

2 차보고서

Team Project



수업명 데이터베이스

교수님 강동현

팀 명 UICIAN

제출일 23/11/18

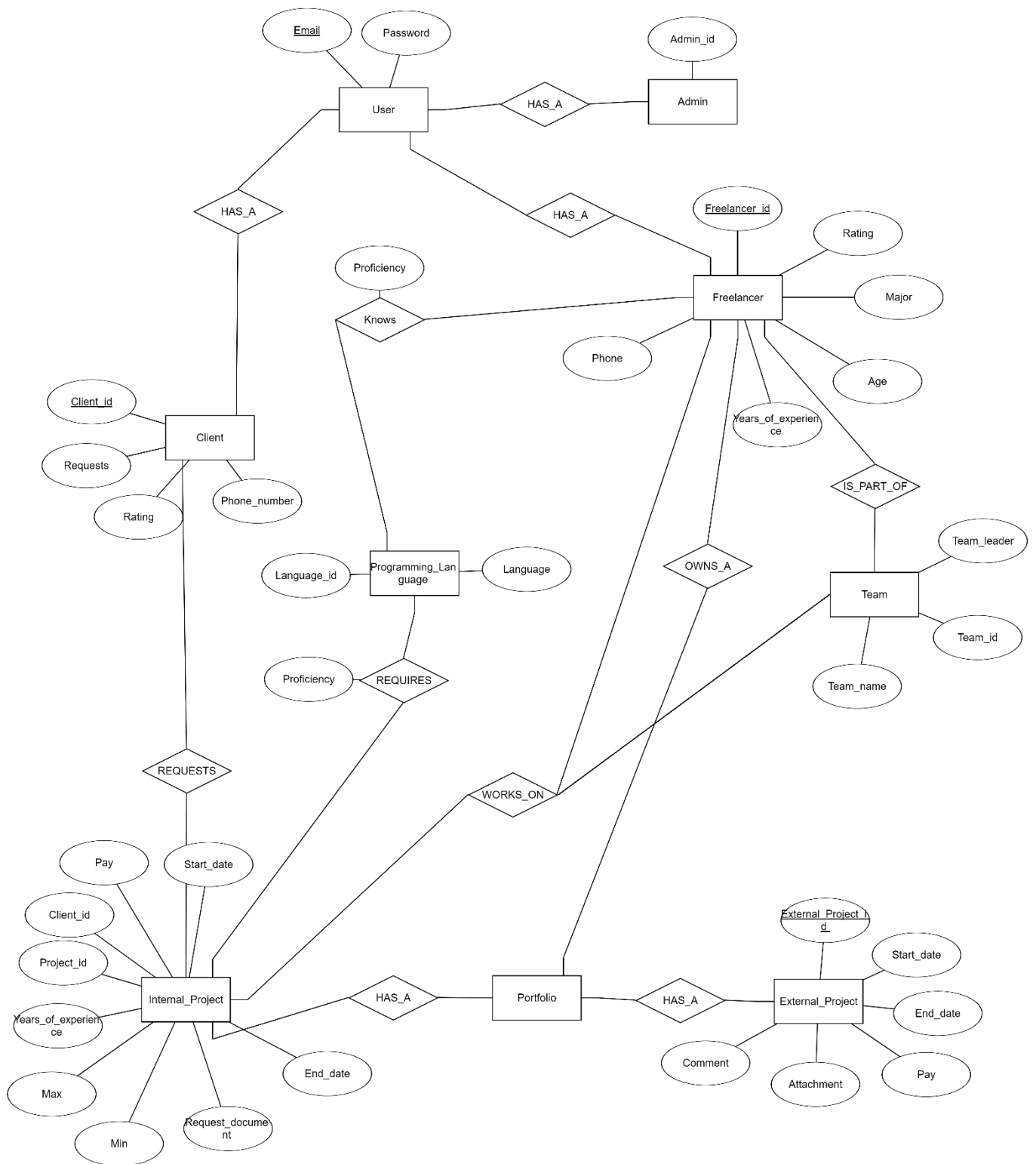
김민석 2011190094, 윤호재 2014190080, 이진우 2014190008

