

아그리콜라 요구사항 명세서

(Agricola Requirement Specification)

2023년 00월 00일

한양대학교 ERICA 소프트웨어학부

민준수

조재훈

조재영

장현수

최윤희

목차

1. 서론 (Introduction)	4
가. 목적 (Purpose)	4
나. 범위 (Scope)	4
다. 용어 및 약어 정의 (Definitions of Terms and Abbreviations)	4
라. 개요 (Overview)	5
2. 제품 설명 (Product Description)	5
가. 프로젝트 결과물 전망 (Project Deliverables Outlook)	5
1) 시스템 인터페이스 (System Interface)	5
2) 사용자 인터페이스 (User Interface)	5
3) 하드웨어 인터페이스 (Hardware Interface)	5
4) 소프트웨어 인터페이스 (Software Interface)	6
5) 통신 인터페이스 (Communication Interface)	6
나. 결과물의 기능 (Product Functionality)	6
다. 사용자 특성 (User Characteristics)	7
라. 제약조건 (Constraints)	8
마. 가정 (Assumptions)	8
3. 상세 요구사항 (Detailed Requirements)	8
가. UI 요구사항 (UI Requirements)	9
나. 요구사항 명세표 (Requirements Specification Table)	10
다. 데이터 요구사항 (Data Requirements)	16
라. 성능 요구사항 (Performance Requirements)	19
마. 논리적 데이터베이스 요구사항 (Logical Database Requirements)	19
바. 코딩 컨벤션 (Coding Conventions)	19
사. 소프트웨어 시스템 특성 (Software System Characteristics)	22

4. 추가 정보 (Additional Information).....	23
가. 소프트웨어 요구사항 명세서 (SRS)	23
나. 문서 이력 (Document History).....	23

1. 서론 (Introduction)

가. 목적 (Purpose)

본 문서는 '아그리콜라 온라인' 서비스를 위한 소프트웨어 요구사항명세서 (SRS)이다. 본 서비스는 S사 B2B 부서에서 설계 및 구현하기 위한 것으로 이를 위한 요구 사항을 정리, 분석하고, 기재된 내용을 바탕으로 '아그리콜라 온라인' 서비스를 설계 및 구현한다.

본 문서는 S사 B2B 부서의 개발팀 및 관련된 BGK사의 모든 직원을 주요 독자로 한다.

나. 범위 (Scope)

본 프로젝트의 주요 기능 개발 범위는 다음과 같다.

- 로그인 및 회원가입 기능
- 게임 로비 및 방 생성/참가 기능
- 플레이어 간 채팅 기능
- 프로필 정보 확인 기능
- 게임 시작 설정 및 라운드 진행 기능
- 직업/설비 카드 관련 기능
- 플레이어의 방 관련 기능
- 가축 관련 기능
- 농사 관련 기능
- 가족 구성원 관련 기능
- 점수 계산 기능
- 구걸 기능

다. 용어 및 약어 정의 (Definitions of Terms and Abbreviations)

.

.

.

라. 개요 (Overview)

본 문서의 구성은 다음과 같다. 2장에서는 ‘아그리콜라 온라인’ 서비스의 제품 설명 및 전반적인 개요를 알아본다. 그를 위해 프로젝트 결과물의 전망과 결과물의 세부 기능, 사용자의 특성, 제약 조건, 가정과 의존성 등의 각 절로 나누어 살펴본다. 3장에서는 ‘아그리콜라 온라인’ 서비스의 상세 요구사항을 알아본다. UI 요구사항, 기능 및 비기능적 요구사항, 데이터 요구사항, 성능 요구사항, 논리적 데이터베이스 요구사항, 코딩컨벤션, 소프트웨어 시스템 특성, 기타 요구사항을 각 절에서 살펴본다. 기능 및 비기능적 요구사항에서는 요구사항 명세표를 포함하며 데이터 요구사항에서는 데이터 흐름도와 ERD 다이어그램을 포함한다.

2. 제품 설명 (Product Description)

가. 프로젝트 결과물 전망 (Project Deliverable Outlook)

1) 시스템 인터페이스 (System Interface)

서버는 클라우드 호스팅 플랫폼에서 실행되며 HTTPS 프로토콜을 사용하여 인터넷을 통해 클라이언트와 통신한다. 서버는 Django 프레임워크를 사용하여 HTTP 요청을 처리하고 socketio를 사용하여 클라이언트와의 실시간 통신을 용이하게 한다. 서버는 또한 데이터베이스 관리 시스템과 인터페이스하여 사용자 정보와 채팅 로그를 저장한다.

2) 사용자 인터페이스 (User Interface)

서버의 인터페이스는 시스템 관리자에게 제공된다. 기본적인 키보드와 마우스, 모니터를 이용한 인터페이스가 제공되며, 시스템 전체의 정보와 서비스 정보, 자원 정보, 콘텐츠 정보를 등록, 삭제, 유지 관리를 지원한다. 클라이언트의 경우 키보드와 마우스를 이용하여 사용자의 입력을 받고, 모니터 화면을 통해 출력한다.

