**2021년 04월 24일**

|  |  |
| --- | --- |
| 과목 | 소프트웨어 공학 |
| 담당교수 | 김명호 |
| 학번 이름 | 20143537 오범준  20190402 이재민  20176755 김의찬  20182417 민지환  20192455 이주영  20186593 신예송 |

**< 소프트웨어 공학 13조: Requirement Analysis Document >**

**Selected Project**

* Industrial Visit Planning & Booking System

**Project Summary**

* 저희 팀이 맡은 프로젝트는, 산업체 견학 패키지 예약 및 커뮤니티 웹 어플리케이션 프로젝트입니다
* 해당 웹 어플리케이션을 통해, 유저들은, 관심있는 분야 및 직종의 다양한 산업체 견학 패키지에 대한 정보를 얻을 수 있을 뿐만 아니라, 해당 패키지를 예약할 수도 있습니다
* 뿐만 아니라, 커뮤니티의 장으로서도 작용하여, 원하는 정보를 서로 적극적으로 공유하는 등, 큰 도움을 얻을 수 있는 플랫폼 입니다

**3 Sub Systems**

1. “User Account Management & Web Main Site Interaction”
2. “Web Community site & User Comment Management”
3. “Industrial Booking System “

**Problem Statement**

현재, 기업 구직과 관련된

정보 소셜 미디어장 , 그리고

기업 구직 사이트는 산발적으로 복잡하게 나뉘어져 있는 상황입니다.

기업에 관련된 정보를 얻기 위해 다수의 사이트에 등록해야 하며,

일관되지 않은, 다양한 정보 사이에서, 적절한 정보를 선별하는 과정이

사용자에게는 큰 부담입니다

정보 소셜 미디어 시장 및 커뮤니티로는 잡 플래닛, glassdoor ,

기업 구직 관련 사이트로는 잡코리아. 등이 성공적인 사례입니다

저희 project는 2종류 사이트의 장점을 모두 가지고 있습니다

유저들간의 활발한 소통을 가능하게 하는 커뮤니티 플랫폼 기능을 할 뿐만 아니라,

구직과 관련된 기업 정보 제공, 기업 추천 기능 등 기업 구직과 관련된 다양한 도움을 줍니다

구직과 관련된 개인정보를 등록함으로써,

짧은 시간 내에 효율적으로 기업과 관련된 정보를 찾을 수 있을 뿐만 아니라

커뮤니티를 통해 보다 생생하고 최신의 기업 및 구직 정보를 얻을 수 있습니다

**Scenario**

처음, 유저는 사이트에 대한 회원가입 , 로그인 절차를 거칩니다

이후 메인페이지에서 자신에게 추천되는 기업 정보, 최근 인기 기업 정보 등에 대한

정보를 손쉽게 볼 수 있습니다.

뿐만 아니라, 자신이 원하는 조건에 맞는 기업 및 산업체를 검색하여

맞춤형 정보를 제공받을 수 있습니다.

특정 산업체 및 산업체 패키지 사항을 클릭함으로써

보다 구체적인 정보를 볼 수 있을 뿐만 아니라

패키지 예약, 후기 작성 등의 추가적인 작업을 할 수 있습니다.

한편, 메인페이지에서 커뮤니티 페이지로 가는 링크를 통해

커뮤니티 페이지로 이동할 수 있고

준비, 후기, 자유 게시판으로 나눠져있는 커뮤니티 공간 속에서

보다 적극적으로 정보를 얻을 수 있습니다.

게시글 및 댓글의 경우, 작성자의 학교, 학과가 보여지기 때문에

신뢰성 있는 정보를 접할 수 있습니다.

자신이 예약 및 찜한 패키지의 목록

혹은 작성한 게시글 및 댓글의 목록은

마이페이지 페이지에서 관리함으로써 보다 편리하게

자신의 활동 내역을 확인해 볼 수 있습니다

**1st. Sub System : User Account Management & Web Main Site Interaction**

* **Requirements**

FR1 : 방문자는 자신이 다니는 학교, 학년등의 개인정보를 입력하고 회원가입을 할 수 있다.

