



# Beddit

Analysis Document

21812062 김주환

## Revision History

Revision date	Version #	Description	Author
2023.04.07	1.0	First Draft	김주환
2023.04.17	1.1	Use case diagram	김주환
2023.04.20	1.2	Domain analysis User Interface Prototype	김주환
2023.04.22	1.3	Glossary References	김주환
2023.05.02	1.4	Non-Functional Requirement	김주환

# Contents

1. Introduction.....	4
2. Use case analysis.....	5
3. Domain analysis.....	25
4. User Interface prototype .....	27
5. Glossary.....	36
6. References .....	37

## 1. Introduction

User가 특정 커뮤니티를 이용하는 이유는 단순히 정보수집의 목적도 있지만, ‘자아 표현의 목적’도 가지고 있다. 사진이나 글을 업로드하여 다른 User로부터 피드백을 받게 되는데, 이는 인정에 대한 욕구를 채워준다, 그래서 User는 만족감을 얻고 커뮤니티를 이용하게 됩니다.

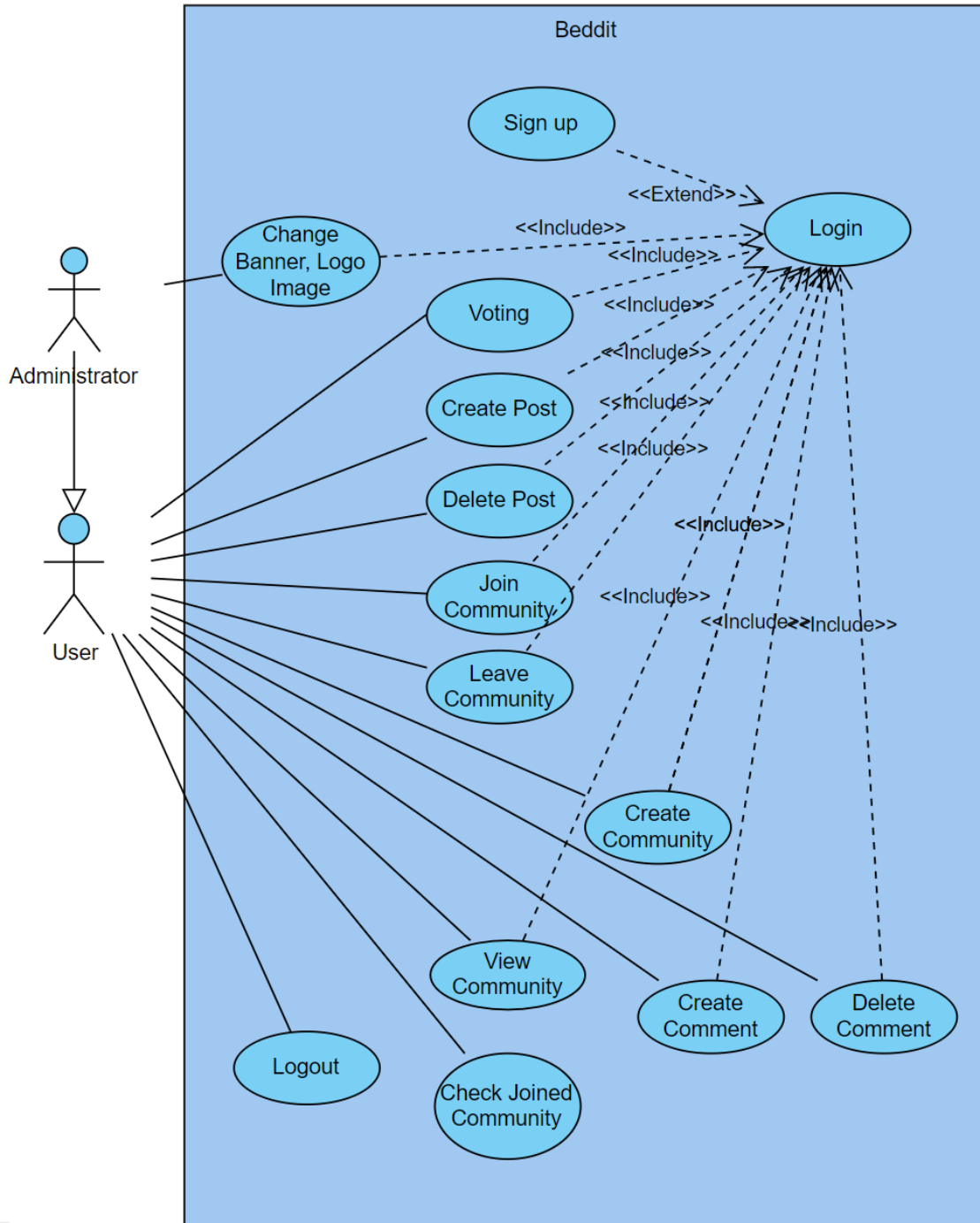
또한 User는 커뮤니티 그룹이 세분화되어 이전과 다르게 자신이 관심있는 공통 주제에 대한 소통을 세분화되어 있는 커뮤니티에서 같은 집단에 속해져 있는 듯한 느낌을 받으며 유대 관계 형성에 도움을 받기도 한다. 기업이 서비스와 함께 커뮤니티를 제공해야 하는 이유는 고객의 체류 시간, 특정 목적이 없어도 해당 앱을 습관적으로 접속한다. SNS를 하듯, 특별한 이유 없이 해당 앱을 자주 방문하고 오래 머물 수 있다.