3) 하드웨어 인터페이스 (Hardware Interface)

- 서버 스펙
 - 프로세서: Intel Core i5 또는 i7(또는 Apple M1 칩)
 - 램: 8GB 이상
 - 저장공간: 256GB SSD 이상

- 네트워크 인터페이스: Wi-Fi 또는 이더넷
- 운영 체제: macOS
- 클라이언트 스펙
 - 프로세서: Intel Core i3 또는 동급
 - 램: 4GB 이상
 - 저장공간: 128GB SSD 이상
 - 네트워크 인터페이스: Wi-Fi 또는 이더넷
 - 운영 체제: macOS

4) 소프트웨어 인터페이스 (Software Interface)

- 서버 기술 스택 (Socket IO에서 사용하는 브라우저의 환경)
 - 운영 체제: macOS(최신 버전)
- 클라이언트 기술 스택
 - 웹 브라우저: Safari, Google Chrome, Mozilla Firefox, Microsoft Edge(최신 버전)
 - 운영 체제: macOS(최신 버전)

5) 통신 인터페이스 (Communication Interface)

- 서버 : 100Mbps Ethernet
- 클라이언트 : 802.11a/b/g wireless network

나. 결과물의 기능 (Product Functionality)

- 로그인 및 회원가입 기능 : 본 서비스는 가입된 유저를 대상으로 제공된다. 따라서 유저에게 회원가입 기능을 제공하고 이에 따라 로그인을 통해 게임에 접속할 수 있도록 한다.
- 게임 로비 및 방 생성/참가 기능 : 본 서비스는 서비스를 이용하는 유저들에게 멀티플레이 기능을 제공하므로 사용자간의 게임 플레이를 위한 방 만들기 및 참여기능과 함께 로비기능을 제공한다.
- 플레이어 간 채팅 기능 : 게임 플레이 중 플레이어간의 소통이 가능하도록 채팅 기능을 제공한다.

- 프로필 기능 : 각 유저는 닉네임, 이미지, 게임 승/패 횟수 등 자신의 정보들이 기록되어 모든 유저들이 확인할 수 있는 기능을 제공한다.
- 게임 시작 설정 및 라운드 진행 기능 : 게임이 시작되면 게임 화면 설정 및 원활한 게임 진행을 위한 초기 설정 및 플레이어들의 자원 설정 등이 이루어진다. 이는 라운드 진행 기능, 수확 단계 추가 기능, 턴 진행, 선 플레이어 결정 기능들을 포함한다.
- 직업/설비 카드 관련 기능 : 직업 카드, 주요 설비 및 보조 설비 카드들의 기능을 구현하고 게임 진행시 카드 구매 및 활성화 기능을 제공한다.
- 플레이어의 방 관련 기능 : 게임 진행시 각 플레이어의 방을 구매하여 확장하고 모든 방을 업그레이드하는 기능을 제공한다.
- 가축 관련 기능 : 울타리 치기 및 외양간 설치 기능과 해당하는 공간에 가축을 기르는 기능을 제공하며 추가로 특정 단계에서 일정 조건을 만족하면 가축이 번식할 수 있도록 하는 기능을 제공한다.
- 농사 관련 기능 : 밭 일구기를 통한 농지 확장 기능과 씨 뿌리기를 통한 곡식과 채소 심기 및 수확 단계에서의 수확 기능을 제공한다.
- 가족 구성원 관련 기능 : 가족 구성원의 행동 진행 화면에서의 행동 기능과 가족 구성원 놀리기 기능 및 신생아 기능, 가족 먹여살리기 단계에서 가족 구성원 수에 맞춰 음식을 먹여야 하는 기능을 제공한다.
- 점수 계산 기능 : 게임이 종료되면 각 플레이어별 해당 게임에서의 분야별 점수와 이를 합산한 총 점수를 계산해 제공하고 그에 따른 순위를 판단해주는 기능을 제공한다.
- 구걸 기능 : 게임 진행시 플레이어가 특정 행동을 할 때 행동에 필요한 자원이 부족한 경우 구걸 기능을 통해 일회성으로 자원을 빌릴 수 있도록 하는 기능을 제공한다.

다. 사용자 특성 (User Characteristics)

- 서비스 관리자는 충분히 시스템에 대한 정보를 습득하고, 시스템 전반에 대한 이해를 갖춘 사람으로 한정한다. 시스템 문제 발생 시 이에 대처할 수 있는 충분한 능력을 갖추고 있다고 가정한다. 컴퓨터공학 또는 그 유사한 학문을 전공하거나 그에 상응하는 자격을 갖추고 있다고 가정한다.
- 아그리콜라 온라인 사용자는 보드게임에 대한 관심도가 높으며 보편적인 인기 보드게임을 일정한 이상 플레이 해본 경험이 있는 사람으로써 아그리콜라 보드게임의 기본적인 내용 및 규칙들을 쉽게 파악할 수 있는 능력을 갖추고 있으며, 아그리콜라 게임을 온라인으로 이용할 때에 진행에 무리가 없는 정도의 학습능력을 갖추고 있는 사람이라 가정한다. 일반적으로 13 세 이상 65세 이하의 대한민국 국민을 클라이언트 사용자로 가정한다.

라. 제약조건 (Constraints)

제품설명서의 가절에서 언급한 내용을 바탕으로 시스템을 설계 및 구현한다. 그 이외의 사항들은 개발자가 선호하는 방향을 선택하여 설계 및 구현하되, 단 다음의 항목을 준수한다.

