1. Project Team

프로젝트 명: 먹구름

팀원: 채승표 :

학번: 20191490

전공: 전자공학부 전자시스템 전공

업무 경험: 유니티로 게임 제작 경험, 점자를 사진으로 찍어 번역해주는 어플 개발

강점 소개: C++을 잘 다룬다.

홍진우

학번: 20191497

전공: 전자공학부 전자시스템 전공

업무 경험: 유니티로 게임 제작 경험, 웹 사이트 제작 경험

강점 소개: 모르는 부분이 있으면 알 때까지 공부한다.

2. Project Vision

현재 위치의 날씨정보를 기반으로 음식 및 가까운 음식점 추천해주는 어플 개발

Project Scope

안드로이드 앱을 만들어서 실제로 사용 가능하도록 구현하는 것이 목표.

3. Functional Requirements

Use Case Description

|  |  |  |
| --- | --- | --- |
| **Use case name** | 1.사용자 회원가입 | |
| **Scenario** | 사용자가 서비스를 이용하기 위해 계정을 생성한다. | |
| **Triggering event** | 사용자가 최초로 앱을 사용 | |
| **Brief description** | 사용자가 사용자의 정보(개인정보, 선호 음식 등)를 입력한다. 시스템은 사용자의 정보를 데이터 베이스에 저장한다. | |
| **Actors** | 사용자 | |
| **Related use cases** |  | |
| **Stakeholders** |  | |
| **Preconditions** | 사용자가 앱을 최초로 사용한다. | |
| **Post conditions** | 사용자의 정보가 데이터 베이스에 저장된다. | |
| **Flow of activities** | Actor | System |
| 1. 사용자가 자신의 정보(개인정보, 선호 음식 등) 입력 | 1.1 시스템이 데이터 베이스에 정보 저장 |
| **Exception conditions** | 1. 이미 사용자 명의로 생성된 계정이 있으면 계정이 존재한다고 알리기  2. 생성하려는 계정이 이미 존재한다면 같은 계정이 존재한다고 알리기 | |

|  |  |  |
| --- | --- | --- |
| **Use case name** | 2.사용자의 현재 위치 받아오기 | |
| **Scenario** | 시스템이 사용자 주변의 식당을 고르기 위해 사용자의 위치를 받아온다. | |
| **Triggering event** | 사용자가 앱을 실행 | |
| **Brief description** | 사용자가 앱을 실행한다. 시스템은 사용자의 위치를 받아온다. | |
| **Actors** | 사용자 | |
| **Related use cases** | 없음 | |
| **Stakeholders** | GPS | |
| **Preconditions** | 시스템이 사용자의 위치 정보를 모른다. | |
| **Post conditions** | 시스템이 사용자의 위치 정보를 안다. | |
| **Flow of activities** | Actor | System |
| 1. 사용자가 앱을 실행한다. | 1.1 시스템은 GPS 신호를 받아온다.  1.2 시스템은 GPS 신호를 통해 사용자의 위치 정보를 얻는다. |
| **Exception conditions** | 1. 사용자의 GPS가 꺼져 있다면, 키라고  2. 생성하려는 계정이 이미 존재한다면 같은 계정이 존재한다고 알리기 | |

|  |  |  |
| --- | --- | --- |
| **Use case name** | 3. 현재 위치의 날씨 정보 받아오기 | |
| **Scenario** | 사용자의 위치를 기반으로 기상청에서 날씨 정보를 가져온다. | |
| **Triggering event** | 시스템이 사용자의 위치 정보를 받아온다. | |
| **Brief description** | 사용자의 위치 정보를 기반으로 기상청 api에서 날씨 정보를 받아온다. | |
| **Actors** |  | |
| **Related use cases** | 2. 사용자의 현재위치 받아오기 | |
| **Stakeholders** | 기상청 | |
| **Preconditions** | 사용자의 위치 정보를 가지고 있다; 해당 위치의 날씨 정보를 모른다. | |
| **Post conditions** | 시스템이 해당 위치의 날씨 정보를 안다. | |
| **Flow of activities** | Actor | System |
|  | 1.1 시스템은 기상청 api를 통해 날씨 정보를 받아온다. |
| **Exception conditions** | 사용자의 위치 정보를 확인할 수 없으면 다시 2번으로 돌아간다. | |