중고거래 앱 당근마켓은 중고거래 플랫폼에서 지역 공동체 플랫폼으로 발전하고 있다. 지역을 기반으로 중고거래를 할 수 있게 한 이유는 해당 중고거래를 하면서 모인 이용자들을 커뮤니티로 이끌기 위해 서였다. 당근마켓의 동네생활 서비스는 2020년 9월에 오픈한 서비스로, 이웃끼리 유용한 동네정보, 소식 또는 소소한 일상을 나누면서 자유롭게 소통하는 곳으로 즉 하나의 지역기반 커뮤니티로 발전했다.

비단 당근마켓 뿐만 아니라 우리 주변에는 다양한 커뮤니티가 존재한다. 대표적으로 주로 대학생이 이용하는 에브리타임, 직장인들이 이용하는 블라인드, 기혼 여성이 주로 이용하는 맘 카페 등이 대표적이다. 이 들을 이용한 커뮤니티 마케팅을 이용하여 잠재 고객 혹은 타겟 고객에 대한 공략이 가능해진다.

## 2. Use case analysis

### 1) Use case diagram



[그림 2-1] use case diagram

## 2) Use case description

Use Case #1: Login	
GENERAL CHARACTERISTICS	
Summary	User가 아이디, 비밀번호를 통해 로그인 성공 및 실패 여부를 확인하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 자신의 아이디와 비밀번호를 통해 회원가입을 한 상태여야 한다.
Trigger	'Beddit' 웹사이트에 접속을 하고 커뮤니티에 대한 기능을 이용하고자 할 때 (글쓰기, 댓글 달기, 투표하기)
Success Post Condition	User가 입력한 로그인과 비밀번호가 DB에 저장된 내용과 일치하면 로그인에 성공한다.
Failed Post Condition	User가 입력한 로그인과 비밀번호가 DB에 저장된 내용과 불일치하면 로그인에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인할 때 시작된다.
1	User는 웹사이트 메인 화면에서 로그인 버튼을 클릭한다.
2	User는 아이디와 비밀번호를 입력하여 로그인을 시도한다.
3	User가 입력한 아이디와 비밀번호를 'Beddit'에 등록된 회원인지 확인한다 가입된 User라면 로그인에 성공하고 이전페이지로 되돌아온다.
4	이 Use case는 로그인이 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
3	3a. User가 입력한 아이디와 비밀번호가 일치하지 않는 경우 3a.1. 아이디 또는 비밀번호가 틀렸다는 메시지를 출력한다. 3b. 아이디 또는 비밀번호가 입력되지 않은 경우 3b.1. 아이디 또는 비밀번호가 입력되지 않았다는 메시지를 출력한다.
RELATED INFORMATION	

Performance	$\leq 1$ Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #2: Logout	
GENERAL CHARACTERISTICS	
Summary	User가 로그아웃 하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어있는 상태여야 한다.
Trigger	User가 로그아웃버튼을 클릭할 때
Success Post Condition	User는 로그아웃에 성공한다.
Failed Post Condition	User는 로그아웃에 실패한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인이 된 상태에서 로그아웃 버튼을 눌렀을 때 시작된다.
1	User는 웹사이트 메인 화면에서 로그아웃 버튼을 클릭한다.
2	시스템은 해당 User의 로그인 상태를 조정한다.
4	이 Use case는 로그아웃이 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
3	3a. User가 입력한 아이디와 비밀번호가 일치하지 않는 경우 3a.1. 아이디 또는 비밀번호가 틀렸다는 메시지를 출력한다. 3b. 아이디 또는 비밀번호가 입력되지 않은 경우 3b.1. 아이디 또는 비밀번호가 입력되지 않았다는 메시지를 출력한다.
RELATED INFORMATION	
Performance	≤ 1 Seconds
Frequency	Variable
Concurrency	None
Due Date	