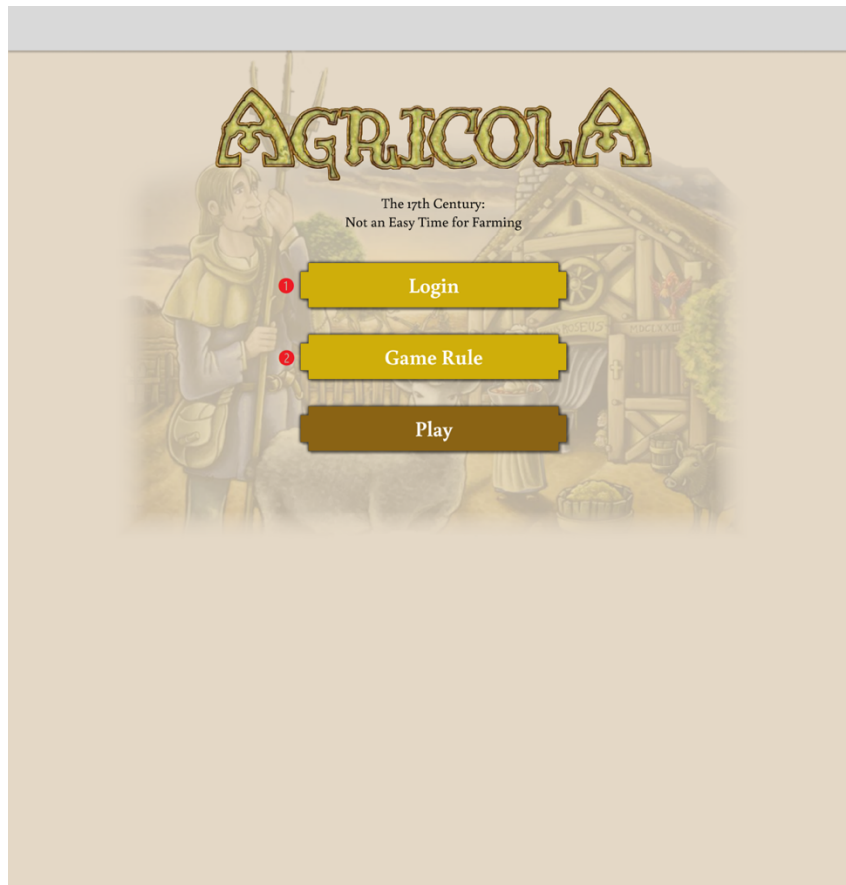
- 사용되는 기술은 이미 널리 성능이 입증된 것을 사용한다.
- 별도의 라이선스가 필요하거나 저작권료 지불이 필요한 기술 또는 소프트웨어의 사용을 피한다. (단, 유일한 기술 또는 소프트웨어로 본 시스템의 필수적으로 필요한 경우는 제외한다.)
- 시스템 전반의 성능 향상을 모색하는 방향으로 결정한다.
- 사용자에게 보다 친숙하고 편리한 방향으로 결정한다.
- 가급적 오픈 소스 소프트웨어를 사용한다.
- (GPL을 따르는 소프트웨어를 사용할 경우, 추후 상업화에 문제가 없는지 충분히 검토한다.)
- 추후의 상업적인 용도로 사용할 것을 고려하여, 시스템의 단가와 유지 보수비용을 고려한다.
- 시스템의 추후 확장성과 가용성을 고려한다.
- 소프트웨어 제작 시 소스코드의 최적화를 통하여 시스템 자원의 낭비를 예방한다.
- 소스코드 작성 시 추후의 유지보수를 고려하여 충분한 주석을 추가한다.

마. 가정 (Assumptions)

본 문서는 아그리콜라 보드게임의 기본 규칙에 기반한 시나리오의 필요한 기능을 위주로 작성된 요구 명세서다.

3. 상세 요구사항 (Detailed Requirements)

가. UI 요구사항 (UI Requirements)



[그림 1-1] 메인 페이지 UI



[그림 1-2] 로비 페이지 UI



[그림 1-3] 메인 페이지 UI

나. 요구사항 명세표 (Requirements Specification Table)

- 기능적 요구사항

REQ - #	구분	요구사항	비고
REQ-001	유저 등록 및 로그인	사용자는 아그리콜라 웹 게임 접속 시, '로그인' 버튼을 통해서 로그인할 수 있어야 한다.	[그림 1]의 ① 번 버튼 참조
REQ-002	유저 등록 및 로그인	사용자는 아그리콜라 웹 게임 접속 시, '게임 룰' 버튼을 클릭해 게임 룰을 확인할 수 있어야 한다.	[그림 1]의 ② 번 버튼 참조
REQ-003	유저 등록 및 로그인	사용자는 소셜로그인을 통해서 간편 로그인할 수 있어야 한다.	
REQ-004	로그인 및 로비 이동	사용자는 로그인 후 자동으로 온라인 아그리콜라 로비에 접속할 수 있어야 한다.	
REQ-005	로비화면 - 기본	사용자는 로비에서 '프로필 설정' 버튼을 통해서 닉네임 및 프로필 사진을 설정할 수 있어야 한다.	[그림 1]의 ④ 번 버튼 참조, 로비화면 진입을 위한 선행조건으로 시스템 로그인이 필요하다