FR2 : 방문자는 ID와 PW을 통해 서로 구분되며 로그인을 통해 시스템에 접속한다.

FR3 : 방문자는 홈페이지 메인 화면에서 추천기업, 최근 인기있는 게시물들을 볼 수 있다.

FR4 : 방문자는 각 기업들의 견학에 대한 후기별점을 볼 수 있다.

* **Use Case Model**

**Traceability Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| **REQ#** | **PW** | UC1 | UC2 |
| **FR1** |  | **O** |  |
| **FR2** |  | **O** |  |
| **FR3** |  | **O** | **O** |
| **FR4** |  | **O** |  |

**Use Case 1 : 회원가입 및 로그인**

|  |  |
| --- | --- |
| Use Case UC-1 | 회원가입 및 로그인 |
| Related Requirement: | FR1, FR2 |
| Initiating Actor: | 방문자 |
| Actor`s Goal: | * 방문자는 자신의 개인정보를 입력하고 홈페이지 회원가입을 할 수 있다.. * 방문자는 ID와 PW을 이용하여 홈페이지에 로그인할 수 있다. |
| Participating Actors: | 산업체 db |
| Preconditions: | * 방문자는 ID와 PW를 통해 서로 구분되며 로그인을 통해 시스템에 접속한다. (UC-1) * 방문자는 개인정보를 입력하여 회원가입을 한다. |
| Postconditions: | * 로그인을 하면 이용자 db에서 ID와 PW을 확인한 후 맞으면 로그인을 시킨다. * 방문자가 회원가입을 하면 그 정보는 이용자 db에 저장된다. |
| Flow of Events for Main Success Scenario: |  |
| 1. 방문자는 자신이 원하는 서비스를 이용하기 위해 회원가입을 한다. 2. 방문자는 ID와 PW을 치고 홈페이지에 로그인한다. | | |

**Use Case 2 : 메인 홈페이지**

|  |  |
| --- | --- |
| Use Case UC-2: | 메인 홈페이지 |
| Related Requirement: | FR3 |
| Initiating Actor: | 방문자 |
| Actor`s Goal: | * 방문자는 홈페이지에서 인기 산업체, 인기 글 ,산업체 홍보와 같은 정보를 볼 수 있다. * 방문자는 다양한 카테고리를 홈페이지에서 보고 선택할 수 있다. |
| Participating Actors: | 이용자 DB, 산업체 DB, 커뮤니티 DB |
| Preconditions: | * 방문자가 로그인을 하고 홈페이지에 들어오면 통계자료를 바탕으로 한 인기있는 산업체, 인기 글들을 볼 수 있다. * 방문자는 다양한 분야의 산업체를 볼 수 있다. |
| Postconditions: | * 방문자의 나이대, 관심분야에 따라 산업체 DB에서 통계를 내서 홈페이지에 띄워준다. * 산업체 DB를 카테고리화 해 이용자가 편하게 고를 수 있게 해준다. |
| Flow of Events for Main Success Scenario: |  |
| 1. 방문자가 로그인을 하고 홈페이지에 들어온다 2. 방문자는 홈페이지에서 자신이 관심있어할만한 정보를 볼 수 있다. 3. 방문자는 다양한 카테고리를 보고 흥미있는 분야를 선택해 들어갈 수 있다. | | |

* **Domain Model**

**Extracting the responsibility**

|  |  |  |
| --- | --- | --- |
| Responsibility Description | Type  (D/k,E/CB) | Concept Name |
| Use case모델을 잘 따르고, 전체 컨셉을 잘 실행시킬 수 있다. | D, C | Controller |
| 사용자 정보를 입력받는다. | K, E | KeyboardEntry |
| 사용자 정보가 진실된지 확인한다. | K, E | CheckInfo |
| 사용자 정보를 기반으로 기업을 추천한다. | D, B | Recommendation |
| 통계를 통해 최근 인기 있었던 게시글을 게시한다. | D, B | viewer |
| 사용자가 남기는 후기, 별점을 통계에 집어넣는다. | D, B | DB connection |