Use Case #3: Sign up	
GENERAL CHARACTERISTICS	
Summary	User가 회원가입 하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어있지 않은 상태여야 한다.
Trigger	User가 로그인 버튼을 클릭할 때
Success Post Condition	User가 회원가입에 성공한다. 이후에는 해당 계정 정보로 로그인 가능해진다.
Failed Post Condition	User는 회원가입에 실패한다. 사유와 함께 로그아웃 상태를 유지한다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그아웃 된 상태에서 회원가입 버튼을 클릭한다.
1	User는 웹사이트 메인 화면에서 회원가입 버튼을 클릭한다.
2	시스템은 회원가입 화면을 출력한다.
3	User는 아이디와 비밀번호를 입력하고 회원가입을 하거나 또는 자신의 구글 아이디를 이용하여 회원가입을 시도한다.
4	시스템은 회원가입 성공 여부를 판단한다. 성공했다면 로그인 이 된 상태로 이전페이지로 돌아간다.
5	이 Use case는 회원가입이 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
4	4a. User가 입력한 아이디가 존재하는 경우 4a.1. User가 입력한 이메일의 User가 이미 가입되어 있다는 메시지를 출력한다. 4b. User가 입력한 비밀번호와 비밀번호 확인이 다른 경우 4b.1. 비밀번호가 다르다는 메시지를 출력한다. 4c. User가 구글을 이용해 회원가입을 시도하는 경우 4b.1. 구글 로그인 화면을 출력한다.
RELATED INFORMATION	
Performance	≤ 1 Seconds
Frequency	User당 1번

Concurrency	None
Due Date	

Use Case #4: Create Post	
GENERAL CHARACTERISTICS	
Summary	User가 커뮤니티에 게시글을 업로드 하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어 있는 상태여야 한다.
Trigger	User가 Create Post 버튼을 클릭할 때
Success Post Condition	User가 작성한 게시글이 업로드에 성공한다. 해당 게시글 정보가 DB서버에 저장된다.
Failed Post Condition	User는 작성한 게시글이 업로드에 실패한다. 해당 게시글 정보가 DB 서버에 저장 되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태에서 특정 커뮤니티 채널에 접속하여 Create Post 버튼을 클릭한다.
1	User는 웹사이트 메인 화면에서 자신이 업로드 하고자 하는 커뮤니티에 접속한다.
2	User는 Create Post 버튼을 누른다.
3	User는 자신이 작성하고자 하는 게시글의 제목과 내용을 작성하고 Post 버튼을 누른다.
4	시스템은 User가 작성한 게시글의 정보를 확인하고 DB 서버에 해당 정보를 저장한다.
5	이 Use case는 User가 작성한 게시글의 업로드가 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
4	3a. User가 작성한 게시글 제목과 내용이 없는 경우 3a.1. Post 버튼을 비 활성화한다. 3b. User가 이미지를 업로드 하려고 하는 경우 3b.1. 파일을 선택할 수 있는 창이 출력된다. 3c. User가 취소 버튼을 누른 경우 3c.1. Create Post 버튼을 누르기 이전화면으로 되돌아 간다.

RELATED INFORMATION	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #5: Delete Post	
GENERAL CHARACTERISTICS	
Summary	User가 게시글을 삭제하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어 있는 상태여야 한다. 삭제하고자 하는 게시글은 유저 본인이 업로드 한 게시글이어야 한다.
Trigger	User가 Delete Post 버튼을 클릭할 때
Success Post Condition	User가 작성한 게시글이 삭제에 성공한다. 해당 게시글 정보가 DB서버에서 삭제된다.
Failed Post Condition	User는 작성한 게시글이 삭제에 실패한다. 해당 게시글 정보가 DB 서버에 그대로 남게 된다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태에서 특정 커뮤니티에 접속하여 자신이 업로드한 게시글의 Delete Post 버튼을 클릭한다.
1	User는 자신이 등록한 게시글을 삭제하고자 하는 커뮤니티에 접속한다.
2	User는 Delete Post 버튼을 누른다.
3	시스템은 User가 작성한 게시글의 정보를 확인하고 DB 서버에 해당 정보를 삭제한다.
4	이 Use case는 User가 작성한 게시글의 삭제가 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
4	2a. User가 작성하지 않은 게시글일 경우 2a.1. Delete Post 버튼을 비활성화한다. 3a. User가 로그인 하지 않은 상태인 경우 3a.1. Delete Post 버튼을 비활성화 한다.
RELATED INFORMATION	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #6: Create Community	
GENERAL CHARACTERISTICS	
Summary	User가 원하는 커뮤니티를 개설하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어 있는 상태여야 한다.
Trigger	User가 Create Community 버튼을 클릭할 때
Success Post Condition	User 본인이 원하는 커뮤니티 개설에 성공한다. 유저가 개설한 커뮤니티의 정보가 DB서버에 저장된다.
Failed Post Condition	User 본인이 원하는 커뮤니티 개설에 실패한다. 유저가 입력한 커뮤니티 정보가 DB 서버에 저장되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태에서 본인이 원하는 커뮤니티를 개설한다.
1	User는 웹사이트 메인 화면에서 Create Community 버튼을 클릭한다.
2	User는 본인이 만들고자 하는 커뮤니티의 이름을 작성한다.
3	시스템은 User가 입력한 커뮤니티의 정보를 확인하고 DB 서버에 해당 정보를 저장한다.
4	이 Use case는 User가 자신의 커뮤니티 개설에 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
4	3a. User가 입력한 커뮤니티 이름이 존재하는 경우 3a.1. 해당 커뮤니티의 존재 메시지를 출력한다.
RELATED INFORMATION	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #7: Join Community	
GENERAL CHARACTERISTICS	
Summary	User가 원하는 커뮤니티를 가입하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어 있는 상태여야 한다.
Trigger	User가 Join Community 버튼을 클릭할 때
Success Post Condition	User가 원하는 커뮤니티 가입에 성공한다. 유저가 가입한 커뮤니티의 정보가 해당 유저의 DB서버에 저장된다.
Failed Post Condition	User가 원하는 커뮤니티 가입에 실패한다. 유저가 가입 시도한 커뮤니티 정보가 DB 서버에 저장되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태에서 본인이 가입하기를 원하는 커뮤니티를 접속한다.
1	User는 커뮤니티 메인 화면에서 Join Community 버튼을 클릭한다.
2	시스템은 User가 가입하기를 원하는 커뮤니티의 정보를 확인하고 DB 서버에 해당 정보를 저장한다.
3	이 Use case는 User가 커뮤니티 가입에 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
2	3a. User가 이미 가입한 커뮤니티일 경우 3a.1. 해당 Join Community 버튼이 Leave Community 버튼으로 변환된다.
RELATED INFORMATION	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #8: Leave Community	
GENERAL CHARACTERISTICS	
Summary	User가 가입한 커뮤니티를 탈퇴하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어 있는 상태여야 한다.
Trigger	User가 Leave Community 버튼을 클릭할 때
Success Post Condition	User가 원하는 커뮤니티 탈퇴에 성공한다. 유저가 탈퇴한 커뮤니티의 정보가 해당 유저의 DB서버에서 삭제된다.
Failed Post Condition	User가 원하는 커뮤니티 탈퇴에 실패한다. 유저가 탈퇴 시도한 커뮤니티 정보가 유저 DB 서버에 저장되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태에서 본인이 탈퇴하기를 원하는 커뮤니티를 접속한다.
1	User는 커뮤니티 메인 화면에서 Leave Community 버튼을 클릭한다.
2	시스템은 User가 탈퇴하기를 원하는 커뮤니티의 정보를 확인하고 유저 DB 서버에 해당 정보를 삭제한다.
3	이 Use case는 User가 커뮤니티 탈퇴에 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
2	2a. User가 가입하지 않은 커뮤니티일 경우 2a.1. 해당 Leave Community 버튼이 Join Community 버튼으로 전환된다.
RELATED INFORMATION	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	