REQ-006	로비화면 - 기본	사용자는 로비에서 '설정' 버튼을 통해서 초대 허용 및 거부 기능, 배경음 및 효과음 on/off 를 설정할 수 있어야 한다.	
REQ-007	로비화면 - 기본	사용자는 로비에서 '친구' 버튼을 통해서 온, 오프라인 친구 확인 및 친구 추가를 할 수 있어야 한다.	
REQ-008	로비 화면 - 게임 룸	사용자는 로비에서 다른 사용자가 생성한 게임 대기방의 목록을 조회할 수 있어야 하며, 클릭을 통해 참여할 수 있어야 한다.	[그림 2]의 5 번 창 참조
REQ-009	로비 화면 - 게임 룸	사용자는 로비에서 '방 생성' 버튼을 통해서 게임 대기방 생성 모달 창을 확인할 수 있어야 한다.	[그림 2]의 3 번 버튼 참조
REQ-010	로비 화면 - 게임 룸	게임 대기방 생성 모달 창에는 대기방 명, 대기방 비밀번호, 채팅 기능, 내 차례 제한시간에 대해 설정할 수 있어야 한다.	[그림 2]의 6 번 창 참조
REQ-011	로비 화면 - 게임 룸	게임 대기방의 모든 조건을 선택 완료한 후 '확인' 버튼을 통해서 게임 대기방을 생성할 수 있어야 한다.	
REQ-012	로비 화면 - 게임 룸	게임 대기 방의 사용자는 '초대하기' 버튼을 통해서 친구목록의 사용자에게 초대장을 보낼 수 있어야 한다.	
REQ-013	로비 화면 - 게임 룸	게임 대기 방의 사용자는 '나가기' 버튼을 통해서 게임 대기방에서 나갈 수 있으며, 자동으로 로비로 돌아갈 수 있어야 한다.	
REQ-014	로비 화면 - 게임 룸	게임 대기 방 생성자는 대기 방의 사용자를 클릭하여, 프로필 조회 및 강제퇴장을 할 수 있어야 한다.	
REQ-015	로비 화면 - 게임 룸	게임 대기 방 생성자는 인원이 모두 모였을 시 활성화되는 '게임시작' 버튼을 통해서 대기방의 모든 플레이어와 게임을 시작할 수 있어야 한다.	
REQ-016	게임 준비단계	게임이 시작되면 게임 보드판과 보드판 위에 존재하는 모든 요소가 화면에 나타나야 한다.	게임화면 진입을 위한 선행조건으로 로비 입장, 대기 방 생성자의 게임 시작이 필요하다
REQ-017	게임 준비단계	게임이 시작되면, 랜덤하게 선(先) 플레이어가 지정돼야 하며, 선(先) 플레이어를 기준으로 시계방향으로 차례가 진행돼야 한다.	
REQ-018	게임 준비단계	게임이 시작되면, 각 플레이어에게는 5*3 규격의 농장이 제공되며, 농장에는	[그림 3]의 9 번 창 참조

		2 명의 가족구성원과 방 2 칸짜리 나무집이 제공돼야 한다.	
REQ-019	게임 준비단계	게임이 시작되면, 각 플레이어에게는 7 장의 직업 카드와 7 장의 보조설비 카드가 제공된다.	
REQ-020	게임 준비단계	게임이 시작되면, 선플레이어에게는 음식 2 개 나머지 플레이어에게는 음식이 3 개씩 제공된다.	
REQ-021	게임 준비단계	게임은 총 6 주기로 진행되며, 각 주기는 4/3/2/2/2/1 라운드로 총 14 라운드로 구성되어 있어야 한다.	
REQ-022	게임 준비단계	각 주기의 라운드 카드는 랜덤한 순서로 배정돼야 한다.	
REQ-023	게임 준비단계	사용자는 우측 하단의 '보유 중인 카드' 버튼을 통해서 보유 중인 보조설비, 주요설비 및 직업카드 이미지를 확인할 수 있어야 한다.	[그림 3]의 10 번 버튼 참조
REQ-024	게임 준비단계	상단, 좌측, 우측에는 다른 플레이어의 재료현황, 가축현황, 직업, 보조설비, 주요설비 정보가 표시돼 있다.	
REQ-025	게임 준비단계	상단, 좌측, 우측 플레이어 정보를 클릭하면 해당 플레이어의 농장을 확인할 수 있어야 한다.	[그림 3]의 12 번 버튼 참조
REQ-026	게임 준비단계	사용자는 스크롤을 아래로 내리면, 본인의 농장을 확인할 수 있어야 한다.	
REQ-027	게임 준비단계	각 카드의 이미지에는 해당 카드를 얻기 위한 조건과 사용방법과 해당 카드의 효과가 표시돼 있다.	
REQ-028	게임 준비단계	사용자는 우측 하단의 채팅 기능을 통해 게임 내에서 다른 사용자와 대화할 수 있어야 한다.	[그림 3]의 8 번 창 참조
REQ-029	라운드 준비	매 라운드 시작 직전에, 해당 라운드의 라운드 카드가 오픈돼야 한다.	
REQ-030	라운드 준비	라운드 카드 오픈 후 행동칸과 라운드 카드에는 해당 자원만큼 추가돼야 하며, 화살표 표시가 있는 행동칸과 라운드 카드에는 해당 자원이 누적해서 추가돼야 한다.	
REQ-031	라운드 준비	각 행동 칸과 오픈 된 라운드 카드는 버튼의 형태로 제공되며, 본인의 차례에 클릭을 통해서 해당 행동 및 라운드 카드를 이용할 수 있다. 또, 라운드 시작 전에 모두 활성화된다.	

