

1. Introduction

당신은 풀스택 개발자로서 새로운 웹 페이지 개발에 대한 의뢰를 받는다. 의뢰 받은 내용은 [코딩 프리랜서 헤드헌팅 사이트]를 개발하는 것이다. 해당 사이트는 프리랜서(코더)와 외주를 맡기고자 하는 의뢰자를 매칭하고, 매칭된 의뢰에 대하여 관리하는 사이트이다. 본 사이트에서 의뢰자는 의뢰와 함께 필요한 인력에 대한 정보를 올릴 수 있다. 프리랜서는 의뢰에 신청하여 의뢰자가 수락할 경우, 의뢰를 진행하고 그에 대한 보수를 받을 수 있다. 프리랜서 및 의뢰자로 구분되는 모든 회원들과 의뢰에 대한 모든 내용은 사이트 관리자가 모니터링하며 관리할 수 있어야 한다.

2. System Goals

당신은 [코딩 프리랜서 헤드헌팅 사이트]의 관리자, 프리랜서, 의뢰자가 간단한 UI를 통해 의뢰 관리, 의뢰 신청 및 수락, 의뢰에 대한 평가 등을 하는 시스템을 구현해야 한다.

이와 같은 목적을 이루기 위해 시스템은 구체적으로 다음과 같은 조건을 만족해야 한다.

- ✓ 시스템은 RDBMS(관계형 데이터베이스 관리 시스템)을 통해 사이트에서 필요한 데이터들을 관리하며, 네트워크를 통해 데이터베이스 서버와 통신할 수 있어야 한다.
- ✓ 시스템은 사용자 (관리자 / 프리랜서 / 의뢰자) 로그인 기능을 갖춰야 한다. 시스템 관리자 계정은 유일하며 미리 등록되어 있다. 프리랜서, 의뢰자의 경우 가입 시스템을 통해 가입을 할 수 있어야 한다.
- ✓ 시스템은 데이터를 관리하며, 사용자에게 맞는 통계 데이터를 제공한다..
- ✓ 시스템의 구체적인 기능은 '5.System Requirements Specification'에 따라 설계 되어야 한다.

3. Glossary Of Terms

이하 본문에서 사용하는 용어들은 아래의 용어 정의를 따라 사용된다.

- ✓ 관리자: System Administrator에 해당하며, 해당 시스템의 사용자들을 관리하고 현재 시스템에 등록된 의뢰를 관리한다.
- ✓ 의뢰자: 특정 소프트웨어가 필요한 사람으로, 본 시스템의 프리랜서에게 의뢰를 맡기고자 하는 사용자이다.