Use Case #9: Create Comment	
GENERAL CHARACTERISTICS	
Summary	User가 게시글의 댓글을 작성하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어 있는 상태여야 한다.
Trigger	User가 Create Comment 버튼을 클릭할 때
Success Post Condition	User가 게시글 댓글 작성에 성공한다. 유저가 작성한 댓글의 내용이 해당 커뮤니티 DB 서버에 저장된다.
Failed Post Condition	User가 게시글 댓글 작성에 실패한다. 유저가 작성한 댓글의 내용이 해당 커뮤니티 DB 서버에 저장되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태에서 댓글을 작성하고자 하는 게시글의 상세 페이지에 접속한다.
1	User는 댓글의 내용을 작성하고 Create Comment 버튼을 누른다.
2	시스템은 User가 작성한 댓글의 내용을 해당 게시글 DB서버에 저장한다.
3	이 Use case는 User가 댓글 작성에 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
2	2a. User가 댓글 내용을 작성 안 한 경우 2a.1. 해당 Create Comment 버튼이 비 활성화 된다.
RELATED INFORMATION	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #10: Delete Comment	
GENERAL CHARACTERISTICS	
Summary	User가 작성한 게시글의 댓글을 삭제하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어 있는 상태여야 한다.
Trigger	User가 Delete Comment 버튼을 클릭할 때
Success Post Condition	User가 게시글 댓글 삭제에 성공한다. 유저가 작성한 댓글의 정보가 해당 커뮤니티 DB 서버에서 삭제된다.
Failed Post Condition	User가 게시글 댓글 삭제에 실패한다. 유저가 작성한 댓글의 내용이 해당 커뮤니티 DB 서버에서 삭제되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태에서 댓글을 삭제하고자 하는 게시글의 상세 페이지에 접속한다.
1	User는 본인이 작성한 댓글의 Delete Comment 버튼을 누른다.
2	시스템은 User가 작성한 댓글의 내용을 해당 게시글 DB서버에 저장한다.
3	이 Use case는 User가 작성한 댓글을 삭제하는데 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
2	1a. User가 작성하지 않은 댓글의 경우 1a.1. 해당 Delete Comment 버튼이 비 활성화된다.
RELATED INFORMATION	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #11: Voting	
GENERAL CHARACTERISTICS	
Summary	User가 다른 이들이 올린 게시글의 좋아요 및 싫어요 와 같은 투표하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어 있는 상태여야 한다.
Trigger	User가 Voting(Like, Dislike)버튼을 클릭할 때
Success Post Condition	User가 게시글 투표에 성공한다. 유저가 투표한 게시글의 정보가 해당 커뮤니티 DB 서버에 저장된다.
Failed Post Condition	User가 게시글 투표에 실패한다. 유저가 투표한 게시글의 정보가 해당 커뮤니티 DB 서버에 저장되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태에서 게시글 목록 또는 투표하고자 하는 게시글의 상세페이지에 접속한다
1	User는 본인이 투표를 원하는 게시글의 Voting 버튼을 누른다.
2	시스템은 User가 작성한 댓글의 내용을 해당 게시글 DB서버에 저장한다.
3	이 Use case는 User가 작성한 댓글을 삭제하는데 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
2	2a. User가 이미 투표한 게시글의 경우 2a.1. 해당 Voting 버튼이 활성화가 되어있다. 2b. User가 해당 게시글에 투표를 이미 한 상태이고 활성화된 Voting 버튼을 한번 더 누른 경우 2b.1. 유저가 이전에 선택한 투표(좋아요 / 싫어요)가 취소된다. 2c. User가 해당 게시글에 투표를 이미 한 상태이고 활성화 안된 Voting 버튼을 한 번 더 누른 경우 2b.1. 유저가 이전에 선택한 투표(좋아요 / 싫어요)가 취소되