REQ-032	라운드 플레이	보드판 우측 하단의 주요설비 버튼을 클릭하면, 현재 게임에 남아있는 주요설비 이미지를 확인할 수 있다.	[그림 3]의 7 번 버튼 참조
REQ-033	라운드 플레이	주요설비 이미지에는 해당 주요설비를 얻기 위한 조건과 사용방법, 점수 등이 표시돼 있다.	
REQ-034	라운드 플레이	‘보유 중인 카드’ 버튼을 누르면 활성화된 카드와 비활성화된 카드가 분리되어 있어야 한다.	[그림 3]의 10 번 버튼 참조
REQ-035	라운드 플레이	활성화된 카드 중 ‘아무 때나’ 조건이 달린 카드는 해당 카드를 클릭해 차례와 관계없이 해당 행동을 할 수 있어야 한다.	
REQ-036	라운드 플레이	플레이어는 해당 라운드에서 보유한 가족구성원의 수만큼 차례가 주어진다.	
REQ-037	라운드 플레이	기본적으로 가족구성원은 농장의 방 1 칸 당 1 명이 존재할 수 있다. 예외적으로 ‘급한 가족 놀리기’ 행동 칸을 이용할 시 방 1 칸에 2 명의 가족 구성원이 존재할 수 있다.	
REQ-038	라운드 플레이	행동 칸과 오픈 된 라운드 카드는 플레이어 중 한 명이라도 이용하면 즉시 해당 라운드에서 비활성화 된다.	
REQ-039	라운드 플레이	행동 칸과 오픈 된 라운드 카드는 우측 상단 도움말 버튼을 통해서 세부 정보를 확인할 수 있다.	
REQ-040	라운드 플레이	플레이어는 본인의 가족구성원을 행동카드 혹은 오픈 된 라운드 카드 중 한 곳을 클릭하여 배치시켜야 한다.	
REQ-041	라운드 플레이	가족구성원이 배치된 행동 칸 혹은 오픈 된 라운드카드로 이동하며, 해당 내용을 수행한 후 본인의 차례가 종료된다.	
REQ-042	라운드 플레이	모든 플레이어의 차례가 모두 종료되면, 해당 라운드가 종료된다.	
REQ-043	라운드 플레이	라운드가 종료되면, 모든 가족 구성원이 해당 플레이어 농장의 방으로 돌아가며,	
REQ-044	라운드 플레이	‘점수 계산표’ 버튼을 통해서 점수 계산표를 확인할 수 있다.	[그림 3]의 11 번 버튼 참조
REQ-045	수확단계	각 주기의 마지막 라운드가 끝날 때는 추가적으로 수확 단계가 진행된다.	
REQ-046	수확단계	수확 단계는 농장단계 → 가족 먹여 살리기 단계 → 번식 단계 순으로 진행된다.	

REQ-047	수확단계	수확 단계의 '농장 단계'에서는 농지의 가장 위에 있는 곡식 혹은 채소 1 개가 개인창고로 이동된다.	
REQ-048	수확단계	수확 단계의 '가족 먹여 살리기 단계'에서는 가족 구성원 1 명당 음식 2 개를 지불해야 한다. '신생아'는 음식 1 개를 지불해야 한다.	
REQ-049	수확단계	수확 단계의 '번식 단계'에서는 같은 종류의 가축이 2 마리 이상인 경우 해당 가축 1 마리를 추가로 얻는다.	
REQ-050	게임 종료	게임 종료 시, 점수 계산표에 따라 점수가 계산되며, 계산 과정과 결과를 게임 종료 창을 통해 모든 플레이어에게 보여준다.	
REQ-051	게임 종료	게임 종료 시, 계산된 점수에 따라 순위가 부여되며, 모든 플레이어에게 순위판을 보여준다.	
REQ-052	게임 종료	순위 판에는 각 부분 점수와 총 점수 및 순위가 표시된다.	
REQ-053	게임 종료	게임 종료 시, 플레이어의 게임 정보가 저장되며 플레이어의 평균 순위 및 판수가 저장된다.	
REQ-054	게임 종료	게임 종료 시, 순위판의 플레이어 옆의 친구추가 버튼을 통해서 플레이어를 친구로 추가할 수 있다.	
REQ-055	게임 종료	게임 종료 시, 순위판 하단의 나가기 버튼을 통해서 로비로 돌아갈 수 있다.	
REQ-056	게임 세부 규칙	농지를 개간할 때, 본인의 농장이 올라오며 최초 개간 시에는 빈 칸 중 아무 곳을 클릭하여 농장을 개간할 수 있다. 이 후의 농장 개간 시에는 개간된 농지의 상, 하, 좌, 우 중 1 칸에만 개간할 수 있다.	
REQ-057	게임 세부 규칙	'씨 뿌리기' 행동 시 비어 있는 농지에만 씨를 뿌릴 수 있다.	
REQ-058	게임 세부 규칙	방의 개수와 상관없이 집에는 가축을 1 마리만 키울 수 있다.	
REQ-059	게임 세부 규칙	울타리는 농장 칸의 상, 하, 좌, 우 4 군데에 설치 가능하며, 농장 칸이 열린 곳 없이 모두 설치되어야 가축을 키울 수 있다.	
REQ-060	게임 세부 규칙	울타리로 둘러싸인 농장 1 칸당 같은 종류의 가축을 2 마리 키울 수 있다.	

