 여부, 지원자 자격 명시
Action	<ol style="list-style-type: none"> <li>1. 일자리 등록 메뉴를 들어간다.</li> <li>2. 등록할 일자리의 카테고리를 등록한다.</li> <li>3. 일자리를 등록하는 사람의 간단한 신분 정보와 랭킹 점수 제공을 위한 정보 제공 동의 여부를 선택한다.</li> <li>4. 일자리 조건을 입력한다.</li> <li>5. 타 학교 게시판에 게시하여 타 학교 인원의 지원을 받을 것인지 등록한다.</li> <li>6. 지원자에게 요구하는 최소 조건을 명시한다.</li> <li>7. 입력이 모두 완료되면 DB에 일자리 정보를 등록한다.</li> </ol>
Outputs	<p>사용자의 입력이 모두 완료되면 Database 에 일자리 정보를 등록한다.</p>
Destination	<p>사용자의 화면 , Database</p>

Requirements	블랙 리스트에 등록된 사용자는 일자리 등록을 할 수 없다.
--------------	----------------------------------

### E. Job Search

Name	일자리 검색
Description	다른 사용자가 시스템에 등록한 학교 내부의 일자리 정보를 검색한다.
Inputs	User Input : 일자리 카테고리, 타 학교 여부, 일자리 조건, 지원자 자격 등 Database : 등록되어있는 일자리 정보
Action	1. 일자리 검색 메뉴에 들어간다. 2. 원하는 일자리의 카테고리, 타 학교 여부, 일자리 조건, 지원자 자격 등을 선택하고 검색 버튼을 누른다. 3. 데이터베이스에서 사용자가 입력한 정보를 바탕으로 해당하는 일자리 정보를 받아온다. 4. 사용자 화면에 데이터베이스에서 받아온 정보를 리스트 형태로 출력한다. 5. 만족하는 일자리가 없다면, 일치하는 항목이 없다는 메시지를 사용자의 화면에 출력한다.
Outputs	사용자의 검색 입력에 맞는 정보를 데이터베이스에서 찾아 화면에 출력한다.
Destination	사용자의 화면 , Database
Requirements	없음.

## F. Recruit Recommendation

Name	일자리 추천
Description	마이페이지의 개인정보에 등록한 추가 정보와, 최근 검색 목록을 바탕으로 사용자의 요구 사항과 비슷한 일자리를 추천한다.
Inputs	Database : 마이페이지에 등록한 관심 분야 및 원하는 일자리 조건, 사용자가 검색한 최근 기록
Action	<ol style="list-style-type: none"> <li>1. 사용자는 일자리 추천 메뉴를 통해 알고리즘이 추천하는 일자리를 소개받을 수 있다.</li> <li>2. Database에 등록된 사용자의 관심 분야 및 원하는 일자리 조건을 충족하는 일자리가 등록되어있는지 검색하고, 존재한다면 추천 메뉴에 출력한다.</li> <li>3. 사용자의 최근 검색 기록을 데이터베이스에 저장하고, 최근 검색 기록에 있는 조건을 만족하는 새로운 일자리가 등록된다면 추천 메뉴에 출력한다.</li> </ol>
Outputs	데이터베이스에 저장된 사용자의 정보를 바탕으로 조건에 맞는 일자리를 검색해 출력한다.
Destination	사용자의 화면 , Database
Requirements	블랙 리스트에 등록된 사용자에게는 추천 일자리가 뜨지 않는다.

## G. Workforce Matching

Name	일자리 매칭
Description	일자리 검색 / 추천을 통해 일자리에 지원하고 일자리를 등록한 사용자는 지원자 정보를 확인한 후 수락하면 매칭이 완료된다.
Inputs	User Input : 지원자의 지원 서류, 등록자의 수락 여부

Action	1. 지원자는 일자리 검색 및 추천을 통해 찾은 일자리에 지원한다. 2. 지원을 하기 위해서는 본인의 간단한 신상 정보 및 랭킹 정보를 제공한다는 정보 제공 동의를 해야 한다. 3. 등록자는 지원자의 정보를 확인한 후 수락 여부를 결정한다. 4. 등록자가 수락한다면 매칭이 완료되고, 수락하지 않는다면 매칭이 실패했다는 메시지를 지원자에게 알림으로 출력한다. 5. 매칭이 완료된 후 2주 동안 일자리 정보는 데이터베이스에 유지되므로, 매칭이 취소된다면 등록자는 2주 이내에 재 등록 신청을 통해 일자리를 재 등록 할 수 있다.
Outputs	일자리 매칭 정보를 데이터베이스에 저장한다. 일자리 매칭이 완료되더라도 2 주 동안은 데이터베이스에서 일자리 정보를 삭제하지 않는다.
Destination	사용자의 화면 , Database
Requirements	지원자가 블랙 리스트에 등록되어 있다면 지원 시 등록자에게 경고 메시지가 알림으로 출력 된다.