- ✓ 프리랜서: 특정 의뢰를 수행하는 사용자로, 본인의 개발 경력(년 수), 사용가능 언어, 나이, 포트폴리오 등 의뢰 수행 역량에 대한 정보를 본 시스템에 입력 및 관리한다
- ✓ 포트폴리오: 프리랜서가 자신의 개발 역량을 증명하기 위해 올리는 진행했던 개발 내역, 의뢰 등의 정보를 의미한다. 시스템 내적 포트폴리오, 외적 포트폴리오 두 가지로 나누어진다. 시스템 내적 포트폴리오는 해당하는 프리랜서가 본 시스템에서 진행하고 완료했던 의뢰에 대한 정보를 의미하고, 시스템 외적 포트폴리오는 본 시스템 이외에 프리랜서가 진행했던 개발내용에 대한 문서(pdf, docx 등)를 의미한다.
- ✓ 의뢰: 특정 소프트웨어 개발을 원하는 의뢰자가 본 시스템에 올리는 모집 공고와 같은 개념으로, 의뢰 문서, 의뢰 금액, 개발 시작 날짜 및 개발 종료 날짜 등이 관리된다.
- ✓ 의뢰 목록: 본 사이트에 등록된 의뢰들의 목록을 의미하며, 프리랜서는 이 목록을 확인 후, 의뢰에 신청할 수 있다.
- ✓ 의뢰 신청: 의뢰자가 올린 의뢰 목록을 보고 프리랜서가 이 의뢰를 수주하겠다고 신청하는 것을 의미한다. 의뢰에 신청한다고 바로 의뢰를 진행하는 것이 아닌, 의뢰자가 프리랜서의 정보를 확인하고 이를 수락하여야 진행할 수 있다.
- ✓ 의뢰 문서: 의뢰자가 의뢰를 올릴 때, 개발 하고자 하는 소프트웨어의 요구사항을 담고 있는 문서를 의미한다. 한 의뢰당 한 개 이상의 문서를 올릴 수도 있으며(ex. 각 모듈 별 Spec 문서), 프리랜서는 의뢰 목록에서 이 문서를 열람할 수 있어야 한다.
- ✓ 팀: 2명 이상의 프리랜서가 모여 구성한 팀으로, 프리랜서는 팀 단위로 의뢰에 지원할 수도 있다. 팀을 만든 프리랜서가 팀장이 되며, 팀원 프리랜서의 아이디를 추가하는 방식으로 팀을 구성한다.
- ✓ 의뢰 완료 요청: 프리랜서가 개발을 끝낸 후, 의뢰자에게 의뢰 완료를 요청할 수 있으며, 의뢰자는 자신이 올린 의뢰에 대하여 의뢰 완료 요청에 대해서 수락과 거부를 할 수 있다. 수락 시, 의뢰는 의뢰 완료 상태가 되며 의뢰자와 프리랜서는 서로에 대한 평점을 지정할 수 있다. 거부 시, 평점을 지정하지 않고, 프리랜서에게 거부 사유를 알리는 메시지가 전달된다. 프리랜서는 이 메시지를 받고, 다시 완료 요청을 보낼 수 있다.
- ✓ 평점: 의뢰 종료 시, 프리랜서와 의뢰자가 서로에게 의뢰에 대한 만족도를 표현하는 수치적 정도이다. 0~5점의 값을 가진다.
- ✓ 프로그래밍 언어 별 능숙도: 각 프리랜서는 자신이 사용할 줄 아는 프로그래밍 언어와 그 정도를 0~5점으로 나타낼 수 있다. (ex. Python-3, Java-4)

- ✓ 일반 사용자: 시스템 관리자가 아닌 프리랜서, 의뢰자로 가입한 사용자를 의미한다.

4. System Requirements Specification

4.1. 시스템 구조 명세

- ✓ 시스템은 상용 **RDBMS(관계형 데이터베이스 관리 시스템)**를 사용하여야 하며, 이를 회원 정보 및 제출 데이터들을 관리하며, 네트워크를 통해 데이터베이스 서버와 통신할 수 있어야 한다. 관계형 DBMS를 사용해야 한다.
- ✓ 상용 브라우저(크롬, IE, 파이어폭스 등)를 이용하여 시스템에 접근할 수 있어야 한다.
- ✓ 시스템은 HTTP Server와 Server side로 구성되어 아래의 요구사항을 만족하는 시스템을 구현하여야 한다. 이때 사용하는 HTTP server의 종류나 programming 언어는 제한하지 않으며, 인터페이스(front-side)와 관련된 요구사항은 없다.
- ✓ 구현과 관련하여 언어나 프레임워크에 대한 제한조건은 없으나 ORM(Object-Relation Mapping)과 같은 구현에 SQL을 직접적으로 사용하지 않는 툴은 제한한다.
(*ex. Python의 sqlalchemy, Rails의 Active Record 등*)

4.2. 사용 가능한 정보

시스템은 다음과 같은 정보들을 사용할 수 있으며, 필요 시 구현기능을 간소화 시키지 않는 선에서 명기되지 않은 정보를 추가적으로 사용할 수 있다. 또한 아래 정보들을 통하여 도출(derived)될 수 있는 새로운 정보들, 혹은 차별화를 위한 추가적 기능에 사용될 정보 등은 생성 가능하다. (단, 보고서에 그 이유를 반드시 서술할 것)

4.2.0. 프리랜서

- ✓ 프리랜서는 각각의 고유한 ID를 가진다.
- ✓ 프리랜서는 이름, 나이, 개발 경력(년 수), 전공 학과, 핸드폰 번호, 프로그래밍 언어 별 능숙도, 포트폴리오 정보를 가진다.
- ✓ 프로그래밍 언어 별 능숙도는 하나 이상의 언어를 입력 해야 하며, 그 범위는 0~5점이다.
- ✓ 프리랜서는 각자의 포트폴리오를 입력 및 관리 할 수 있다. 시스템 내 의뢰를 수행하는 경우 자동으로 내적 포트폴리오가 생성되며 가지고 있어야 한다. 필요에 따라 시

시스템 외적 포트폴리오를 직접 수기 입력할 수 있다.