	고 이후에 선택한 투표가 활성화된다.
<b>RELATED INFORMATION</b>	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #12: Check Joined Community	
GENERAL CHARACTERISTICS	
Summary	User가 자신이 현재 가입한 커뮤니티를 확인하고 싶을 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	User가 'Beddit'에 로그인 되어 있는 상태여야 한다. User가 적어도 1개의 커뮤니티에 가입되어 있어야 한다.
Trigger	User가 왼쪽 상단에 Home 메뉴 버튼을 클릭할 때
Success Post Condition	User가 가입한 커뮤니티의 리스트가 나타난다.
Failed Post Condition	User가 가입한 커뮤니티의 리스트가 나타나지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태와 함께 'Beddit'에 접속해 있어야 한다.
1	User는 왼쪽 상단의 Home 메뉴 리스트를 클릭한다.
2	시스템은 User가 가입한 커뮤니티 리스트를 DB 서버로부터 받아와서 유저에게 보여준다.
3	이 Use case는 User가 자신이 가입한 커뮤니티를 확인하는데 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
2	2a. User가 자신이 가입한 커뮤니티 리스트 중 하나를 클릭한 경우 2a.1. 해당 커뮤니티의 상세 페이지로 이동한다.
RELATED INFORMATION	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #13: Change Banner, Logo Image	
GENERAL CHARACTERISTICS	
Summary	Administrator 가 자신이 개설한 커뮤니티의 로고와 배너를 변경하고자 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	Administrator
Preconditions	관리자가 'Beddit'에 로그인 되어 있는 상태여야 한다. 관리자가 생성한 커뮤니티여야 한다.
Trigger	관리자가 Change(Logo or Banner)버튼을 클릭할 때
Success Post Condition	관리자는 커뮤니티 로고 및 배너 변경에 성공한다. 관리자가 변경한 로고와 배너가 해당 커뮤니티 DB에 저장된다.
Failed Post Condition	관리자는 커뮤니티 로고 및 배너 변경에 실패한다. 관리자가 변경한 로고와 배너가 해당 커뮤니티 DB에 저장되지 않는다.
MAIN SUCCESS SCENARIO	
Step	Action
S	관리자는 'Beddit'에 로그인 된 상태에서 자신이 개설한 커뮤니티에 접속한다.
1	관리자는 오른쪽 커뮤니티 관리 메뉴에서 Change 버튼을 클릭한다.
2	파일 선택 창이 뜨고 Administrator는 해당 창에서 파일을 선택 후 열기 한다.
3	해당 이미지 변경을 save 버튼을 눌러 완료한다.
4	시스템은 관리자가 선택한 이미지를 커뮤니티 로고 또는 배너로 변경한다. 변경한 이미지를 커뮤니티 DB에 저장한다.
5	이 Use case는 관리자가 자신의 커뮤니티 로고와 배너를 변경을 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
2	2a. 관리자가 적절한 이미지 파일을 선택하고 열은 경우 2b.1. 해당 이미지파일이 미리보기 형식으로 Save 옆에 나타난다. 3a. 관리자가 적절한 이미지 파일을 열었고, Save 버튼을 누른 경우

	3a.1. 해당 이미지가 커뮤니티의 로고 또는 배너사진으로 변경된다.
RELATED INFORMATION	
Performance	≤ 3 Seconds
Frequency	Variable
Concurrency	None
Due Date	