**Extracting the Associations**

|  |  |  |
| --- | --- | --- |
| Concept pair | Association description | Association name |
| Controller <-> KeyboardEntry | 컨트롤러는 어떤 사용자 정보가 입력되었는지 확인 받는다. | Requests data |
| KeyboardEntry <-> CheckInfo | CheckInfo 를 통해 keyboardentry의 진실성파악 | Check data |
| KeyboardEntry<-> Recommendation | Recommendation은 keyboardentry를 통해 들어온 데이터에 기반하여 제공된다. | Provides data |
| Recommendation <-> Viewer | Recommendation을 viewer를 위해 보여준다. | Provides data |
| Controller <-> DB connection | 컨트롤러는 후기와 별점을 데이터베이스에 다시 저장한다. | Convey data |

**Extracting the Attributes**

|  |  |  |
| --- | --- | --- |
| Concept | Attributes | Attribute Descrption |
| KeyboardEntry | Key information | Key information of player input keys with keyboard |
| CheckInfo | Verify information | Check that given information is true or not |
| Recommendation | Industry information | Currents data of industry |
| viewer | Statistics information | Currents data of the most hot post |
| DB connection | Preference information | Information of preference |

|  |  |  |
| --- | --- | --- |
| **Use case**  **Concept name** | **Uc1** | **UC2** |
| Controller | **O** | **O** |
| KeyboardEntry | **O** |  |
| CheckInfo | **O** |  |
| Recommendation | **O** | **O** |
| viewer |  | **O** |
| DB connection | **O** |  |

**2nd . Sub System : Industrial Booking System**

* **Requirements**

FR5: 사용자는 메인화면에서 입력도구(마우스or키보드)를 이용해 산업체검색, 마이페이지, 커뮤니티 등의 서비스로 이동할 수 있다.

FR6: 방문자는 산업체의 정보들을 활용한 여러 태그를 통해 산업체를 검색할 수 있다.

FR7: 방문자가 한 산업체를 고르면 산업체의 상세정보, Q&A칸, 좋아요 버튼 등을 볼 수 있다.

FR8: 방문자는 산업체검색 후, 자신이 방문하길 원하는 산업체에 방문을 예약할 수 있다.

FR9: 산업체는 마이페이지를 통해 예약가능한 시간을 변경하거나 신청된 방문자의 예약을 확인할 수 있다.

FR10: 방문자는 자신의 방문예약의 정보를 마이페이지를 통해 확인, 취소, 변경할 수 있다.

FR11: 방문자는 시스템을 통해 예약된 산업체 방문에 대한 알림을 받을 수 있다.

NFR1: 예약의 상세 정보로는 인원과 방문 예정 시각을 포함해야한다.

* **Use Case Model**

**Traceability Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ#** | **PW** | UC1 | UC2 | UC3 | UC4 |
| **FR1** |  | **O** |  |  |  |
| **FR2** |  | **O** |  |  |  |
| **FR3** |  | **O** | **O** |  |  |
| **FR4** |  | **O** |  |  |  |
| **FR5** |  |  |  |  |  |
| **FR6** |  |  | **O** |  |  |
| **FR7** |  |  | **O** |  |  |
| **FR8** |  |  |  | **O** |  |
| **FR9** |  | **O** |  |  | **O** |
| **FR10** |  | **O** |  |  | **O** |
| **FR11** |  |  |  |  | **O** |
| **Max PW** | |  |  |  |  |
| **Total PW** | |  |  |  |  |

**Use Case 1: 산업체 검색**

|  |  |
| --- | --- |
| Use Case UC-2: | 산업체 검색 |
| Related Requirement: | FR6,FR7 |
| Initiating Actor: | 방문자 |
| Actor`s Goal: | * 방문자는 여러 태그를 통해 산업체를 검색할 수 있다. * 검색된 산업체의 상세정보를 확인할 수 있다. |
| Participating Actors: | 산업체 db |
| Preconditions: | * 방문자는 ID와 PW를 통해 서로 구분되며 로그인을 통해 시스템에 접속한다. (UC-1) * 태그로는 직군, 위치, 규모(대기업, 중소기업 등등)이 있다. * 산업체의 상세정보로는 산업체 detail, Q&A, 찜 버튼 등이 있다. |
| Postconditions: | * 검색 후 산업체 db로부터 해당 산업체 정보를 가져오고 방문자는 예약을 할 수 있다. * 추가된 Q&A와 늘어난 찜 수는 산업체 db에 저장된다. |
| Flow of Events for Main Success Scenario: |  |
| ->1. 방문자는 여러 태그를 통해 산업체를 검색한다.  <-2. 방문자는 검색된 산업체를 보고 방문할 산업체를 찾는다.  ->3. 방문자는 산업체를 고른다.  <-4. 방문자는 산업체의 상세정보를 볼 수 있다.  ->5. 방문자는 원하면 Q&A를 추가할 수 있다.  ->3. 방문자는 원하면 찜을 할 수 있다.  5. Include: 산업체 방문예약 (UC-3) | | |