## H. Notification

Name	알림
Description	일자리 등록 이후 지원자의 지원 여부, 블랙 리스트 사용자 경고 등 알림 메시지를 사용자에게 출력한다.
Inputs	Datebase : 일자리 지원 정보, 블랙 리스트
Action	1. 지원자가 등록한 일자리에 지원한다면 알림을 통해 지원 여부를 메시지로 받을 수 있다. 2. 지원자가 블랙 리스트에 등록 되어 있을 경우 경고 메시지를 받을 수 있다. 3. 마이페이지 환경 설정 메뉴를 통해 알림 설정을 할 수 있다.

Outputs	로그인에 성공하면 유저를 위한 페이지를 보여준다. 회원 가입 이후 소속 인증이 되지 않은 회원에게는 인증 배너를 띄운다.
Destination	사용자의 화면
Requirements	없음.

### I. User Evaluating System

Name	사용자 평가 시스템
Description	시스템의 사용 목적이 인력 중개인 만큼, 위험을 줄이기 위하여 사용자들 사이에 평가 시스템을 도입하여 서로의 평가 점수를 확인 할 수 있도록 한다.
Inputs	User Input : 일자리 매칭 이후 서로에 대한 평가를 진행하여 점수 부여
Action	자리 매칭 이후 지원자와 등록자 모두 서로에 대한 평가를 진행하여 점수를 부여할 수 있다.
Outputs	점수를 데이터 베이스에 등록하고 통계를 개인정보에 출력한다.
Destination	사용자의 화면 , Database
Requirements	없음.

### J. Declaration System

Name	신고 시스템
Description	시스템의 사용 목적이 인력 중개인 만큼, 위험을 줄이기 위하여

	시스템을 불건전하게 악용하거나 상습적으로 다른 사람에게 피해를 주는 사용자를 신고 할 수 있다.
Inputs	User Input : 신고 신청
Action	1. 불건전한 구인 글을 올리거나 시스템을 악용하는 사용자를 발견하면 누구나 신고 버튼을 통해 신고 할 수 있다. 2. 매칭 이후 상습적으로 연락 두절이 되거나 등록자에게 고의로 피해를 주는 사용자를 신고 할 수 있다. 3. 신고 버튼을 클릭하면 신고 내용과 카테고리를 작성 할 수 있다. 4. 관리자는 신고 받은 내용을 검토하여 적절한 신고인지 판단 후 신고 내용이 적절하다면 사용자를 블랙 리스트에 등록한다.
Outputs	데이터베이스에 블랙 리스트에 등록된 사용자를 저장한다.
Destination	사용자의 화면 , Database
Requirements	허위 신고를 반복적으로 하는 사용자는 관리자가 사용자의 이용을 정지 시킬 수 있다.

#### K. Blacklist and Suspension

Name	블랙 리스트 및 이용 정지
Description	사용자 평가에서 최저점을 반복적으로 받거나 다른 사용자에게 신고를 받은 사용자를 블랙 리스트에 등록하고, 이용 정지 처분을 내린다.
Inputs	Database : 사용자 평가 점수 , 신고 내용
Action	1. 사용자 평가에서 지속적으로 최저점을 받거나 신고를 받은 사용자를 관리자의 판단 하에 블랙 리스트에 등록한다. 2. 블랙 리스트에 등록된 사용자는 신고 수위에 따라 적절한 기간의 이용 정지 처분을 내린다.



	3. 이용 정지를 당한 사용자는 게시글 검색 및 일자리 확인은 가능하지만, 일자리 등록 및 지원이 불가능해진다. 일자리 추천 항목 또한 뜨지 않는다. 4. 블랙 리스트에 등록된 사용자가 이용 정지 기간 이후에 또다시 신고를 받는 등 불건전한 이용을 지속한다면 계정을 정지한다. 5. 계정 정지를 당한 사용자는 게시글 검색 및 일자리 확인 또한 불가능하다.
Outputs	블랙 리스트 사용자를 이용 정지 및 계정 정지 시킨다.
Destination	Database
Requirements	없음.

## 6.2 Non-functional Requirements

이 파트에서는 비 기능적 요구 사항에 대해 서술한다. 비 기능적 요구 사항은 Product Requirements, Organizational Requirements, External Requirements로 나누어 설명한다.

### A. Product Requirements

#### 1) Usability

본 시스템의 이용 층은 대학생, 대학 교수, 교직원 등으로 다양한 연령대의 사용자가 존재한다. 따라서 누구나 쉽게 이용할 수 있고, 필요한 메뉴와 정보를 한 눈에 알아볼 수 있을 만한 사용자 인터페이스를 제공해야 한다. 이에 따라 광고를 최대한 줄이고, 본 시스템에서 제공하는 기능인 일자리 등록 및 검색, 추천 기능 등 필요한 기능 위주의 깔끔한 인터페이스를 디자인한다.

#### 2) Security

본 시스템에서 수집한 개인 정보를 외부에 유출되지 않도록 철저히 관리해야 한다. 또한 일자리 등록 및 지원 시에 서로에게 제공되는 개인 정보는 사용자가 동의한 정보 만을 제공해야 한다.