Use Case #14: View community	
GENERAL CHARACTERISTICS	
Summary	User가 Beddit을 통해 개설된 특정 커뮤니티에 접속하려고 할 때 사용하는 기능
Scope	Beddit
Level	User Level
Author	Kim Joo Hwan
Last Update	2023-04-07, 2023-04-20
Status	Analysis
Primary Actor	User, Administrator
Preconditions	커뮤니티가 적어도 1개 개설되어 있어야 한다.
Trigger	User가 'Beddit' 홈화면에 올라온 게시글 중에 자신이 방문하고자 하는 커뮤니티 이름을 클릭한다.
Success Post Condition	User가 특정 커뮤니티 접속에 성공한다. 해당 커뮤니티에 등록된 배너, 로고 이미지와 업로드 된 게시글을 볼 수 있다.
Failed Post Condition	User가 특정 커뮤니티 접속에 실패한다. 해당 커뮤니티는 존재하지 않는다는 메시지가 나타난다.
MAIN SUCCESS SCENARIO	
Step	Action
S	User가 'Beddit'에 로그인 된 상태에서 게시글 목록 또는 투표하고자 하는 게시글의 상세페이지에 접속한다
1	User는 본인이 투표를 원하는 게시글의 Voting 버튼을 누른다.
2	시스템은 User가 작성한 댓글의 내용을 해당 게시글 DB서버에 저장한다.
3	이 Use case는 User가 작성한 댓글을 삭제하는데 성공하면 끝난다.
EXTENTION SCENARIO	
Step	Branching Action
2	2a. User가 이미 투표한 게시글의 경우 2a.1. 해당 Voting 버튼이 활성화가 되어있다. 2b. User가 이미 투표한 게시글의 투표버튼을 누른 경우 2b.1. 유저가 이전에 선택한 투표(좋아요/ 싫어요)를 취소하고 이후 선택하거나 투표 선택 한 것을 취소 한다.
RELATED INFORMATION	
Performance	≤ 10 Seconds
Frequency	Variable
Concurrency	None
Due Date	



### 3. Domain analysis

(다음으로 정의된 인터페이스는 클라이언트의 action 및 Firebase DB 서버에서 가져온 데이터를 이용하여 상태를 dispatch 하여 클라이언트에게 보여줍니다.)

#### 1) Community

생성된 커뮤니티의 정보를 담고 있는 인터페이스입니다.

커뮤니티의 이름, 개설한 사용자의 ID, 커뮤니티 타입, 생성된 날짜, 해당 커뮤니티의 가입자 수, 로고와 배너 URL을 담고 있습니다.

#### 2) CommunitySnippet

유저가 가입한 커뮤니티의 정보를 담고 있는 인터페이스입니다.

커뮤니티이름과 해당 유저가 관리자인지의 여부와 로고와 배너 URL을 담고 있습니다.

#### 3) Post

업로드 된 게시글의 정보를 담고 있는 인터페이스입니다.

게시글의 id, 제목과 내용, 댓글의 개수, 투표 상태, 게시글을 업로드한 사용자의 정보 그리고 해당 게시글이 업로드 된 커뮤니티 정보를 담고 있습니다.

#### 4) PostVote

현재 로그인 된 상태의 사용자가 투표한 게시글의 정보가 담겨 있는 인터페이스입니다.

게시글의 id와 커뮤니티 제목, 투표 여부(1 or -1)가 담고 있습니다.

#### 7) Comment

게시글에 달린 댓글의 정보를 담고 있는 인터페이스입니다.

댓글의 내용과 작성한 사용자의 id와 닉네임, 해당 댓글에 달린 게시글의 정보, 댓글 작성 시점이 담겨 있습니다.

#### 5) DirectoryMenuItem

사용자가 현재 접속해 있는 Directory를 나타내는 인터페이스입니다.