목차

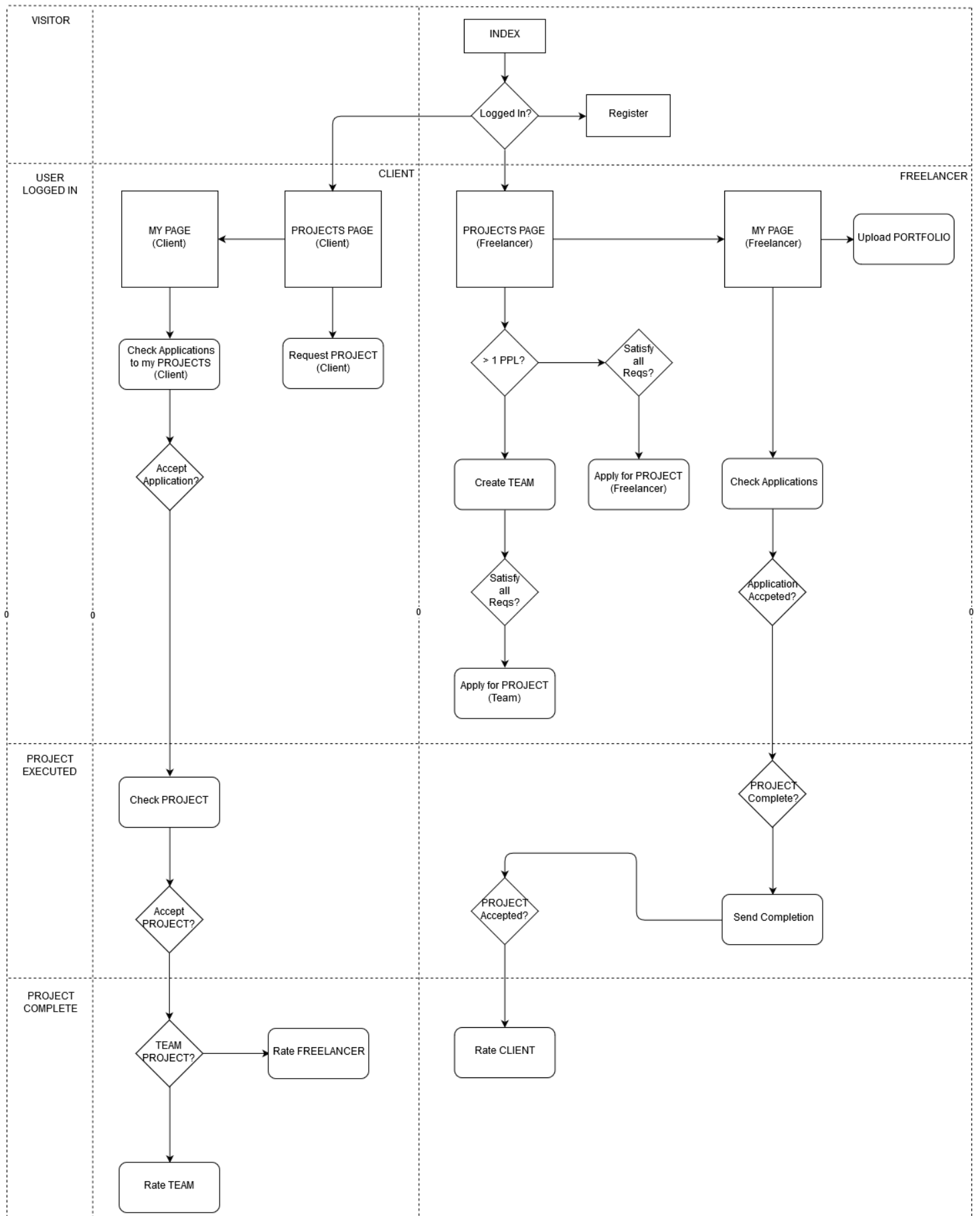
1. ER-Diagram
2. Application Flow Chart
3. Database Table 생성 Scripts
4. 웹사이트 UI 스크린샷
5. 일정 및 팀원 별 수행 역할