- ✓ 프리랜서 별로 각자의 내적 포트폴리오 점수를 기반으로 한 평점이 기록된다.

4.2.1 의뢰자

- ✓ 의뢰자는 각각의 고유한 ID를 가진다.
- ✓ 의뢰자는 이름, 핸드폰 번호에 대한 정보를 가진다.
- ✓ 의뢰자 별로 요청한 의뢰들의 완료 시점 평점을 기반으로 평점이 기록된다..

4.2.2 의뢰

- ✓ 각 의뢰는 한 명의 의뢰자가 존재한다.
- ✓ 의뢰는 의뢰 금액, 개발 시작 날짜, 개발 종료 날짜를 가진다.
- ✓ 의뢰는 의뢰에 필요한 프로그래밍 언어 별 능숙도를 가질 수 있다.
- ✓ 의뢰는 프리랜서가 의뢰를 진행하기 위해 필요한 최소 경력(년 수)을 가질 수 있다.
- ✓ 의뢰는 한 개 이상의 의뢰 문서를 가진다.
- ✓ 의뢰는 의뢰 진행에 필요한 최소/ 최대 프리랜서 수를 가진다. 이 때, 최소 프리랜서 수가 2명 이상일 경우, 팀 단위 신청만 가능하다.

4.2.3 팀

- ✓ 각 팀은 고유한 이름을 가진다.
- ✓ 팀은 팀에 속한 프리랜서 리스트를 갖는다.
- ✓ 팀이 하나의 의뢰를 마치는 경우, 해당 의뢰 평점을 팀원들이 동일하게 받는다.

4.3. 기능적 요구 사항

위와 같은 정보를 활용하여, 관리자와 사용자에게 아래와 같은 주기능을 제공할 수 있어야 한다.

4.3.1. 가입, 로그인 기능

- 사용자(관리자 / 프리랜서 / 의뢰자)들이 본 사이트를 이용하기 위해서는 로그인 이 필요하다. 이를 위해 본 사이트는 사용자 인증 기능을 갖추어야 한다.
- 관리자의 계정 정보는 시스템 상에 미리 저장되어 있으나 일반 사용자(프리랜서, 의뢰자)의 경우 각각 필요한 정보를 저장하는 회원 가입 기능이 있어야 한다.

4.3.2 관리자 기능

- 관리자는 본 사이트의 모든 일반 사용자 계정 정보를 목록으로 확인할 수 있으며, key를 제외한 정보를 수정하거나, 계정 자체를 삭제 할 수 있어야 한다. 단, 모든 사용자의 비밀번호는 암호화되어 있으며, 관리자도 확인할 수 없다.
- 관리자는 본 사이트의 모든 의뢰를 목록으로 확인할 수 있으며, 각 의뢰의 정보를 수정하거나, 삭제 할 수 있어야 한다. 의뢰자가 수락하여 진행 중인 의뢰에 대해서는 수정, 삭제가 불가능하다.
- 관리자는 본 사이트의 모든 팀을 열람할 수 있으며, 현재 의뢰를 수행중인 팀을 제외한 모든 팀들의 수정,삭제가 가능하다.