#### 3) Reliability

지원자가 시간 낭비를 하지 않도록 이미 마감된 일자리에 대해 빠르고 정확한 변경이 있어야 한다. 또한 사용자가 최근에 검색했던 조건을 가진 일자리가 새로 등록된다면 사용자에게 빠르게 일자리 추천 메뉴를 통해 정보를 전달한다.

#### 4) Performance

본 시스템에서 사용되는 일자리 추천 기능은 추천의 정확도가 낮으면 사용자의 이용률에 영향을 줄 수 있기 때문에 최대한 사용자의 요구 사항과 일치하는 데이터를 파악해야 한다. 또한 수많은 이용자 데이터와 일자리 데이터를 효율적으로 관리하여 데이터를 나타내는데 문제가 없어야 한다.

### B. Organizational Requirements

#### 1) Development Requirements

본 시스템의 빠른 개발과 배포를 위하여 parallel development를 도입한다. 이에 따라 일자리 추천 등의 알고리즘, 프론트엔드, 백엔드로 역할을 크게 나누어 서로 독립적으로 동시에 개발이 가능하도록 한다. 또한 개발에 있어서 모든 과정과 소스 코드는 github를 사용하여 관리한다.

#### 2) Environment Requirements

사용자가 시스템을 이용하기 위해서는 데이터베이스 서버에 접근하여 사용자 정보와 일자리 정보를 받아 와야 하므로 모바일 디바이스는 네트워크에 연결되어 있어야 한다. 또한 데이터베이스를 관리하고 디바이스와 연결할 서버는 항상 구동되어 있어야 한다.

### C. External Requirements

#### 1) Regulatory Requirements

본 시스템은 회원 가입을 진행하거나 일자리 등록 및 지원 과정에서 개인정보를 수집하고 제공함에 대한 동의를 구해야 하며, 취급 방침에 대한 알리를 제공해야 한다. 또한 관련 약관을 미리 고지하고, 가입 이후에도 약관을 확인 할 수 있도록 제공한다.

## 2) Safety Requirements

학교 관계자가 아닌 외부 인원이 회원 가입을 할 경우 인력 중개라는 목적을 가진 본 시스템에서 큰 문제가 발생할 수 있다. 따라서 학교 소속 인증 절차에서 허점이 없도록 철저히 진행해야 한다. 또한 학교 내부 인원이라 하더라도 불건전한 목적으로 시스템을 악용하거나 타인에게 피해를 주는 이용자는 블랙 리스트로 등록하여 시스템을 관리한다.

## 6.3 Scenario Examples

### A. Login Scenario

#### 1) Initial Assumption

사용자는 이미 회원 가입이 완료되어 있으며, 앞서 회원 가입 시 설정한 ID와 password를 이용하여 시스템에 로그인 하고자 한다.

#### 2) Normal flow of events

로그인 페이지를 통해 사용자가 ID와 password를 입력하고 로그인 버튼을 누르면, 시스템은 Database에 저장되어 있는 회원 정보와 입력한 정보를 대조하여 해당 정보들이 일치하는지 확인한다.

#### 3) What can go wrong

입력한 정보가 기존에 등록된 ID와 password와 일치하지 않는다면 올바른 정보라는 메시지를 출력하고 입력한 정보를 초기화 하여 사용자가 올바른 회원 정보를 입력하도록 한다.

#### 4) System state on completion

올바른 ID와 password가 입력되었다면 사용자는 시스템에 로그인 되어 메인 페이지로 이동한다.

## **B. Search Scenario**

### **1) Initial Assumption**

사용자가 검색하고자 하는 일자리의 카테고리 정보들을 입력하고 해당 조건에 부합하는 일자리를 검색하고자 한다.

### **2) Normal flow of events**

일자리 검색 메뉴에 들어간 후 사용자가 검색하고자 하는 일자리의 내용과 일치하도록 카테고리, 시급, 근무 기간, 타 학교 여부, 지원자 자격 조건을 검색 필터로 설정한 후 검색 버튼을 누른다. 이 때 검색될 일자리들의 정렬을 최신 순, 시급 순, 조회수 순 등의 순서로 설정할 수 있으며, 기본 설정은 최신 순으로 한다. 시스템은 입력된 정보와 데이터베이스를 대조하여 일치하는 일자리를 불러온다.

### **3) What can go wrong**

아무런 검색 조건 없이 검색 버튼을 누르면 데이터 베이스에 등록되어 있는 모든 일자리 정보를 검색한다. 데이터베이스에 조건과 일치하는 정보가 등록되어 있지 않다면, 어떠한 정보도 출력하지 않고 해당하는 정보가 존재하지 않는다는 메시지를 출력한다.

### **4) System state on completion**

검색 조건과 일치하는 일자리 정보들을 데이터 베이스에 불러와 정렬 조건에 맞게 사용자 화면에 출력한다.

## **C. Registration Scenario**

### **1) Initial Assumption**

사용자가 일자리를 등록하여 지원자를 모집하려고 한다.