부록. ERD 작성을 위한 모든 Requirements 정리

1. ER-Diagram (변동사항 無)



2. 시스템 설계 (Application Flow Chart)



3. Database Table 생성 Scripts

```
01. -- Freelancer Table Create SQL
02. CREATE TABLE Freelancer
03. (
04.     `idx`          INT          NOT NULL    AUTO_INCREMENT,
05.     `email`        VARCHAR(45)  NOT NULL    COMMENT 'Email address',
06.     `password`     VARCHAR(255) NOT NULL    COMMENT 'Password',
07.     `name`         VARCHAR(45)  NOT NULL    COMMENT 'Name',
08.     `age`          INT          NOT NULL    COMMENT 'Age',
09.     `major`        VARCHAR(255) NOT NULL    COMMENT 'Major',
10.     `phone`        VARCHAR(13)  NOT NULL    COMMENT 'Phone',
11.     `experience`   INT          NOT NULL    COMMENT 'Years of experience',
12.     `rating`       DOUBLE       NULL       COMMENT 'Rating',
13.     PRIMARY KEY (idx)
14. );
15.
16.
17. -- Client Table Create SQL
18. CREATE TABLE Client
19. (
20.     `idx`          INT          NOT NULL    AUTO_INCREMENT,
21.     `email`        VARCHAR(45)  NOT NULL    COMMENT 'Email address',
22.     `password`     VARCHAR(255) NOT NULL    COMMENT 'Password',
23.     `phone`        VARCHAR(13)  NOT NULL    COMMENT 'Phone',
24.     `rating`       DOUBLE       NULL       COMMENT 'Rating',
25.     PRIMARY KEY (idx)
26. );
27.
28.
29. -- Internal_project Table Create SQL
30. CREATE TABLE Internal_project
31. (
32.     `idx`          INT          NOT NULL    AUTO_INCREMENT,
33.     `client_idx`   INT          NOT NULL    COMMENT 'Client idx',
34.     `start_date`   DATE         NOT NULL    COMMENT 'Start date',
35.     `end_date`     DATE         NOT NULL    COMMENT 'End date',
36.     `min_part`     INT          NOT NULL    COMMENT 'Minimum participants',
37.     `max_part`     INT          NOT NULL    COMMENT 'Maximum participant',
38.     `experience`   INT          NOT NULL    COMMENT 'Years of experience',
39.     `pay`          DOUBLE       NOT NULL    COMMENT 'Pay',
40.     `req_doc`      BLOB         NOT NULL    COMMENT 'Request document',
41.     PRIMARY KEY (idx),
42.     FOREIGN KEY (client_idx) REFERENCES Client (idx)
43.     ON DELETE CASCADE ON UPDATE CASCADE
44. );
45.
46. -- Programming_language Table Create SQL