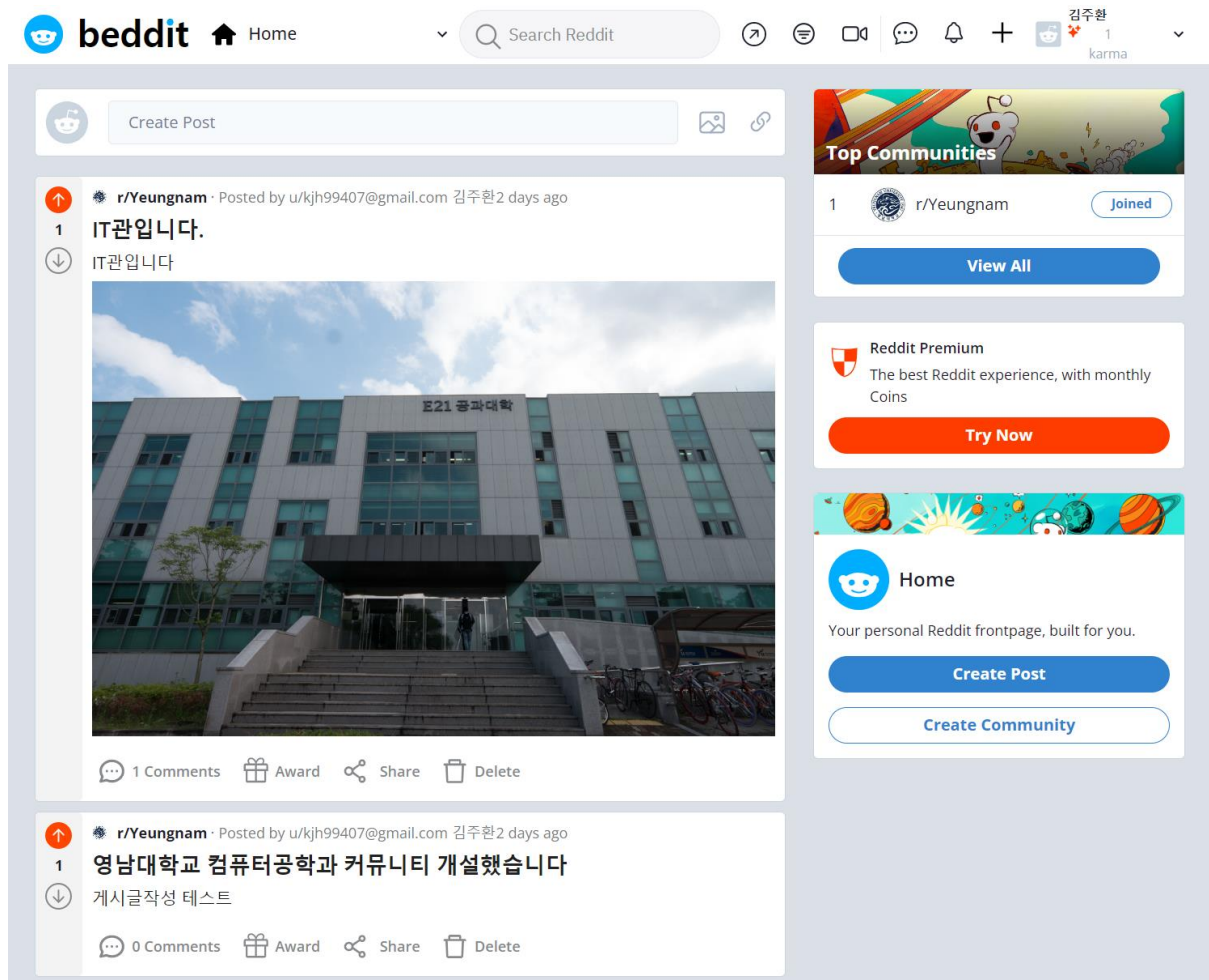
해당 디렉토리의 이름(커뮤니티 제목), 디렉토리의 주소(커뮤니티 주소), 아이콘(커뮤니티 로고) 정보를 담고 있습니다.

#### 6) User

유저 정보를 담고 있는 인터페이스입니다.

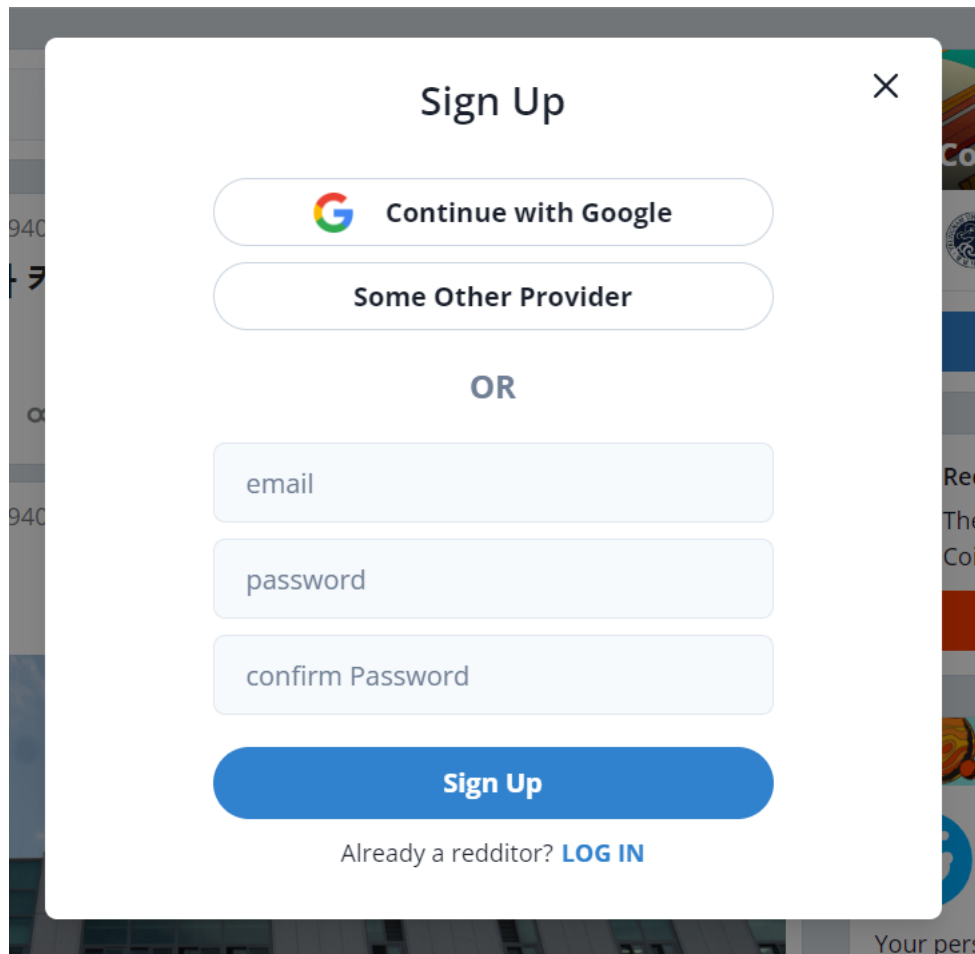
이메일, 닉네임 정보가 담겨 있습니다.