## 2) Normal flow of events

일자리 등록 메뉴에 들어간 후 사용자가 등록하고자 하는 일자의 내용과 일치하도록 카테고리, 시급, 근무 기간, 타 학교 여부, 지원자 자격 조건을 설정한 후 등록 버튼을 누른다. 지원자에게 제공될 간단한 신상 정보 제공에 동의한다.

## 3) What can go wrong

사용자가 블랙 리스트에 등록되어 있을 경우에는 일자리 등록이 불가능하다. 사용자가 일자리 등록 메뉴에 들어간다면 일자리 등록이 불가능하다는 메시지를 출력하고 메인 화면으로 돌아간다.

## 4) System state on completion

사용자가 일자를 등록한다면 데이터베이스에 일자리 정보가 등록되고, 일자리 검색 메뉴에서 해당 정보를 열람할 수 있다.

# D. Recommendation Scenario

## 1) Initial Assumption

사용자가 추가적으로 등록한 관심 분야, 원하는 일자리 조건과 최근 검색 기록 등을 통해 일자를 추천 받고자 한다.

## 2) Normal flow of events

데이터 베이스에서 사용자의 관심 분야, 일자리 조건, 최근 검색 기록과 일치하는 일자리 정보를 검색해 사용자에게 추천 일자리로 제공한다.

## 3) What can go wrong

사용자가 등록한 추가 정보가 없거나 최근 검색 기록이 없을 경우 일자리 추천 메뉴에는 아무런 정보도 출력 되지 않는다. 대신 추천 정보가 없다는 문구와 함께 추가 정보를 등록하면 일자리 추천을 받을 수 있다는 안내 메시지를 출력

한다.

#### 4) System state on completion

사용자가 등록한 추가 정보와 검색 기록을 바탕으로 적절한 일자리 정보를 데이터베이스에서 검색하여 출력한다.

### E. Matching Scenario

#### 1) Initial Assumption

사용자(등록자)가 등록한 일자리에 다른 사용자(지원자)가 지원을 하려 한다.

#### 2) Normal flow of events

지원자는 일자리 검색 / 추천 메뉴를 통해 찾은 일자리에 지원한다. 지원서에 지원자의 간단한 신상 정보를 작성한 후 제출한다. 지원자는 등록자에게 개인정보가 제공됨을 동의한다.

등록자는 지원자의 정보를 확인한다. 사용자 평가 시스템으로 책정된 평가 점수와 신상 정보를 바탕으로 지원의 수락 여부를 결정한다. 등록자가 지원을 수락한다면 일자리 매칭이 완료된다.

#### 3) What can go wrong

일자리 매칭이 된 후 얼마 지나지 않아 계약이 결렬된 경우 등록자는 매칭 직후 2주 이내에 일자리를 재 등록할 수 있다.

#### 4) System state on completion

등록자가 등록한 일자리에 지원자가 지원을 한 후 일자리 매칭이 완료된다.

## F. Rating Scenario

### 1) Initial Assumption

일자리 매칭이 완료되고 실제로 일을 진행한다.

### 2) Normal flow of events

실제로 일을 진행한 이후 지원자와 등록자는 서로를 평가할 수 있다. 불성실하거나, 처음에 제시한 조건과 다른 일자리를 제공하는 사용자들에게 낮은 점수를 부여하고, 성실한 사용자나 좋은 일자리를 제공하는 사용자들에게는 높은 점수를 부여한다.

### 3) What can go wrong

상대방의 평가를 고의로 낮게 하는 경우가 있을 수 있다. 이러한 경우는 상대방의 신고를 통해 관리자가 인지할 수 있고, 신고가 누적된다면 블랙 리스트에 등록하여 시스템을 악용하지 못하도록 막는다.

### 4) System state on completion

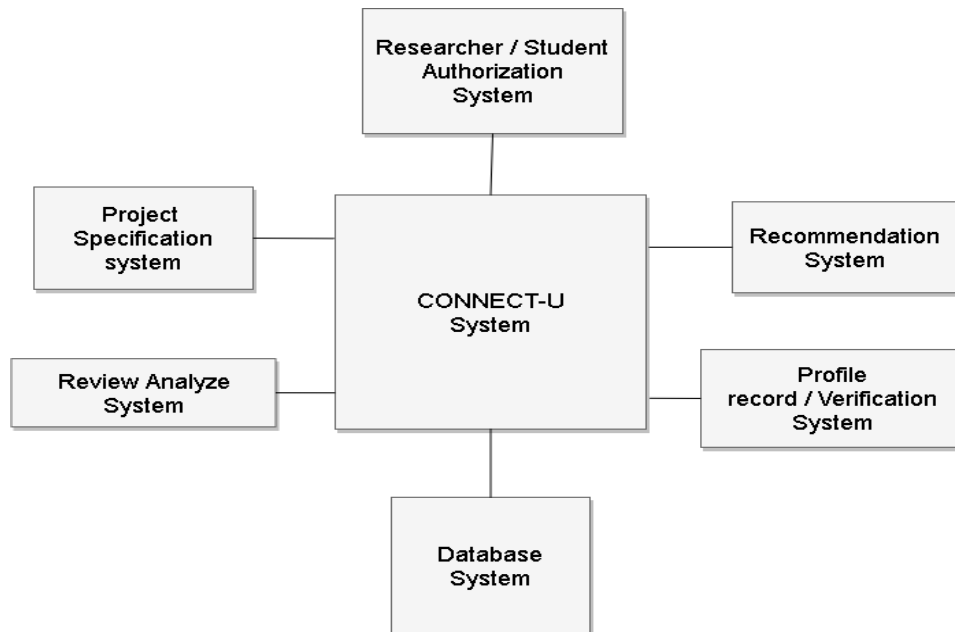
사용자 평가 시스템을 통하여 서로 좋은 인력과 좋은 일자리를 찾을 수 있다.