47. CREATE TABLE Programming_language
48. (
49.     `idx`          INT          NOT NULL    AUTO_INCREMENT,
50.     `name`         VARCHAR(100) NOT NULL    COMMENT 'Name',
51.     PRIMARY KEY (idx)
52. );
```

```

54.
55. -- Team Table Create SQL
56. CREATE TABLE Team
57. (
58.     `idx`          INT          NOT NULL    AUTO_INCREMENT,
59.     `name`         VARCHAR(255) NOT NULL    COMMENT 'Name',
60.     `comment`      VARCHAR(255) NULL       COMMENT 'Comment',
61.     `leader_idx`   INT          NOT NULL    COMMENT 'Leader',
62.     PRIMARY KEY (idx),
63.     FOREIGN KEY (leader_idx) REFERENCES Freelancer (idx)
64.     ON DELETE CASCADE ON UPDATE CASCADE
65. );
66.
67. -- External_project Table Create SQL
68. CREATE TABLE External_project
69. (
70.     `idx`          INT          NOT NULL    AUTO_INCREMENT,
71.     `start_date`   DATE         NOT NULL    COMMENT 'Start date',
72.     `end_date`     DATE         NOT NULL    COMMENT 'End date',
73.     `pay`          DOUBLE       NOT NULL    COMMENT 'Pay',
74.     `attachment`   BLOB         NULL       COMMENT 'Attachment',
75.     `comment`      VARCHAR(255) NULL       COMMENT 'Comment',
76.     PRIMARY KEY (idx)
77. );
78.
79.
80. -- Admin Table Create SQL
81. CREATE TABLE Admin
82. (
83.     `idx`          INT          NOT NULL    AUTO_INCREMENT,
84.     `email`        VARCHAR(45)  NOT NULL    COMMENT 'Email address',
85.     `password`     VARCHAR(255) NOT NULL    COMMENT 'Password',
86.     PRIMARY KEY (idx)
87. );
88.
89.
90.
91. -- Portfolio Table Create SQL
92. CREATE TABLE Portfolio
93. (
94.     `freelancer_idx` INT NOT NULL,
95.     `int_or_ext`     BIT NOT NULL,
96.     `int_project_idx` INT NULL,
97.     `ext_project_idx` INT NULL,
98.     FOREIGN KEY (int_project_idx)
99.     REFERENCES Internal_project (idx) ON DELETE CASCADE ON UPDATE CASCADE,
100.    FOREIGN KEY (ext_project_idx)
101.    REFERENCES External_project (idx) ON DELETE CASCADE ON UPDATE CASCADE,
102.    FOREIGN KEY (freelancer_idx)
103.    REFERENCES Freelancer (idx) ON DELETE CASCADE ON UPDATE CASCADE,
104.    UNIQUE(freelancer_idx, int_project_idx, ext_project_idx)
105. );
106.

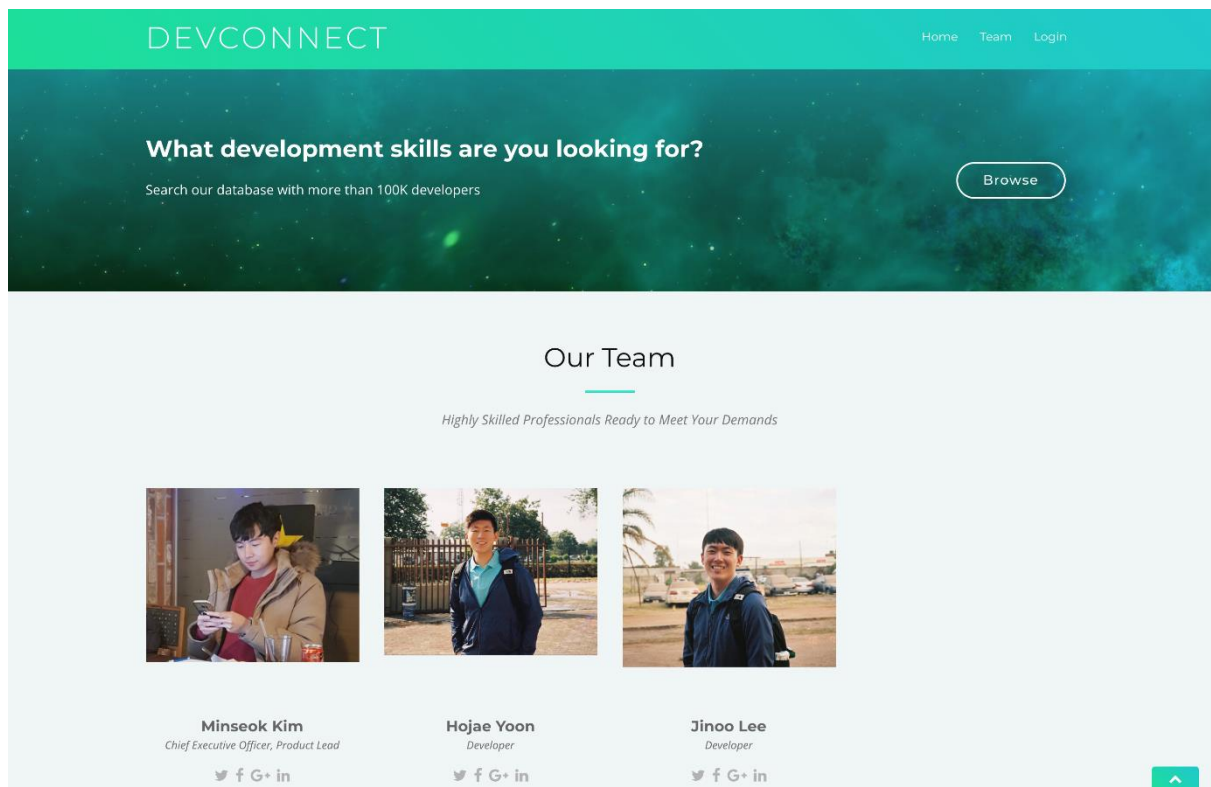
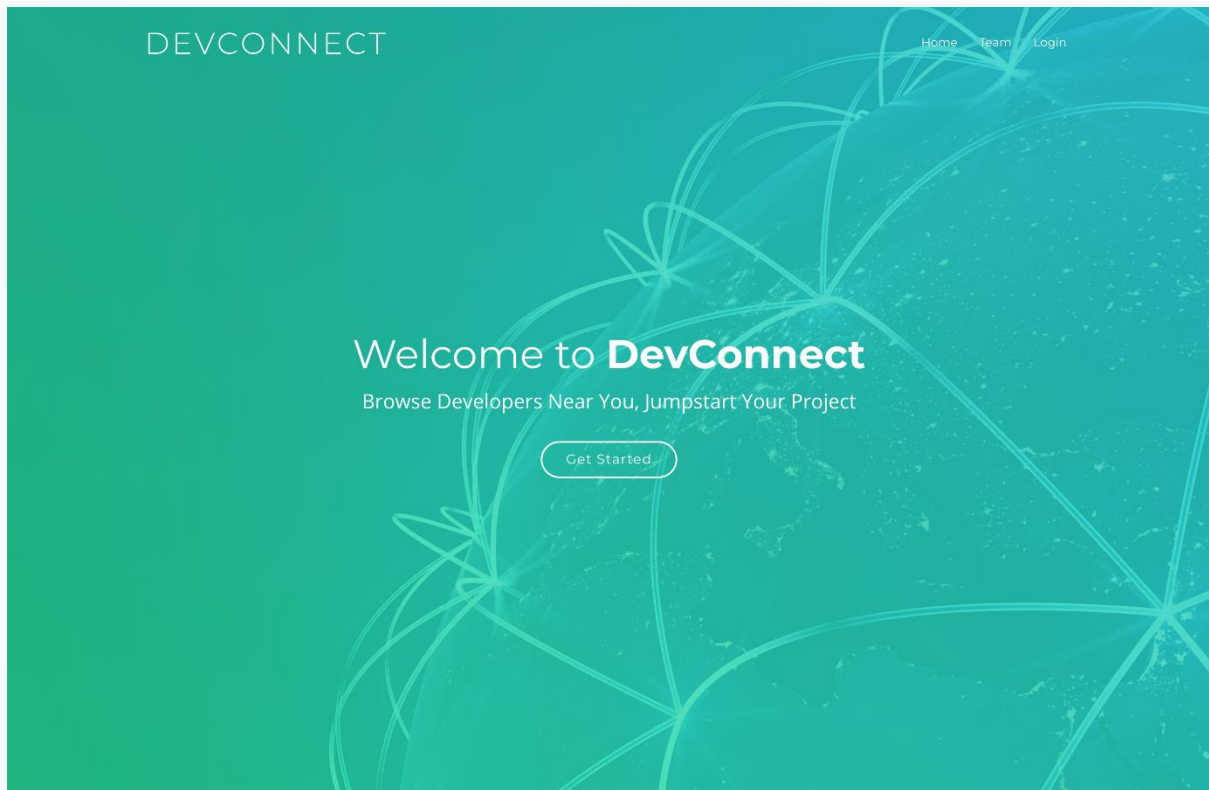