**Use Case 2: 산업체 방문예약**

|  |  |
| --- | --- |
| Use Case UC-3: | 산업체 방문예약 |
| Related Requirement: | FR8 |
| Initiating Actor: | 방문자 |
| Actor`s Goal: | * 방문을 예약 할 수 있다. * 예약의 상세정보를 입력할 수 있다. |
| Participating Actors: | 예약 db 사용자 db |
| Preconditions: | * 검색을 통해 방문자는 방문할 산업체를 고른다. |
| Postconditions: | * 예약된 방문 정보는 db에 저장되고 사용자는 마이페이지에서 예약 정보를 확인할 수 있다. |
| Flow of Events for Main Success Scenario: |  |
| ->1. 방문자는 방문 예약을 한다.  ->2. 방문자는 예약의 상세정보를 입력한다.  3. Include: 데이터입력(UC-1)  <-4. 예약의 정보가 db에 저장되고 마이페이지에 업데이트된다.  5. Include: 마이페이지(UC-4) | | |

**Use Case 3 : 마이페이지**

|  |  |
| --- | --- |
| Use Case UC-4: | 마이페이지 |
| Related Requirement: | FR9, FR10, FR11 |
| Initiating Actor: | 산업체, 방문자 |
| Actor`s Goal: | * 산업체는 방문자의 예약을 확인할 수 있다. * 방문자는 자신의 예약된 방문 정보를 확인 및 변경할 수 있다. * 사용자는 자신의 상세정보를 수정할 수 있다. * 방문자는 예약된 방문 정보를 통해 방문 예약 알림을 받을 수 있다. |
| Participating Actors: | 예약 db 사용자 db |
| Preconditions: | * 방문자는 ID와 PW를 통해 서로 구분되며 로그인을 통해 시스템에 접속한다. (UC-1). |
| Postconditions: | * 예약 db로부터 해당 예약 정보를 가져온다. * 사용자 db와 예약 db에 변경된 정보를 업데이트한다. * 방문 예약의 알림이 설정된다 |
| Flow of Events for Main Success Scenario: |  |
| <-1. 사용자는 방문 예약의 정보를 확인한다.  ->2. 사용자는 자신의 상세정보 또는 예약정보를 수정한다.  ->3. 방문 예약의 알림이 설정된다.  <-4. 변경된 정보를 db에 업데이트한다. | | |

* **Domain model**

Extracting the responsibility

|  |  |  |
| --- | --- | --- |
| Responsibility Description | Type  (D/k,E/CB) | Concept Name |
| 관련된 모든 액션들 조정 | D,C | Controller |
| 사용자로부터 입력을 받고 전달 | D,B | InputEntry |
| 출력를 화면에 표시 | D,B | ServiceDisplay |
| 사용자의 입력 저장 | K,E | UserInput |
| DB로부터 데이터 읽기, 쓰기 | D,B | DBOperator |
| 알람을 울려 예약 시간을 알린다 | D,B | AlarmOperator |
| DB레코드들을 검색조건에 맞게 필터링 | D,C | PostProcessor |
| 사용할 수 있는 산업체들의 태그들 저장 | K,E | SearchHelper |