## 7. System Models

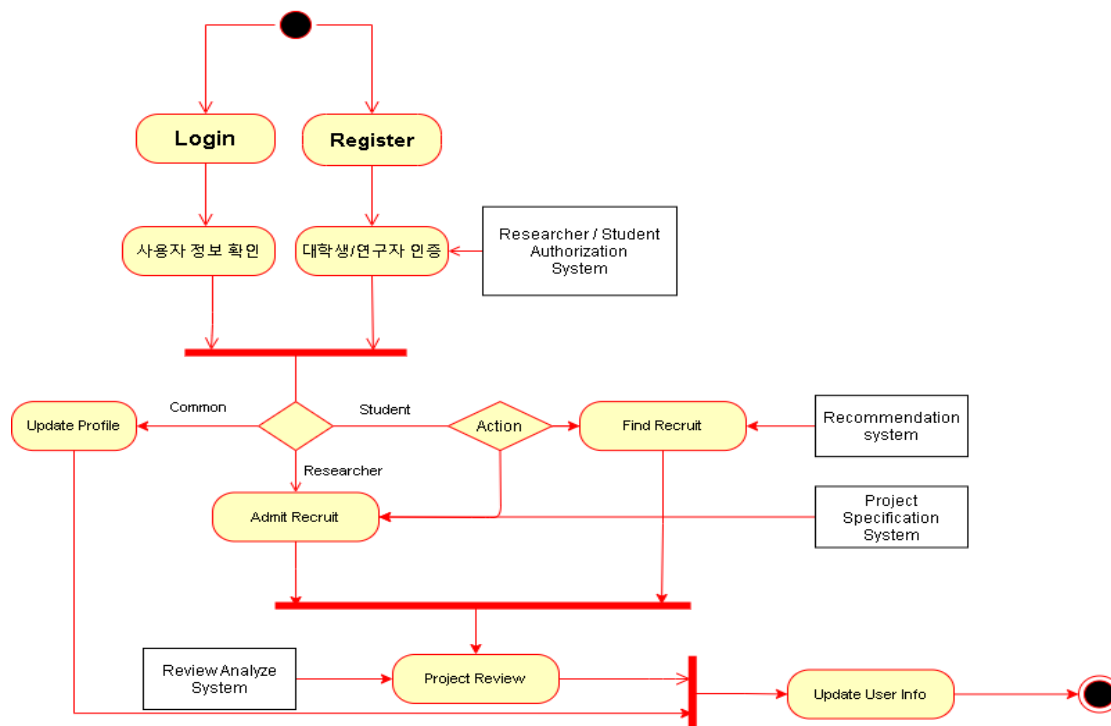
이번 System Models 챕터에서는 각 시스템 컴포넌트와 전체 시스템, 그리고 시스템을 둘러싼 환경 사이의 관계를 여러 가지 Diagram을 통해 기술한다.