4.3.3 프리랜서 기능

- 프리랜서는 자신의 아이디를 제외한 모든 정보를 수정할 수 있다..
- 프리랜서는 자신의 포트폴리오를 확인, 수정할 수 있다. 이 때 포트폴리오는 시스템 외적 포트폴리오와 내적 포트폴리오로 나뉘는데 외적 포트폴리오는 프리랜서가 직접 문서를 올릴 수 있어야 하고, 내적 포트폴리오는 본 시스템에서 진행했던 의뢰의 기본 정보와 의뢰 완료 결과물에 대한 내역을 확인 할 수 있어야 한다.
- 프리랜서는 현재 구인 중인 의뢰 목록을 확인할 수 있어야 한다. 의뢰 목록은 의뢰 시작 날짜 순, 의뢰 금액 순으로 정렬하여 볼 수 있어야 한다.
- 프리랜서는 전체 의뢰 목록을 볼 수 있다. 또한 의뢰 목록 중 자신이 최소 조건 (경력, 프로그래밍 언어 별 능숙도)을 만족하는 신청 가능 한 리스트를 따로 볼 수 있다.
- 프리랜서는 의뢰 목록 중 하나의 의뢰를 클릭함으로써 의뢰의 자세 정보(기본 정보를 포함한 필요 경력, 필요 프로그래밍 언어 별 능숙도, 의뢰 문서)를 확인할 수 있어야 한다.

- 프리랜서는 개인으로서 의뢰를 신청할 수 있다. 이 때, 의뢰의 최소 조건을 만족해야 한다.
- 프리랜서는 자신이 신청한 의뢰의 목록을 따로 확인할 수 있다. 간단한 정보를 확인할 수 있어야 한다.
- 프리랜서는 자신이 진행중인 의뢰의 목록과 간단한 정보를 확인할 수 있다.
- 프리랜서는 진행중인 의뢰에 대해 결과물 보고서를 첨부하여 의뢰 완료 요청을 보낼 수 있다.
- (의뢰 완료 요청에 대해 의뢰자가 수락할 시) 프리랜서는 해당 의뢰의 의뢰자에 대한 평점을 지정할 수 있다.
- 의뢰 완료 요청에 대해 의뢰자가 거부할 시 프리랜서는 거부 이유 메시지를 받게 된다. 거부된 의뢰여도 프리랜서는 언제든지 의뢰요청을 다시 신청할 수 있다.

4.3.4 의뢰자 기능

- 의뢰자는 의뢰를 올릴 수 있다. 의뢰에는 4.2의 정보들이 포함되어야 한다.
- 의뢰자는 자신이 올린 의뢰 목록을 확인할 수 있다.
- 의뢰자는 자신이 올린 의뢰 목록 중에서 하나를 선택해 클릭해서 해당하는 의뢰에 신청한 프리랜서(개인 혹은 팀) 목록을 볼 수 있다. 이 목록에서 프리랜서의 경력, 프로그래밍 언어 별 능숙도, 평점, 포트폴리오 등의 정보를 확인할 수 있어야 한다. 팀의 경우 팀원들의 개개인의 상세 정보를 확인할 수 있다. 이 후, 원하는 프리랜서를 선택하여 의뢰를 진행할 수 있다.
- 의뢰자는 현재 진행 중인 의뢰 목록을 확인할 수 있어야 한다. 이 목록은 의뢰 시작 시간, 의뢰 금액 순으로 정렬 가능해야 한다.
- 의뢰자는 의뢰 완료 요청을 확인 할 수 있어야 한다. 의뢰자는 의뢰 완료 요청에 대하여 수락 혹은 거부할 수 있다. 수락할 시, 프리랜서의 평점을 지정하게 되며 프리랜서에게 수락 신호를 보낸다. 거부 할 시 거부 사유를 반드시 작성하여야 한다.

4.3.5 의뢰 기능

- 의뢰에서 최소 프리랜서 수가 2명 이상일 경우, 팀 단위의 신청만 받을 수 있으

며 팀원의 수와 필요 개발자 수가 동일해야 한다.

- 의뢰에 대해서 팀 단위 신청의 경우, 의뢰자가 의뢰에 필요한 프로그래밍 언어별 능숙도를 설정한 값은 팀원 중 한 명이 만족해도 충족한 것으로 인정한다. 예를 들어, A 의뢰에서 Python 3, Java 4 로 설정을 했다면, 팀원 중 한 명이 Python 3, 다른 한 명이 Java 4 라면 만족한 것으로 인정한다.
- 의뢰에 대해서 팀 단위 신청의 경우, 의뢰자가 의뢰에 필요한 최소 경력 값은 팀원 모두가 만족하는지를 확인한다. 예를 들어, A 의뢰에서 최소 경력을 10년으로 설정 했다면, 모든 팀원의 경력이 10년 이상이어야 신청할 수 있다.