REQ-061	게임 세부 규칙	외양간은 농장 1 칸에 1 개씩 설치 가능하며, 외양간에는 가축 1 마리를 키울 수 있다.	
REQ-062	게임 세부 규칙	울타리로 둘러싸인 농장속의 외양간 1 개당 울타리 내에서 키울 수 있는 가축의 수가 2 배가 된다.	
REQ-063	게임 세부 규칙	방을 추가할 때는 기존 방의 상, 하, 좌, 우 중 한 군데에 추가 가능하다.	
REQ-064	게임 세부 규칙	집은 나무집 → 흙집 → 돌집 순으로 업그레이드할 수 있다.	
REQ-065	게임 세부 규칙	집 업그레이드는 부분적으로 할 수 없으며, 업그레이드 비용은 방 수의 배수로 지불해야 한다.	
REQ-066	게임 세부 규칙	가축을 얻을 때, 사용자의 농장에 가축을 둘 곳이 없으면 해당 가축은 사라진다.	
REQ-067	게임 세부 규칙	‘가족 구성원 늘리기’ 혹은 ‘급한 가족 구성원 늘리기’ 행동 칸을 통해 늘어난 가족 구성원은 ‘신생아’로 취급된다.	
REQ-068	게임 세부 규칙	‘신생아’는 가족 구성원으로 분류되지 않아 차례를 가지지 않으며, 다음 라운드부터 가족구성원으로 분류된다.	
REQ-069	게임 세부 규칙	‘가족 구성원 늘리기’ 행동 칸을 통해 가족 구성원을 늘릴 때, 라운드 종료까지 ‘신생아’가 들어갈 방이 없다면 ‘신생아’는 사라진다.	
REQ-070	게임 세부 규칙	점수 계산 시, 구걸 카드는 1 장당 - 3 점으로 계산된다.	

[표 1-1] 기능적 요구사항

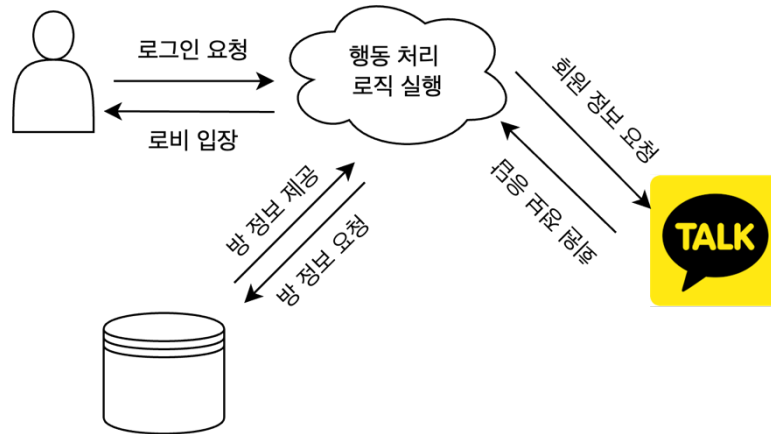
- 비기능적 요구사항

요구사항 #	구분	요구사항
REQ-071	보안	사용자의 회원 정보 및 게임 데이터는 DB 에 저장돼야 한다.
REQ-072	보안	사용자는 DB 에 임의 접근이 불가능해야 한다.
REQ-073	성능	게임 진행 시, 데이터는 500ms 이내의 속도로 처리돼야 한다.
REQ-074	사용성	사용자가 게임에서 이탈한 뒤, 재접속해도 진행 중인 게임에 다시 참여할 수 있어야 한다.
REQ-075	사용성	사용자는 다양한 디바이스와 브라우저에서 작동해야 한다.
REQ-076	사용성	사용자는 게임을 확대 및 축소할 수 있어야 한다.
REQ-077	인터페이스 반응성	위쪽과 좌, 우의 사이드 바를 클릭하면 즉시 다른 플레이어의 농장이 펼쳐져야 한다.
REQ-078	인터페이스 반응성	하단 스크롤을 통해서 즉시 본인의 농장을 확인할 수 있어야 한다.

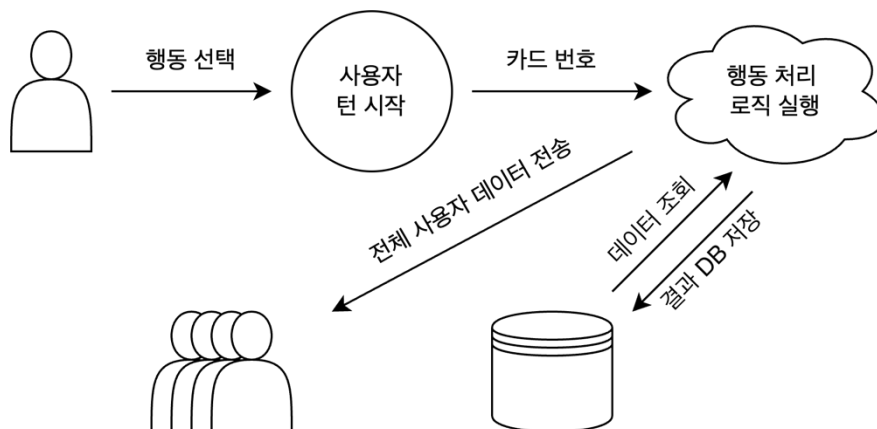
[표 1-2] 비기능적 요구사항

다. 데이터 요구사항 (Data Requirements)