## 7.1 Context Models

### A. Context Diagram



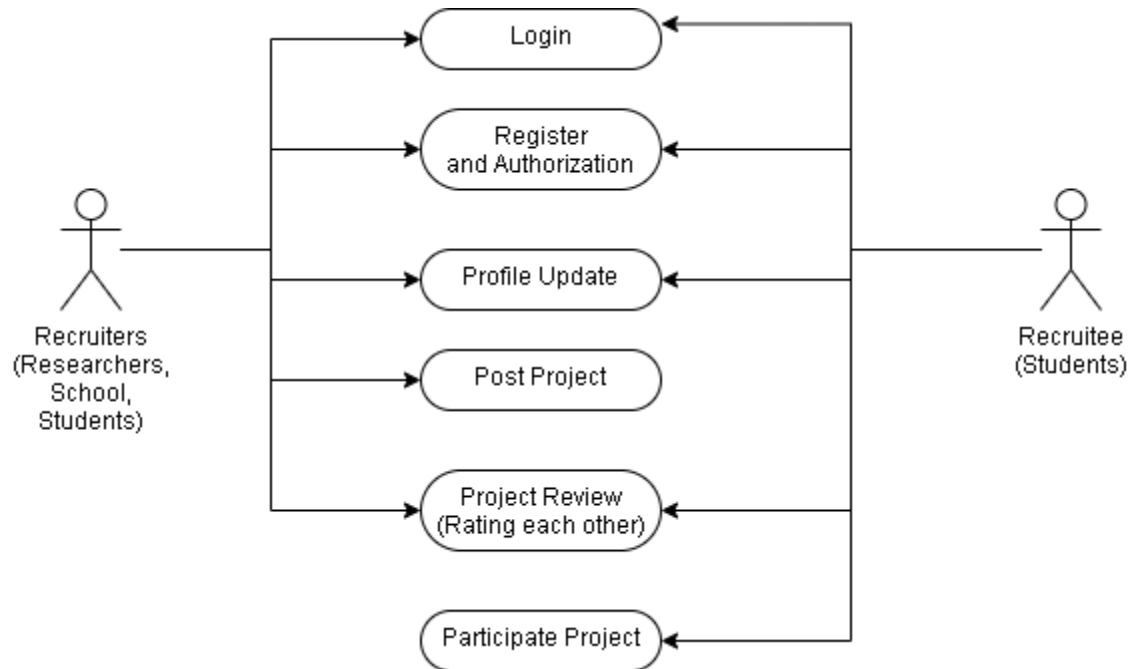
### B. Process Diagram





## 7.2 Interaction Models

### A. Use case Diagram



### B. Tabular Description of Use case Diagram

#### 1) Login

Use case	Login
Actor	Recruiters, Recruitees
Description	접속한 사용자가 시스템에 등록되어 있는 유저 정보와 일치하는지 확인한다.
Trigger	사용자가 이메일과 비밀번호를 입력한 뒤 로그인 버튼을 누른다.
Success Response	사용자의 이메일이 데이터베이스에 존재하며, 비밀번호가 데이터베이스에 저장된 값과 일치하는 경우 사용자 정보와 접속 권한을 포함한 객체를 반환한다.
Failure Response	사용자 이메일이 데이터베이스에 없는 경우 회원 가입을 유도하는 팝업을 띄운다. 비밀번호가 일치하지 않을 경우 오류 메시지를 띄운다.

#### 2) Register and Authorization

Use case	Register and Authorization
Actor	Recruiters, Recruitees

Description	회원의 정보를 최초로 등록하는 회원 가입 절차로, 최초 이용 시에 한 번만 진행. 교내 일자리를 중개하기 때문에 교내 이메일을 통한 인증을 최초 한 번 진행한다.
Trigger	사용자가 회원 가입 버튼을 누른다.
Success Response	사용자의 아이디와 일치하는 아이디가 존재하지 않는 경우, 사용자의 비밀번호가 비밀번호 요구사항에 일치하는 경우, 소속 대학과 학과정보, 이메일을 통한 인증 및 연락처, 주소에 대한 항목이 모두 충족된 경우
Failure Response	각 항목 중 비어 있는 항목에 대해 비어 있음을 사용자에게 알리고, 각 항목의 요구사항을 만족하지 않는 경우 해당 항목을 알려준다. 인증이 실패한 경우 재인증을 요청한다.

### 3) Profile Update

Use case	Profile Update
Actor	Recruiters, Recruitees
Description	접속한 사용자가 시스템에 등록되어 있는 사용자의 새로운 정보나 변경된 정보를 갱신하는 과정이다.
Trigger	사용자가 Career Update를 클릭한다.
Success Response	개인정보 수정 메뉴를 통해 회원 가입 시 입력했던 정보를 수정하고, 추가 정보를 입력할 수 있다. 추가 정보에는 관심 분야와 원하는 일자의 조건을 등록할 수 있다. 나의 정보 확인 메뉴로 지금까지 참여한 일자리, 등록한 일자리, 본인의 랭킹 점수를 확인할 수 있다. 환경 설정 메뉴에서는 알림 설정, 권한 설정 등 시스템의 기본적인 설정을 할 수 있다.
Failure Response	-

### 4) Post Project

Use case	Post Project
Actor	Recruiters
Description	사용자가 Project 참여 인원 모집 공고, 추가 researcher 모집 공고, 혹은 일자리 공고를 등록한다.
Trigger	사용자가 올리는 Post의 카테고리, 모집 상세 내용(모집 인원, 주제, 페이, 기간 등) 및 원하는 조건을 작성 후 Post한다.
Success Response	일자리 모든 필수 상세 내용 및 최소 조건이 쓰인 상태에서, 사용

	자가 블랙리스트에 등록되지 않은 상태라면 Recruit 포스트를 등록한다. 만일 recruit의 모든 모집 인원이 가득 찬다면 해당 포스팅은 완료 상태로 처리한다.
Failure Response	사용자가 블랙리스트에 등록된 경우 차단이 풀리는 기한을 팝업으로 띄운다. 포스트에 어떤 필수 상세 내용이나 조건이 쓰이지 않은 상태라면 해당 조건을 채워주라는 팝업창을 띄운다.