4.3.6 팀 기능

- 프리랜서는 팀을 만들 수 있다. 이 경우, 만든 프리랜서가 팀장이 된다.
- 팀장은 팀원의 ID를 입력하여 팀원을 통해 추가할 수 있다.
- 팀장은 팀원을 추방 시킬 수 있다.
- 팀장은 팀을 삭제할 수 있다.
- 팀장은 팀으로서 의뢰에 신청할 수 있다. 이 경우, 의뢰 완료 요청, 의뢰 신청 프리랜서 리스트 등 모든 정보는 모두 팀 단위로 따로 관리할 수 있어야 한다.
- 한 명의 프리랜서는 여러 개의 팀에 속할 수 있으며, 동시에 여러 팀의 팀장이 될 수도 있다.

4.3.2. 필수 기능 리스트

System Requirement Specification 바탕으로 아래의 필수 기능들을 구현하여야 하며 **최종 시연 평가는 아래의 필수 기능 리스트를 기준으로 평가한다.** (검정-1, 빨강-2)

- | | |
|---|--|
| 1 | 관리자는 미리 정의된 아이디와 패스워드를 통해 로그인할 수 있어야 한다. |
| 2 | 일반 사용자(프리랜서, 의뢰자)는 본 시스템에 가입을 할 수 있어야 한다. 가입 시, 각각에 알맞은 정보를 입력해야 한다. |
| 3 | 관리자는 본 시스템의 일반 사용자 계정 정보, 의뢰 목록, 팀 목록, 거부된 의뢰 |

목록을 확인할 수 있으며 이에 대한 수정이 가능하다.

- 4 프리랜서는 자신의 정보를 수정할 수 있어야 한다. 이 때, 자신의 시스템 내적 포트폴리오를 확인하고, 외적 포트폴리오를 업로드 할 수 있어야 한다.
- 5 프리랜서는 현재 구인 중인 의뢰 목록을 확인할 수 있어야 한다. 의뢰 목록은 의뢰 시작 날짜 순, 의뢰 금액 순으로 정렬하여 볼 수 있어야 한다.
- 6 프리랜서는 개인으로서 의뢰를 신청할 수 있어야 한다. 이 때, 의뢰의 최소 조건(경력, 프로그래밍 언어 별 능숙도)을 만족해야 한다.
- 7 프리랜서는 자신이 진행중인 의뢰의 목록과 간단한 정보를 확인할 수 있어야 한다.
- 8 프리랜서는 진행중인 의뢰에 대해 의뢰 완료 요청을 보낼 수 있다.
- 9 프리랜서는 의뢰 완료 시(의뢰 완료 요청에 대해 의뢰자가 수락할 시) 의뢰자의 평점을 지정할 수 있다.
- 10 의뢰자는 의뢰를 올릴 수 있다. 의뢰에는 4.2의 정보들이 포함되어야 한다.
- 11 의뢰자는 현재 진행 중인 의뢰 목록을 확인할 수 있어야 한다. 이 목록은 의뢰 시작 시간, 의뢰 금액 순으로 정렬 가능해야 한다.
- 12 의뢰자는 의뢰 목록을 클릭해서 해당하는 의뢰에 신청한 프리랜서(개인 혹은 팀) 목록을 볼 수 있다. 이 목록에서 프리랜서의 경력, 프로그래밍 언어 별 능숙도, 평점, 포트폴리오 등의 정보를 확인할 수 있어야 한다. 이 후, 원하는 프리랜서를 선택하여 의뢰를 진행할 수 있다.
- 13 의뢰자는 의뢰 완료 요청을 확인 할 수 있어야 한다. 의뢰자는 의뢰 완료 요청에 대하여 수락 혹은 거부 할 수 있다. 수락할 시, 프리랜서의 평점을 지정하게 되며 프리랜서에게 수락 신호를 보낸다.
- 14 의뢰자는 의뢰를 거부할 수 있으며, 거부 시 메시지가 의뢰인에게 전달된다.
- 15 프리랜서는 의뢰 완료가 거부 당할 시 거부 사유 메시지를 확인할 수 있으며, 다시 한번 의뢰 완료 요청이 가능하다.