|  |  |  |
| --- | --- | --- |
| **Use case name** | 4. 날씨와 사용자 정보를 기반으로 음식 선정하기 | |
| **Scenario** | 날씨와 사용자 정보를 기반으로 음식을 추천한다. | |
| **Triggering event** | 사용자가 앱을 실행한다. | |
| **Brief description** | 날씨에 어울리는 음식을 사용자의 선호 정도에 따라서 나열한다. | |
| **Actors** | 사용자 | |
| **Related use cases** | 1. 사용자 회원가입; 3. 현재 위치의 날씨 정보 받아오기 | |
| **Stakeholders** | 없음 | |
| **Preconditions** | 사용자의 음식 선호도가 존재해야 한다; 현재 위치의 날씨 정보가 존재해야 한다. | |
| **Post conditions** | 추천 음식 리스트를 얻는다; | |
| **Flow of activities** | Actor | System |
|  | 1.1 날씨에 어울리는 음식을 선정한다.  1.2 사용자 선호도에 따라 음식을 나열한다. |
| **Exception conditions** | 날씨 정보가 없다면 다시 3번으로 돌아간다. | |

|  |  |  |
| --- | --- | --- |
| **Use case name** | 5. 추천한 음식 정보 및 식당을 보여주기 | |
| **Scenario** | 반경 N km내에서 존재하는 추천한 음식 정보와 평점이 높은 식당 여러 개를 보여준다. | |
| **Triggering event** | 사용자가 앱을 실행한다. | |
| **Brief description** | 추천한 음식과 위치 정보를 기반으로 검색 API를 사용하여 주변에 있는 식당을 검색한다.  평점이 높은 가게를 추천한다. | |
| **Actors** | 사용자 | |
| **Related use cases** | 4. 날씨와 사용자 정보를 기반으로 음식을 선정한다. | |
| **Stakeholders** | 검색 사이트, 음식점 | |
| **Preconditions** | 위치 정보와 추천 음식 리스트를 보유하고 있다. | |
| **Post conditions** | 추천하는 음식 및 식당 리스트를 생성한다. | |
| **Flow of activities** | Actor | System |
| 1.4 사용자가 날씨에 어울리는 음식 및 식당을 추천받는다. | 1.1 정보를 바탕으로 검색 API를 사용하여 음식점 검색  1.2 평점이 높은 식당 여러 개를 추출  1.3 사용자에게 제공 |
| **Exception conditions** | 추천한 음식 식당이 주변에 없을 경우, 다른 음식을 검색한다. | |

Use Case Diagram

텍스트, 도표, 스크린샷, 원이(가) 표시된 사진

자동 생성된 설명

4. Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| 해당 Use Case | Non-Functional Requirement | Quality | Quality Attributes |
| 4. 날씨와 사용자 정보를 기반으로 음식 추천하기 | 사용자가 선호할 만한 음식을 추천한다. | Reliability | 사용자 만족도를 90%이상으로 목표한다. |
| 5. 추천한 음식 정보 및 식당을 보여주기 | 음식과 식당 정보가 화면에 빠르게 나오도록 한다. | Performance Efficiency | 평균 1초만에 화면에 나타나도록 한다. |

4. Domain Model 분석하기

|  |  |  |
| --- | --- | --- |
| Class 명 | Concept 설명 | 주요 attributes |
| Manage\_account | 계정 생성, 로그인 처리를 하는 Controller | 사용자 정보(개인정보, id/pw등) |
| Now\_Weather | 현재 위치 받고, 위치를 기반으로 기상청에서 날씨 정보를 받아온다. | Weather, now\_position |
| Food\_recommand | 날씨 정보와 현재 위치를 기반으로 음식 추천한다. | Food\_namd, Food\_description |
| Restaurant\_recommand | 추천한 음식을 기반으로 음식점을 추천한다. | Restaurant\_name,  Restaurant\_description |

5. Design Model 설계하기

5-1) System Sequence Diagram 그리기

텍스트, 도표, 번호, 라인이(가) 표시된 사진

자동 생성된 설명

5-2) Operation Contract

|  |  |
| --- | --- |
| Contract C01: Create\_account | |
| Operation | Create\_account() |
| Cross Reference | Use Case: 사용자 회원가입 |
| Preconditions | None |
| Postconditions | 계정이 생성된다. |

|  |  |
| --- | --- |
| Contract C02: Login | |
| Operation | Login() |
| Cross Reference | Use Case: 사용자 회원가입 |
| Preconditions | 계정이 존재해야 한다. 로그인이 안된 상태여야 한다. |
| Postconditions | 로그인이 된다. |

|  |  |
| --- | --- |
| Contract C03: Get\_Position | |
| Operation | Get\_Position() |
| Cross Reference | Use Case: 사용자의 현재위치 받아오기 |
| Preconditions | None |
| Postconditions | 현재 위치를 받아온다. |