```

```

109. -- Team_member Table Create SQL
110. CREATE TABLE Team_member
111. (
112.     `team_idx`      INT      NOT NULL,
113.     `freelancer_idx` INT      NOT NULL,
114.     FOREIGN KEY (team_idx)
115.     REFERENCES Team (idx) ON DELETE CASCADE ON UPDATE CASCADE,
116.     FOREIGN KEY (freelancer_idx)
117.     REFERENCES Freelancer (idx) ON DELETE CASCADE ON UPDATE CASCADE,
118.     UNIQUE(team_idx, freelancer_idx)
119. );
120.
121.
122.
123. -- Programming_language_knowledge Table Create SQL
124. CREATE TABLE Programming_language_knowledge
125. (
126.     `language_idx`   INT      NOT NULL,
127.     `freelancer_idx` INT      NOT NULL,
128.     `proficiency`    INT      NOT NULL,
129.     FOREIGN KEY (language_idx)
130.     REFERENCES Programming_language (idx) ON DELETE CASCADE ON UPDATE CASCADE,
131.     FOREIGN KEY (freelancer_idx)
132.     REFERENCES Freelancer (idx) ON DELETE CASCADE ON UPDATE CASCADE
133. );
134.
135.
136. -- Internal_project_language_requirement Table Create SQL
137. CREATE TABLE Internal_project_language_requirement
138. (
139.     `project_idx`    INT      NOT NULL,
140.     `language_idx`   INT      NOT NULL,
141.     `proficiency`    INT      NOT NULL,
142.     FOREIGN KEY (project_idx)
143.     REFERENCES Internal_project (idx) ON DELETE CASCADE ON UPDATE CASCADE,
144.     FOREIGN KEY (language_idx)
145.     REFERENCES Programming_language (idx) ON DELETE CASCADE ON UPDATE CASCADE,
146.     UNIQUE(project_idx, language_idx)
147. );
148.
149.
150.
151. -- Current_project Table Create SQL
152. CREATE TABLE Current_project
153. (
154.     `project_idx`    INT      NOT NULL,
155.     `single_only`    BIT      NOT NULL,
156.     `team_only`      BIT      NOT NULL,
157.     `freelancer_idx` INT      NULL,
158.     `team_idx`       INT      NULL,
159.     FOREIGN KEY (freelancer_idx)
160.     REFERENCES Freelancer (idx) ON DELETE RESTRICT ON UPDATE RESTRICT,
161.     FOREIGN KEY (project_idx)
162.     REFERENCES Internal_project (idx) ON DELETE RESTRICT ON UPDATE RESTRICT,
163.     FOREIGN KEY (team_idx)
164.     REFERENCES Team (idx) ON DELETE RESTRICT ON UPDATE RESTRICT,
165.     UNIQUE (project_idx, freelancer_idx, team_idx)
166. );