Extracting the Associations

|  |  |  |
| --- | --- | --- |
| Concept Pair | Association Description | Association Name |
| Controller<->InputEntry | 사용자의 입력을 받고 그에 맞는 작업을 할 수 있도록 한다 | Input receives |
| DBOperator  <->Controller | 데이터를 DB로부터 가져오거나 내보낸다 | DB requests |
| Controller<->ServiceDisplay | 사용자에게 검색 조건에 맞는 정보를 보여준다 | Output sends |
| Controller  <->AlarmOperator | 예약 정보를 알림한다. | Requests alarm |
| Controller<->UserInput | 사용자의 입력이 이동한다. | Conveys data |
| Controller<->postprocessor | 검색 조건과 관련된 db레코들을 찾는다. | Requests filtering |
| Controller<->searchhelper | 사용할 수 있는 산업체들의 태그들을 알아본다. | Requests data |

Extracting the Attributes

|  |  |  |
| --- | --- | --- |
| Concept | Attributes | Attribute Description |
| SearchHelper | 사용할 수 있는  산업체들의 태그 | 사용자가 입력 가능한 산업체들의 태그 |
| AlarmOperator | 예약 정보 | 알림 할 사용자의 예약 정보 |

**Traceability Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case**  **Concept**  **Name** | **UC1** | **UC2** | **UC3** | **UC4** |
| **Controller** | **O** | **O** | **O** | **O** |
| **InputEntry** | **O** | **O** |  | **O** |
| **ServiceDisplay** |  | **O** | **O** | **O** |
| **UserInPut** |  | **O** | **O** |  |
| **DBOperator** | **O** | **O** | **O** | **O** |
| **AlarmOperator** |  |  |  | **O** |
| **PostProcessor** |  | **O** |  |  |
| **SearchHelper** |  | **O** |  |  |

**3rd. Sub System : Web Community site & User Comment Management**

* **Requirements**

FR-1 방문자는 커뮤니티 페이지에서, 후기게시판, 준비게시판, 자유게시판 3개의 게시판 페이지로 들어갈 수 있다

FR-2 유저는 게시판( 3종류중 하나)에 자신의 게시글을 작성,수정,삭제 할 수 있어야 한다

FR-3 방문자가 다른 유저가 작성한 게시물에 댓글을 게시할 수 있어야 한다

FR-4 유저는 커뮤니티에서 다른 유저들이 작성한 게시물에 댓글을 달 수 있다.

FR-5 유저는 커뮤니티의 게시글들에 대한 조회수를 확인할 수 있다.

FR-6 모든 종류의 댓글에는, 대학, 학부 정보가 보여져야 한다

FR-7 유저는 커뮤니티 게시글이 커뮤니티 운영 수칙을 위반한 경우 이를 신고할 수 있다.

FR-8 산업체 패키지 상세 예약 페이지 하단에, 해당 패키지 에 대한 후기 댓글을 작성, 수정, 삭제할 수 있어야 한다

FR-9 메인페이지와 커뮤니티 페이지 사이에 버튼 및 이동 링크가 존재해야 한다

FR-10 방문자는 키워드를 통해 게시글을 검색할 수 있다.

* **Use case model**

**Traceability Matrix**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **REQ#** | **PW** | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** |
| **FR1** |  | **O** |  |  |  |  |
| **FR2** |  |  |  | **O** |  |  |
| **FR3** |  |  |  |  | **O** |  |
| **FR4** |  |  |  | **O** |  |  |
| **FR5** |  |  |  | **O** |  |  |
| **FR6** |  |  |  | **O** |  |  |
| **FR7** |  |  |  | **O** |  |  |
| **FR8** |  |  |  |  |  | **O** |
| **FR9** |  | **O** |  |  |  |  |
| **FR10** |  |  | **O** |  |  |  |