- 정보 흐름도



[그림 2-1] 로그인 & 로비 플로우



[그림 2-2] 게임 플레이 플로우



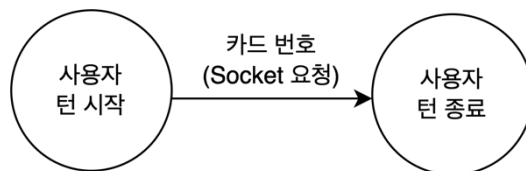
[그림 2-3] 사용자 정보 수정



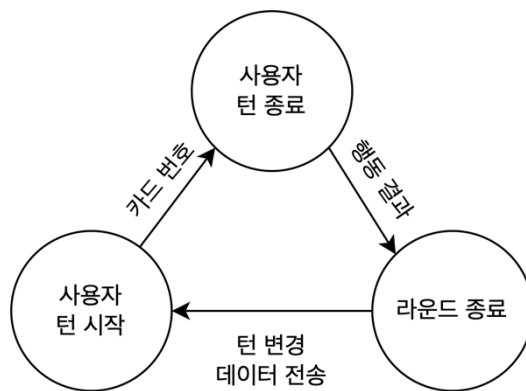
[그림 2-4] 로비 & 방 생성



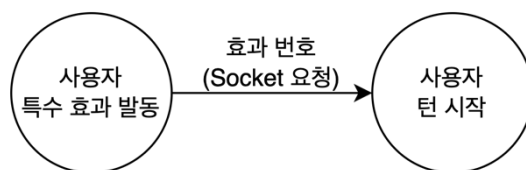
[그림 2-5] 게임 플레이



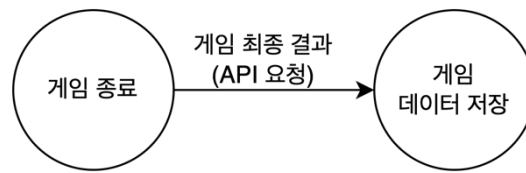
[그림 2-6] 사용자 행동 처리



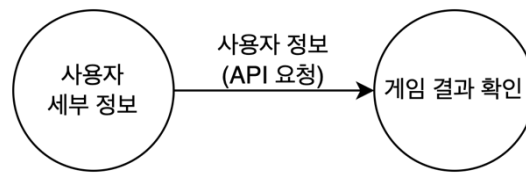
[그림 2-7] 사용자 턴 진행



[그림 2-8] 사용자 카드 사용

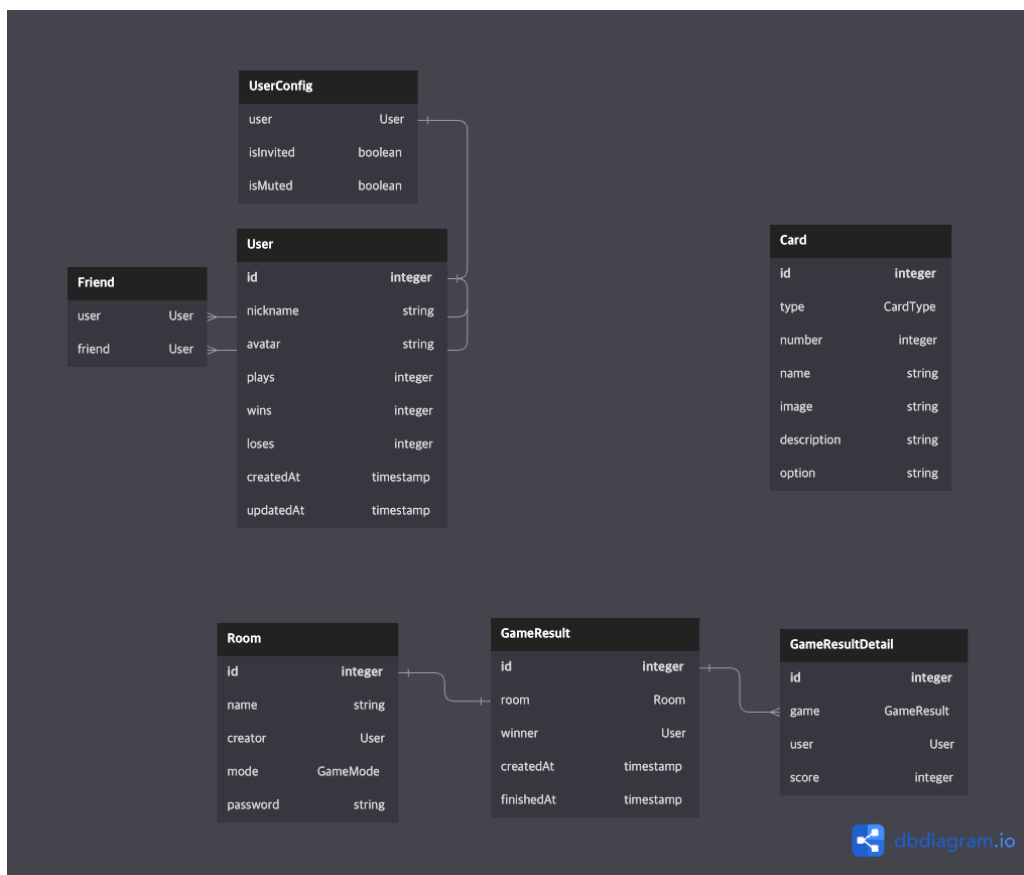


[그림 2-9] 게임 결과 처리 1



[그림 2-10] 게임 결과 처리 2

- ERD(Entity Relationship Diagram)



[그림 3-1] PostgreSQL ER Diagram

라. 성능 요구사항 (Performance Requirements)

. 본 시스템은 동시 접속자수 100명 이상의 클라이언트에 안정적인 QoS를 제공하여야 한다.

또한 25개 이상의 게임 데이터를 유지 관리 할 수 있어야 하며, 등록된 게임 데이터에 대한 메타 데이터를 서비스 할 수 있어야 한다. 또한 각 클라이언트의 하드웨어 특성에 맞는 멀티미디어 데이터를 저장할 수 있어야 한다.