#### 5) Participate Project

Use case	Participate Project
Actor	Recruitees
Description	접속한 사용자가 원하는 카테고리의 게시판에 들어가 원하는 프로젝트 혹은 recruit에 참여 신청을 넣는다.
Trigger	사용자가 원하는 게시판에 들어가 원하는 포스팅을 누른 후, 신청하기 버튼을 누른다.
Success Response	사용자는 해당 게시판에서 자신이 중요하다고 여기는 조건을 위주로 게시판의 Post들을 재정렬할 수 있으며, 정렬 후 원하는 Posting에 들어가 참가 신청을 넣는다. Recruiter는 자신의 Posting에 참여 신청을 한 참가자들의 이력서, 정보 등을 확인한 후 자신이 원하는 사람들의 신청을 수락한다.
Failure Response	Post의 모든 모집 인원이 전부 찬 경우, 모집이 완료되었다는 팝업을 띄운다. Post의 최소 조건을 충족하지 못하는 경우 해당 조건을 충족하지 못했다는 팝업을 띄운다. (학과, 학년)

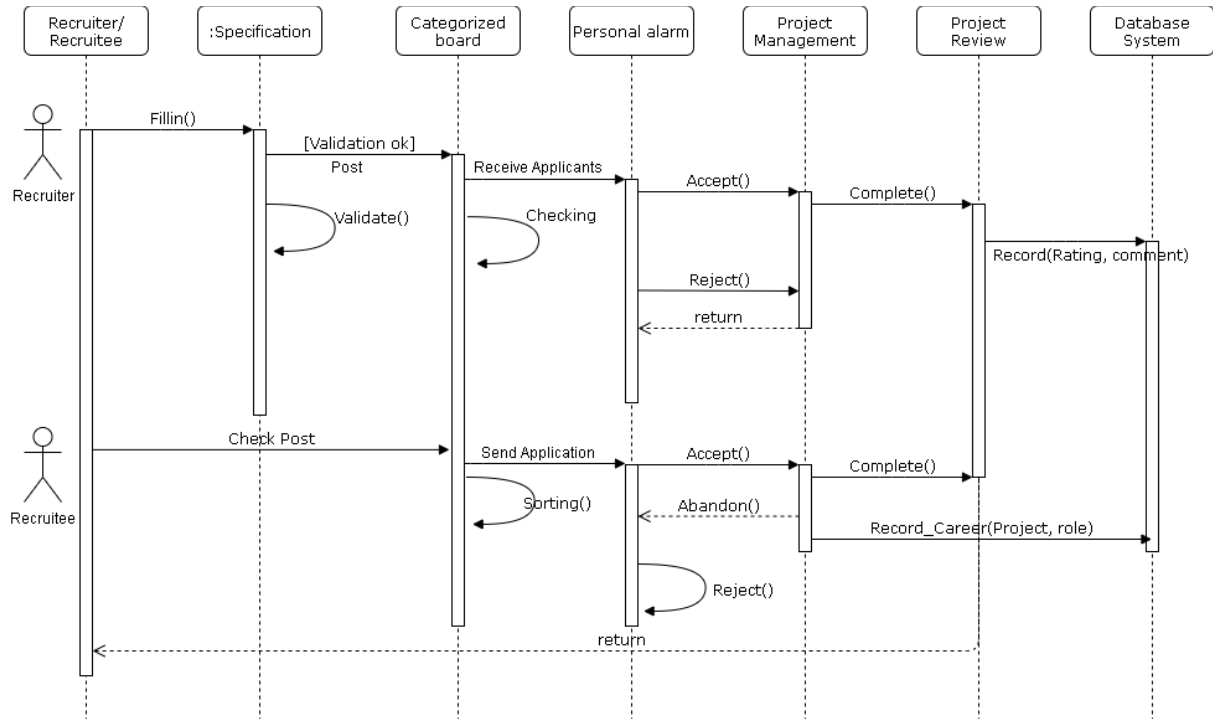
#### 6) Project Review

Use case	Project Review
Actor	Recruiters, Recruitees
Description	프로젝트에 참여한 인원들이 상호 평가를 진행하는 과정이다.
Trigger	프로젝트가 완료된 후 서로에 대한 평가를 의무적으로 진행한다.
Success Response	프로젝트가 끝난 후 5점 만점의 점수 평가와 한 줄 평을 남기도록 한다. 이때 남긴 점수는 상대방의 프로필에 뜨는 평점에 영향을 준다. 악의적인 평가를 남긴 경우, 신고할 수 있으며 그 과정이 타당하다고 여겨지는 경우 악의적 평가는 무효로 되고, 평가를 남긴 이는 블랙리스트에 등록된다.
Failure Response	평가를 진행하지 않고 다른 화면으로 넘어가려 하는 경우, 평가를 진행해야만 다른 활동을 할 수 있다는 알림을 띄운다. 사용자의 class에

남은 평가 여부의 변수를 설정해 중간에 어플이 중단되어도 다시 켜올 때 평가를 먼저 진행할 수 있도록 한다.

## 7.3 Behavioral Models

이 장에서는 본 시스템의 전반적인 흐름에 대한 Sequence Diagram을 통해 System의 Behavioral Model을 도식화했다.