4.4. Extra Credit

- 추가 점수는 위 기능 설계의 팀 단위 Process 를 구현할 경우 주어지며, 최대 점수 이상으로는 부여되지 않는다.
- 그 외 추가 기능

위 사항 외에도 필수 기능 리스트 외의 기능을 추가로 구현하거나 시스템의 약점을 보완한다면 가산점이 부여될 수 있다. 추가 기능에 대한 구상은 자유롭게 가능하다. 단, 본 명세서에 제시된 내용을 활용되어야 한다. 즉, 전혀 상관없는 내용을 추가로 구현한 것은 해당되지 않는다.

5. 프로젝트 일정

(* 각 보고서 별 필수 항목들 파란색으로 표기)

	Due Date	Description
1차 보고서	11월 14일 (~ AM 10:00)	ER-Diagram & Mapping ERD to Relational Schema <ul style="list-style-type: none"> ✓ 시스템의 기초가 되는 1)ER-Diagram을 작성하고, 이를 2)relational schema로 매핑한다. ✓ 이 시점에서 정한 내용들은 추후 변경 가능하지만, 전체적인 일정을 고려하여 가능한 최적화된 분석을 하여 제출할 수 있도록 한다. ✓ 3)ERD 작성과 관련하여 가정한 내용, 4)팀원의 역할 분담 내용, 5)일정을 기록한 보고서를 제출한다.
2차 보고서	11월 21일 (~ AM 10:00)	시스템 설계 문서 및 중간 구현내용 <ul style="list-style-type: none"> ✓ 1)시스템 설계 문서는 State Diagram, Sequence Diagram, Application Flowchart 등 다양한 표현 방식을 자유롭게 포함할 수 있으며, 각 팀의 아이디어를 최대한 나타낼 수 있는 것으로 한다. Database table 생성 script 명기 <ul style="list-style-type: none"> ✓ 또한 2차 보고서에는 시스템이 사용하는 2)Database table 들의 생성 script를 첨부한다. ✓ 1차 보고서에서 3)변경/추가된 내용을 반드시 서술한다.

최종보고 서	12월 12일 (~PM 11:59)	최종 결과 보고서 및 구현 내용 <ul style="list-style-type: none"> ✓ 완성된 시스템에 대한 최종적인 보고서는 1)결과물과 더불어 단계별 프로젝트 2)진행 과정을 포함한다. (사용된 쿼리 포함) ✓ 완성도 평가를 위하여 3)기능별 구체적 구현 내용을 서술한다. ✓ 시스템에 사용된 4)DB 전체를 dump하여 제출한다. ✓ 보고서의 가독성 또한 평가의 요소가 된다. ✓ 1,2차 보고서에서 5)변경/추가된 내용을 반드시 서술한다.
최종시연	12월 12일 (예정)	모든 조를 대상으로 시연 평가 <ul style="list-style-type: none"> ✓ 추후 공지되는 장소에서 팀별 시연 평가 예정.

6. 주의 사항

- ✓ 팀 구성은 자유롭게 최대 3인 1조를 원칙으로 합니다.
- ✓ 각 단계별 제출기한을 엄수, 제출기한 이후의 결과물은 인정하지 않습니다.
- ✓ 보고서와 프로그램을 [팀명]_project1.zip / [팀명]_project2.zip / [팀명]_project3.zip 으로 **압축하여 YSCEC 보고서 게시판에 업로드 후 보고서를 하드카피로 공지된 장소에 제출**합니다. (ex. 언더우드팀_project1.zip , ...)
- ✓ 각 단계별 수정사항이나 차별화를 위한 추가 기능 및 부각하고자 하는 장점이 있다면 보고서에 명시적으로 작성해야 합니다.
- ✓ 과제의 목표는 일반적인 **관계형 데이터베이스**를 이해하고 **SQL**을 사용하는 것입니다. 따라서 본 목표에 부합하지 않는 구현물에 대해서는 감점이 있을 수 있으니 유의하세요. (관계형 데이터베이스, SQL 사용하지 않을 시 감점, SQLite등 사용 금지)