```

4. 웹사이트 UI 스크린샷



5. 일정 및 팀원 별 수행 역할

	모임 일정	김민석	윤호재	이진우
1	11월 14일 (수)	1. ERD 및 Relational Schema 작성을 위한 모든 Requirements 이해 및 정리 2. ER-Diagram 작성 3. ERD에 해당하는 Relational Schema 매핑 4. 1차 보고서 작성 완료		
2	11월 16일 (금)	1. 웹사이트의 기능을 기반으로 <u>Application Flowchart</u> 를 작성하여 Front-end, Back-end, Database 측면에서 필요한 모든 개발 사항 검토 및 분배 2. 시스템 설계 문서 작성		
3	11월 21일 (수)	<u>Front-end 초기 구현**</u> 1. HTML/CSS/JS를 사용하여 사용자 인터페이스 구현 2. 로그인/회원 가입 및 유효성 검사 기능 구현	<u>Back-end 초기 구현**</u> 1. 공동 작업이 가능한 Heroku 서버 개설 2. Node.js 활용한 REST API 구축	<u>Database 초기 구현**</u> 1. 개설된 Heroku 서버에 MySQL 모듈 연동 2. ERD에 맞춰 TABLE 생성 시작
4	11월 22일 (목)	1. 각 팀원 별 구현 파트 간단 시연 후 필요 시 수정 작업 2. 작성 코드 통합 및 2차 보고서 작성		
5*	12월 4일 (화)	<u>Front-end 구현 및 디버깅</u>	<u>Back-end 구현 및 디버깅</u>	<u>Database 구현 및 디버깅</u>
6	12월 5일 (수)	1. 내부적 최종 마감일 2. 팀원 별 진행 과정 및 최종 결과 공유 및 수정 사항 검토 3. 최종 결과 보고서 작성 (변경 및 추가 내용 작성 예정)		
7	12월 7일 (금)	1. 모든 수정 사항 완료 및 발표용 PPT 작성 2. 2~3회의 시연 연습 진행		