|  |  |
| --- | --- |
| Contract C04: Get\_Weather | |
| Operation | Get\_Weather() |
| Cross Reference | Use Case: 날씨 정보 받아오기 |
| Preconditions | 사용자의 현재 위치를 알고 있어야 한다. |
| Postconditions | 현재 위치의 날씨 정보를 받아온다. |

|  |  |
| --- | --- |
| Contract C05: Get\_Food | |
| Operation | Get\_Food(position:Now\_Weather) |
| Cross Reference | Use Case: 추천 음식 선정하기 |
| Preconditions | 날씨 정보가 있어야 한다. |
| Postconditions | 날씨에 어울리는 음식을 선정한다. |

|  |  |
| --- | --- |
| Contract C06: Get\_Restaurant | |
| Operation | Get\_Restaurant(Food:Food\_Recommand) |
| Cross Reference | Use Case: 음식 정보 및 식당 보여주기 |
| Preconditions | 날씨에 어울리는 추천 음식, 현재 위치가 있어야 한다. |
| Postconditions | 현재 위치 근처의 추천 음식점을 보여준다. |

5-3) Sequence Diagram

텍스트, 도표, 번호, 라인이(가) 표시된 사진

자동 생성된 설명

5-4) Class Diagram

텍스트, 폰트, 라인, 도표이(가) 표시된 사진

자동 생성된 설명

6. Design Refinement

6-1) Class Diagram텍스트, 도표, 평면도, 평행이(가) 표시된 사진

자동 생성된 설명

6-2) Design Refinement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Before classfier | After classfier | 적용한 설계 개념 | Architecture Design | NFR과 QA에 대한 영향 |
| Communication | Interface, Classes, Controller | GRASP | 저장된 데이터베이스에 접근하기 때문에 외부에서 접근하지 못하도록 함. | 음식을 불러오는 속도에는 영향이 없고 1초 안에 불러옴. |
| NaverSearchApi | Interface, Classes, Controller | GRASP | Api키를 접근하지 못하도록 하고, 이번트 처리를 효율적으로 하도록 함. | 응답시간이 느려지지 않음. |
| NowPosition | Classes, Singleton | GOF | 다른 클래스들에서 공통적으로 필요한 현재 좌표를 싱글톤객체로 만들어 중복을 방지함. | NONE |
| WeatherApi | Interface, Classes, Controller, Singleton | GOF, GRASP | 싱글톤 객체로 만들어 중복을 방지함.  이벤트 처리를 해당 객체에서 처리함. | NONE |

7. 구현결과 Snapshot

텍스트, 스크린샷, 휴대 전화, 모바일 기기이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 휴대 전화, 모바일 기기이(가) 표시된 사진

자동 생성된 설명1. 사용자 회원가입

사용자 id를 입력하고 중복확인을 누르면, 데이터베이스에 저장된 id와 대조하여 중복된 id가 있는지 확인하고, 중복이 없다면 비밀번호와 선호 음식 종류를 작성하고 게정 생성을 누르면 회원가입이 완료된다.

텍스트, 휴대 전화, 모바일 기기, 정보기기이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 휴대 전화, 정보기기이(가) 표시된 사진

자동 생성된 설명2. 사용자 로그인

회원가입 된 id로 로그인을 진행하여 id와 password가 모두 일치한다면 로그인에 성공하고 음식 추천 화면으로 들어간다.

2-1 사용자 위치 받아오기 및 해당 위치의 날씨 정보 확인하기

텍스트, 폰트, 시계, 번호이(가) 표시된 사진

자동 생성된 설명사용자의 위치를 받아오는 모습을 볼 수 있다. 또한 해당 위치를 기반으로

해당스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명 위치의 날씨 정보를 받아오는 것을 알 수 있다. 여기서 하늘의 값은 0~5는 맑음, 6~8은 구름 많음, 9~10은 흐림으로 볼 수 있다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 정보기기, 모바일 기기, 휴대 전화이(가) 표시된 사진

자동 생성된 설명3. 음식 추천하기

메인 화면에서 ‘음식 추천받기’ 버튼을 누르면 현재 날씨 정보에 따른 음식 종류(양식, 크림리조또)가 선정된다. 그 후에 해당 지역에서 선정한 음식점을 네이버 검색 api를 이용해서 찾는다. 그 후 찾은 결과가 다음과 같이 나오는 것을 볼 수 있고,

텍스트, 휴대 전화, 스크린샷, 모바일 기기이(가) 표시된 사진

자동 생성된 설명3-1) 음식점 추천 화면

다음과 같이 음식점의 이름과 음식점 사이트, 음식점의 사진을 확인할 수 있다.