Use case 1: 커뮤니티 메인메뉴 보여주기

|  |  |
| --- | --- |
| Use Case UC-1: | 메인메뉴 보여주기 |
| Related requirements | FR-1, FR-9 |
| Initiating actor | 방문자 |
| Actor’s goal | * 방문자는 각 커뮤니티의 대표 상단 글( 최신순 )을 3개씩 볼 수 있다 * 방문자는 각 게시판의 주요 게시글 목록 중 하나를 선택해서 해당 게시물을 열람할 수 있다. * 방문자는 메인 페이지에서 게시판 하나를 선택해서, 해당 게시판에 대한 메뉴로 넘어갈 수 있다. |
| Participating Actors | 게시글에 관한 db,  커뮤니티 페이지 |
| Preconditions | * 후기게시판, 준비게시판, 자유게시판 페이지가 별도로 존재해야 한다. * 메인페이지에서 커뮤니티 페이지로 가는 버튼 및 링크가 존재해야 한다. * 방문자가 특정 게시물을 검색할 수 있는 검색창이 존재해야 한다 |
| Postconditions | * 커뮤니티에서, 메인 페이지로 가는 버튼 및 링크도 존재해야 한다. * 클릭한 특정 게시판 페이지로 이동해야 한다 |
| Flow of Events for Main Success Scenario  ->1. 방문자가 메인 페이지에서 커뮤니티 버튼을 통해 커뮤니티 메뉴로 접속한다.  <-2. System은 “각 커뮤니티의 주요 글들 중 3개”와 관련된 DB query를 준비한다.  ->3. DB는 해당 query와 맞는 record, 기록들을 return 해준다  <-4. 커뮤니티 페이지는 각 커뮤니티의 주요 글들 중 3개를 디스플레이( rendering )한다.  <-5. 커뮤니티 페이지는 방문자가 검색할 수 있도록 하는 검색 Inut 창 또한 디스플레이한다. | |

Use case 2: 커뮤니티 게시물 검색

|  |  |
| --- | --- |
| Use Case UC-2: | 커뮤니티 게시물 검색 |
| Related requirements | FR-10 |
| Initiating actor | 방문자 |
| Actor’s goal | * 방문자는 조회하고자 하는 게시물에 대한 내용을 키워드로 하여 검색하여 그 키워드에 부합하는 게시물들을 조회할 수 있다. |
| Participating Actors | 게시글에 관한 DB  커뮤니티 페이지 |
| Preconditions | * 방문자가 메인 페이지 혹은 각 커뮤니티의 페이지로 접속해 있는 상태이어야 한다. |
| Postconditions | * 검색 키워드에 맞는 게시물들이 보여져야 한다. |
| Flow of Events for Main Success Scenario  ->1. 방문자가 커뮤니티 메인 페이지 혹은 각 커뮤니티 게시판에 대한 페이지에 있는 검색창을 활용하여 검색 키워드를 입력한다.  <-2. System은 “사용자가 검색한 게시물 제목”과 관련된 DB query를 준비한다 .  ->3. DB는 해당 query와 맞는 record, 기록들을 return 해준다 .  <-4. 방문자가 입력한 키워드와 일치하는 내용 혹은 제목을 가지고 있는 게시글들에 대한 목록이 나타난다. 이 때 커뮤니티 메인 페이지에서 검색했을 경우 모든 커뮤니티 게시판에서 해당 키워드와 일치하는 내용을 가진 게시물들이 나타나며, 각각의 커뮤니티 페이지에서 검색했을 경우 그 커뮤니티의 게시물에 해당하면서 방문자가 입력한 키워드와 일치하는 내용을 가진 게시물들이 나타난다. | |