## 8. System Evolution

이 챕터에서는 시스템의 배포 이후 시스템이 사용자와 주변 환경 등과 상호작용하면서 발생할 변화에 대하여 예측하고 변화에 대응하여 시스템이 발전해야 할 방향을 제시한다.

### 8.1. Job Offer System

모집자와 지원자를 매칭 시켜주는 절차는 통상적으로 모집자가 인원 모집 공고를 업로드 하고 공고를 지원자가 찾아 지원하여 모집자가 수락하면 매칭이 성사되는 방식이다. 만일 시스템의 사용자가 안정적으로 확보된다면 지원자에게서 이력을 공개한다는 동의 하에 모집자가 먼저 지원자의 포트폴리오를 보고 프로젝트를 같이 진행하자고 제안을 할 수 있는 일자리 제안 기능을 추가하는 것이 가능하다.

### 8.2. Recommendation system by machine learning

초기 시스템에서는 사용자에 대한 데이터가 존재하지 않기 때문에 사용자가 관심 분야로 설정한 카테고리나 100% 일치하는 정확도를 중점적으로 모집 공고를 추천해야 한다. 하지만 시스템을 운영하면서 시스템의 사용량이 늘어난다면 일정 카테고리를 선호한다고 지정해 놓은 사용자들이 어떤 모집 공고를 선택했는지 카테고리별 데이터를 대량으로 축적할 수 있다. 이렇게 축적한 데이터를 기반으로 알고리즘을 학습한다면 사용자가 관심이 있을 법한 내용의 공고를 추천하는 것이 가능해질 것이다.

### 8.3. Sanction against malicious evaluation

시스템에서 팀을 이루어서 활동을 종료하게 되면 같이 활동을 진행한 팀원에 대하여 평가를 하도록 하게 되어있다. 하지만 욕설, 원색적인 비난 등 문제가 될 수 있는 평가와 무차별적인 허위 신고 남발은 시스템 자체에 대한 사용자의 신뢰도에 악영향을 줄 수 있다. 그렇기 때문에 악의적인 평가와 허위 신고를 판단을 하는 것은 중요하지만 자동화로 판단하는 것은 어려운 일이다. 따라서 운영 초기 단계에 중점적으로 신경을 써야 하고 어느정도 데이터가 쌓이면 반자동화 단계로 넘어가 알고리즘이 제대로 기능을 하는지 모니터링하는 작업이 필요하다.

## 8.4. Extension to Web

시스템은 기본적으로 모바일 어플리케이션 서비스이다. 하지만 모바일 어플리케이션과 연동되는 웹 서비스도 제공하게 된다면 사용자의 접근성이 더 높아질 것이고 사용자 수의 증가와 함께 시스템에 대한 만족도도 향상되는 효과를 기대할 수 있다.

## 9. Appendices

이번 챕터에서는 앞서 본문에서 다루지 못한 시스템 개발에 관련된 세부 정보들을 기술한다. 시스템 운영 환경에 필요한 하드웨어 요구사항, 시스템에 사용되는 데이터베이스의 개략적 구조 등에 대해 기술한다.

### 9.1 Hardware Requirements

본 시스템은 Flutter를 통해 Native 앱으로 개발될 것이기 때문에 안드로이드, iOS, 웹 모두를 지원할 것이다. 안드로이드의 경우 안드로이드 5.0이상, iOS는 11.0이상, 웹브라우저의 경우 HTML5와CSS3를 지원하여야 한다. 또한 외부 데이터베이스(Firebase)에서 사용자의 정보를 저장하고, 가져오기 때문에 반드시 네트워크와 연결된 상황에서만 사용할 수 있다.

### 9.2 Database requirements

본 시스템은 시스템의 개발시간 단축 및 개발 효율 상승을 위해 BaaS로써 Firebase를 도입할 것이다. 시스템은 사용자의 정적인 정보, 평점, 프로젝트 소개 게시글 등을 위주로 취급하기 때문에 가볍고 빠른 속도를 우선시해 NoSQL기반의 Firebase DB를 선택했다.

## 10. Index

### 10.1 Tables

Table 1: Glossary.....	13
Table 2: sign up.....	24
Table 3: Sign In & Certification of Belonging.....	25
Table 4: My Page.....	26
Table 5: Recruit Enrollment.....	27
Table 6: Job Search.....	28
Table 7: Recruit Recommendation.....	29
Table 8: Workforce Matching.....	29
Table 9: Notification.....	30
Table 10: User Evaluating System.....	31
Table 11: Declaration System.....	31
Table 12: Blacklist and Suspension.....	32

### 10.2 Figures

Figure 1: Everytime Application Recruitment example.....	10
Figure 2: PhD.Kimnet example.....	11

### 10.3 Diagrams

Diagram1: System Overview Diagram.....	20
Diagram 2: Front-end Architecture.....	21

Diagram 3: Back-end Architecture.....	22
Diagram 4: Job Recommendation System Architecture.....	23
Diagram 5: Context Diagram.....	40
Diagram 6: Process Diagram.....	40
Diagram 7: Use case Diagram.....	41
Diagram 8: Behavioral Model.....	44