사용자의 모든 요청과 명령에 2초 이내에 응답을 해야 하며, 행동 선택 후에 5초 이내에 행동에 대한 결과가 처리되어야 하고, 사용자의 행동 외 명령은 3초 이내에 요청한 동작을 수행해야 한다. 메타 데이터의 요청 시 3초 이내에 응답 결과를 처리하고 해당 정보가 클라이언트에 제공되어야 한다. 사용자의 명령이 짧은 시간에 다양하게 요청될 경우, 최종의 명령을 수행 하여야 한다.

클라이언트 내부에서의 각 사용자 명령에 대해 1초 이내에 응답하여야 한다.

마. 논리적 데이터베이스 요구사항 (Logical Database Requirements)

. 본 시스템의 데이터베이스는 질의 요청에 대해 1초 이내의 응답시간을 가져야 하며, 200장 이상의 모든 카드 정보와 함께 모든 메타 데이터 정보를 저장하고 관리할 성능을 가져야 한다. 중복 질의를 고속으로 처리해야 하며, 일반적인 데이터베이스 시스템의 제약 조건과 성능을 만족해야 한다.

바. 코딩 컨벤션 (Coding Conventions)

- 들여쓰기는 공백 4 개를 사용합니다.

```
def example_function():
    if True:
        print("Indented with 4 spaces")
```

- 한 줄의 최대 길이는 79자로 제한합니다.

```
# Bad
a_very_long_variable_name_that_is_hard_to_read =
some_function_call_with_a_long_argument_that_is_also_hard_to_read()

# Good
a_very_long_variable_name_that_is_hard_to_read =
some_function_call_with_a_long_argument_that_is_also_hard_to_read()
```

- 함수와 클래스 정의는 빈 줄 두 개로 구분합니다.

```
class ExampleClass:

    def __init__(self):
        pass

    def example_method(self):
        pass

def example_function():
    pass
```

- 모듈 레벨 상수는 대문자와 밑줄을 사용하여 정의합니다.

```
EXAMPLE_CONSTANT = 42
```

- 함수와 변수의 이름은 소문자와 밑줄을 사용하여 정의합니다.

```
example_variable = 123

def example_function():
    pass
```

- 클래스의 이름은 각 단어의 첫 글자를 대문자로 합니다.

```
class ExampleClass:
    pass
```

- 비공개 인스턴스 변수의 이름은 밑줄로 시작합니다.

```
class ExampleClass:

    def __init__(self):
        self._private_variable = 123
```

- 상수는 모듈 레벨에서 정의되며, 클래스나 함수 내에서 정의하지 않습니다.

```
EXAMPLE_CONSTANT = 42

class ExampleClass:
    pass

def example_function():
    pass
```

- 함수의 인자와 반환값은 괄호 안에 타입을 명시하여 지정할 수 있습니다.

```
def example_function(argument1: str, argument2: int) -> float:
    return 3.14
```

- 리스트, 딕셔너리, 튜플 등의 컨테이너 타입은 []를 사용합니다.

```
example_list = [1, 2, 3]
example_dict = {"key1": "value1", "key2": "value2"}
example_tuple = (1, 2, 3)
```

- 긴 식이나 긴 문자열은 괄호 안에서 줄을 바꾸어 작성합니다.

```
example_variable = ("This is a very long string that cannot fit in one
line."
                    "So we split it into multiple lines using parentheses.")
```

- 리스트 인덱싱이나 슬라이싱에서는 [:]보다 [;:] 형태로 작성합니다.

```
example_list = [[1, 2], [3, 4], [5, 6]]
first_column = [row[0] for row in example_list] # Good
first_column = [row[:1] for row in example_list] # Bad
```

- 한 줄 주석은 # 기호 뒤에 한 칸을 띄고 작성합니다.

```
example_variable = 123 # This is an example comment
```

- 변수 할당 앞뒤로는 공백을 두고 작성합니다.

```
example_variable = 123
another_variable = "Hello, world!"
```

- 항상 클래스와 함수에 대한 문서화를 작성합니다.

```
class ExampleClass:
    """This is an example class that demonstrates how to write
    documentation."""

    def example_method(self, argument1, argument2):
        """This is an example method that demonstrates how to write
        documentation."""
        pass

    def example_function(argument1, argument2):
        """This is an example function that demonstrates how to write
        documentation."""
        pass
```

사. 소프트웨어 시스템 특성 (Software System Characteristics)

- 신뢰도 (Reliability)
: 사용자의 요청에 정확한 정보와 멀티미디어 데이터를 제공
- 가용성 (Availability)
: 사용자의 요청에 언제든지 허용 응답시간 내에 요청
- 보안 (Security)
: 사용자의 개인 정보의 암호화 및 계정 관리

- 유지가능성 (Maintainability)
: 관리자의 서버 관리 및 유지 보수 용이성 극대화
- 호환성 (Portability)
: 오픈 소스와 리눅스를 이용하여 타 시스템으로의 호환성과 이식성 향상 제고

4. 추가정보 (Additional Information)

가. 소프트웨어 요구사항 명세서 (SRS)

본 소프트웨어 요구사항 명세서는 IEEE의 권고안 (IEEE Recommended Practice for Software Requirements Specifications)을 따라 작성되었음.

나. 문서 이력 (Document History)

날짜	버전	설명	작성자
2023.00.00	0.1	스타일 작업 및 개요	.
2023.00.00	0.1.1	초안	.
2023.00.00	0.2	세부 명세	.
2023.00.00	0.3	데이터 추가	.
2023.00.00	0.5	내용 검토 및 추가	.
2023.00.00	0.6	서식 변경 및 내용 추가	.
2023.00.00	0.7	데이터 구조 추가	.

[표 2] 소프트웨어 요구사항 명세서 문서이력