Use case 3: 커뮤니티 게시물 관리

|  |  |
| --- | --- |
| Use Case UC-3: | 커뮤니티 게시물 관리 |
| Related requirements | FR-2, FR-3, FR-4, FR-5, FR-6, FR-7 |
| Initiating actor | 방문자 |
| Actor’s goal | * 방문자는 내용을 보고자 하는 게시물을 클릭하여 그 게시물의 댓글, 내용, 제목 등을 조회할 수 있다. * 방문자는 자신의 게시물에 대한 제목, 내용을 작성할 수 있다. * 방문자는 자신의 게시물을 작성, 수정 혹은 삭제할 수 있다. |
| Participating Actors | 게시글에 관한 DB  커뮤니티 페이지 |
| Preconditions | * 방문자가 조회하고자 하는 게시물을 클릭한 상태이다. |
| Postconditions | * 클릭한 게시물에 해당하는 페이지로 이동해야 한다 * 작성된 게시물이 db에 저장된다. |
| Flow of Events for Main Success Scenario1 : ( read )  ->1. 방문자가 게시글 목록 중 조회하고자 하는 게시글을 클릭한다.  <-2. 해당 게시글의 제목, 내용 및 댓글 목록들, 조회수가 나타난다.  Flow of Events for Main Success scenario2 : ( create )  ->1. 방문자가 메인 페이지 혹은 각 커뮤니티 페이지에서 글 작성하기 버튼을 클릭한다.  ->2. 방문자가 글을 작성한다.  <-3. 작성한 글이 반영되어 해당 커뮤니티의 게시물 목록에 나타난다.  Flow of Events for Main Success scenario 3 : ( delete )  ->1.b 방문자가 삭제하고자 하는 게시물을 마이페이지에서 선택한다.  ->2. 방문자가 게시물의 삭제 버튼을 누른다.  <-3. db에서 해당 게시물의 내용이 삭제된다.  Flow of Events for Main Success scenario 4 : ( update )  ->1.c 방문자가 수정하고자 하는 게시물을 마이페이지에서 선택한다.  ->2. 방문자가 게시물의 수정하기 버튼을 누른 후 게시물의 내용 혹은 제목을 수정한다.  <-3. 수정된 게시물이 게시글 목록에 나타난다. | |

Use case 4 : 커뮤니티 게시물 댓글 관리

|  |  |
| --- | --- |
| Use Case UC-5: | 커뮤니티 게시물 댓글 관리 |
| Related requirements | FR-2 |
| Initiating actor | 방문자 |
| Actor’s goal | * 방문자는 자신의 게시물 혹은 다른 방문자의 게시물에 댓글을 작성, 수정, 삭제할 수 있다. |
| Participating Actors | 게시글에 관한 DB  커뮤니티 페이지 |
| Preconditions | * 방문자가 커뮤니티 페이지에서 특정 게시물을 보고 있는 상태이다 |
| Postconditions | * 작성,변경,삭제된 댓글 사항이 db에 저장된다. |
| Flow of Events for Main Success scenario1 : ( create )  ->1. 방문자가 특정 게시글에서 댓글 작성하기 버튼을 클릭한다.  ->2. 방문자가 댓글을 작성한다.  <-3. 작성한 댓글이 반영되어 해당 게시물의 댓글 목록에 나타난다.  Flow of Events for Main Success scenario 2 : ( delete )  ->1.b 방문자가 삭제하고자 하는 댓글을 마이페이지, 혹은 특정 게시글 페이지에서 선택한다.  ->2. 방문자가 댓글의 삭제 버튼을 누른다.  <-3. db에서 해당 댓글의 내용이 삭제된다.  Flow of Events for Main Success scenario 3 : ( update )  ->1.c 방문자가 수정하고자 하는 댓글을 마이페이지에서 혹은 특정 게시글 페이지에서 선택한다.  ->2. 방문자가 댓글의 수정하기 버튼을 누른 후 댓글의 내용 혹은 제목을 수정한다.  <-3. 수정된 댓글이 게시글 댓글 목록에 나타난다 | |

Use case 5 : 산업체 패키지 후기 및 댓글 관리

|  |  |
| --- | --- |
| Use Case UC-5: | 산업체 패키지 후기 및 댓글 관리 |
| Related requirements | FR-8 |
| Initiating actor | 방문자 |
| Actor’s goal | * 방문자는 산업체 패키지에 대한 후기를 조회,작성,변경,삭제 할 수 있다 * 방문자는 산업체 패키지에 대한 댓글을 조회,작성,변경,삭제 할 수 있다. |
| Participating Actors | 산업체 패키지에 관한 DB  산업체 패키지 상세 페이지 |
| Preconditions | * 방문자가 산업체 패키지 상세 페이지에서 해당 패키지 내용을 보고 있는 상태이다 |
| Postconditions | * 작성,변경,삭제된 댓글 사항이 db에 저장된다. * 작성,변경,삭제된 후기 사항이 db에 저장된다. |
| Flow of Events for Main Success scenario1 : ( create )  ->1. 방문자가 특정 산업체 패키지 페이지 하단 후기 및 댓글 작성하기 버튼을 클릭한다.  ->2. 방문자가 후기 및 댓글을 작성한다.  <-3. 작성한 후기 및 댓글이 반영되어 산업체 패키지 페이지 하단 후기 및 댓글 목록에 나타난다.  Flow of Events for Main Success scenario 2 : ( delete )  ->1.b 방문자가 특정 산업체 패키지 페이지 하단 후기 및 댓글 삭제하기 버튼을 클릭한다.  ->2. 방문자가 후기 및 댓글의 삭제 버튼을 누른다.  <-3. db에서 해당 댓글 및 후기 내용이 삭제된다.  Flow of Events for Main Success scenario 3 : ( update )  ->1.c 방문자가 수정하고자 하는 후기 및 댓글 을 마이페이지에서 선택한다.  ->2. 방문자가 게시물의 수정하기 버튼을 누른 후 (후기/ 댓글) 의 내용 혹은 제목을 수정한다.  <-3. 수정된 ( 후기 / 댓글 )이 특정 산업체 패키지 페이지 하단 후기 및 댓글 목록에 나타난다 | |