*3차 모임 이후 일정은 회의 후 결정된 계획이며 추후 변경 가능

** 각 팀원은 단계별 진행 과정을 보여주기 위한 스크린샷 및 코드를 날짜별로 저장

부록. ERD 작성을 위한 모든 Requirements 정리

Entity

1. 관리자

2. 의뢰자

Attributes: Client ID, Name, Phone Number, Requests, Rating

- 의뢰 완료 후 평점 기록 (프리랜서가 부여)
- 평점이 부여되는 시점은 프리랜서가 제출한 결과물을 의뢰자가 수락했을 때

3. 프리랜서

Attributes: Freelancer ID, Name, Age, Years of Experience, Major, Phone Number, Programming Languages (1 or more), Language Proficiency, Portfolio, Rating (Ex. 포트폴리오 점수의 평균)

- Programming Languages: Ex. Python – 0~5
- Portfolios: Internal or External

4. 포트폴리오

Attributes: Score (Internal Only), Internal, External

- Internal: 우리 웹사이트에서 맡은 의뢰들
- External: 수기로 입력

5 의뢰

Attributes: Client ID, Pay, Project Starting Date, Project End Date, Request Document (1 or more), (min, max) # of Freelancers, Years of Experiences, [Programming Languages]¹

- 만약 최소 요구되는 프리랜서 수가 두 명 이상이면 팀 단위 신청만 가능하다

6 팀

Attributes: Team Name, List of Freelancer ID

¹ [Bracket] 안에 있는 attribute은 optional

Functional Requirements

General

로그인 기능 (관리자, 프리랜서, 의뢰자 인증)

회원 가입 기능 (프리랜서, 의뢰자)

비밀번호 암호화

Admin

계정 삭제 가능

Key 값 제외 정보 수정 [비밀번호 변경 불가능]

사이트 모든 의뢰 목록 확인 가능

각 의뢰 정보 수정/삭제 가능, **but** 진행중인 의뢰는 수정/삭제 불가능

모든 팀 열람 가능, 현재 프로젝트 진행중이지 않은 팀 수정/삭제 가능

Freelancer

모든 개인 정보 수정 가능 (아이디 제외)

포트폴리오 확인 또는 수정

- 외부: 자율적으로 문서 등록
- 내부: 시스템 내부적으로 수행했던 프로젝트 확인

의뢰 목록은 의뢰 시작날짜 순, 의뢰 금액 순 **정렬**

전체 의뢰 목록 열람 가능

- 구인 중인 항목 추출
- 자신의 경력, 프로그래밍 언어별 능숙도에 만족하는 리스트 추출

의뢰 열람 시

- 자세한 정보 확인 가능 (기본 정보, 경력, 필요 프로그래밍 언어별 능숙도, 의뢰 문서)

개인 프리랜서로 신청 가능 (의뢰 최소 조건 만족)

자신이 신청한 의뢰 목록 확인 가능

자신이 진행중인 의뢰 목록 확인 가능

의뢰 완료 요청 (언제든 다시 보낼 수 있다)

- 결과물 보고서 첨부

(의뢰자 수락 시) 의뢰자에 대한 평점 지정 가능

(의뢰자 거부 시) 거부 이유 메시지 확인 가능

Client

의뢰 등록

의뢰인이 등록한 목록 확인 가능

- 진행중인 목록 (시작 시간, 의뢰 금액 순으로 정렬 가능)
- 전체 목록

신청한 개인 또는 팀 프로필 확인 가능 -> 원하는 프리랜서 선택 가능

의뢰 완료 요청 확인 가능

- 수락: 프리랜서에게 수락 신호 전송, 프리랜서 평점 부여
- 거부: 사유 작성 필요

Min Max 프리랜서 지정 가능

- 2명 이상이면 팀으로 신청 가능
- 팀으로 지원할 시 한 명이 requirement 만족해도 충분하다
- 최소 경력 값은 모두 만족해야 함