* **Domain Model**

**Extracting the responsibility**

|  |  |  |
| --- | --- | --- |
| **Responsibility Description** | **Type**  **(D/k,E/CB)** | **Concept Name** |
| 관련된 모든 액션들 조정 | D,C | Controller |
| 사용자로부터 입력을 받고 전달 | D,B | InputEntry |
| 출력를 화면에 표시 | D,B | ServiceDisplay |
| 사용자의 입력 저장 | K,E | UserInput |
| DB로부터 데이터 읽기, 쓰기 | D,B | DBOperator |
| DB레코드들을 검색조건에 맞게 필터링 | D,C | PostProcessor |
| 사용자의 개인 정보 표시 | K,B | PersonalInfoDisplay |
| 게시물 및 댓글 신고 | D,B | ReportUser |
| 사용자 페이지 이동 시키기 | D,B | RedirectUser |

**Extracting the Associations**

|  |  |  |
| --- | --- | --- |
| **Concept Pair** | **Association Description** | **Association Name** |
| Controller<->InputEntry | 사용자의 입력을 받고 그에 맞는 작업을 할 수 있도록 한다 | Input receives |
| DBOperator  <->Controller | 데이터를 DB로부터 가져오거나 내보낸다 | DB requests |
| Controller <-> RedirectUser | Hyperlink 및 특정 html element를 클릭하면, 새로운 페이지로 이동시킨다 | Redirect Page |
| Controller <-> ReportUser | 특정 게시물 및 댓글을 신고한다 | Report User |
| Controller <-> PersonalInfoDisplay | 게시글 및 댓글에 작성자의 학교, 학과를 표시한다 | Display User Info |
| Controller<->ServiceDisplay | 사용자에게 검색 조건에 맞는 정보를 보여준다 | Output sends |
| Controller<->UserInput | 사용자의 입력이 이동한다. | Conveys data |
| Controller<->postprocessor | 검색 조건과 관련된 db레코들을 찾는다. | Requests filtering |

**Extracting the Attributes**

|  |  |  |
| --- | --- | --- |
| **Concept** | **Attributes** | **Attribute Description** |
| InputEntry | 검색 파라미터( Parameter ) | 검색을 위한 , 검색 input 정보 |
| ReportUser | 신고한 사람, 신고당한 사람의 identity | 신고한 사람의 id 및 정보  신고당한 사람의 id 및 정보 |
| DBOperator | 데이터베이스 query | DB를 읽고 쓰기 위한 DB Queries |
| PersonalInfoDisplay | 사용자 정보 | 데이터베이스에 저장된, 게시물 및 댓글을 작성한 사용자의 학교,학번 정보 |

**Traceability Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case  Concept  Name | UC1 | UC2 | UC3 | UC4 | UC5 |
| Controller | O | O | O | O | O |
| InputEntry |  | O | O | O | O |
| ServiceDisplay | O | O | O | O | O |
| UserInPut |  | O | O | O | O |
| DBOperator | O | O | O | O | O |
| PostProcessor | O | O |  |  |  |
| PersonalInfoDisplay |  |  | O | O | O |
| ReportUser |  |  | O | O | O |
| RedirectUser | O | O | O | O | O |