## 4. User Interface prototype



[그림 5-1] 홈 화면

사용자가 'Beddit'을 처음 방문했을 때 다음과 같은 화면이 나타난다. 사용자는 로그인을 위해서는 오른쪽 상단위에 Log in을 회원가입은 Sign up 버튼을 통해 가능하다. 홈 화면에 올라오는 게시글은 각자의 커뮤니티에서 최근에 업로드 된 게시글이 홈 화면에 출력된다. 각 게시글을 클릭하면 해당 게시글의 상세정보(댓글)를 확인할 수 있다. 오른쪽 상단에 Top Communities는 현재 생성된 커뮤니티 리스트를 불러온다.

A screenshot of a 'Sign Up' modal window. The window has a title bar with 'Sign Up' and a close button (X). It contains two main sections for registration. The first section has two buttons: 'Continue with Google' (with the Google logo) and 'Some Other Provider'. Below these is a separator 'OR'. The second section contains three input fields: 'email', 'password', and 'confirm Password'. At the bottom of this section is a large blue 'Sign Up' button. Below the button is a link: 'Already a redditor? LOG IN'. The background of the modal is white, and it is overlaid on a blurred image of a book cover.

Sign Up

Continue with Google

Some Other Provider

OR

email

password

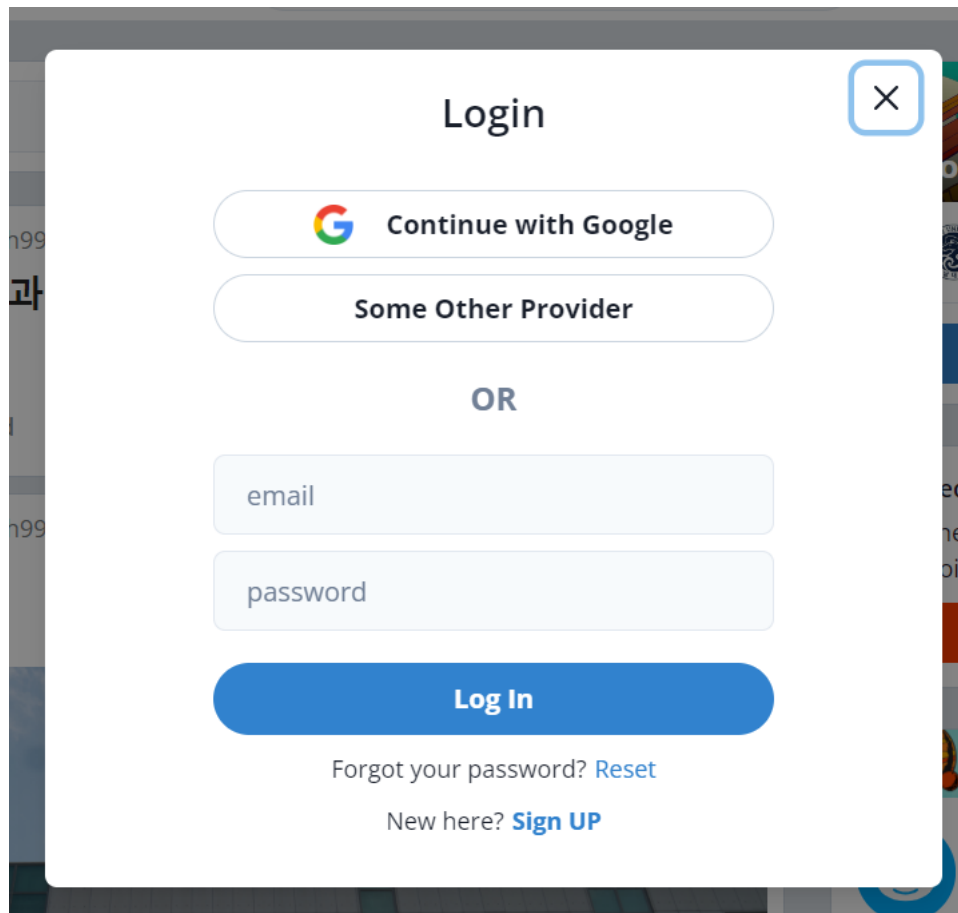
confirm Password

Sign Up


Already a redditor? [LOG IN](#)

[그림 5-2] 회원가입

사용자가 우측 상단에 회원가입 버튼을 클릭했을 때 출력되는 화면이다. 사용자는 자신이 직접 이메일과 패스워드를 통해 가입을 하거나, 구글 아이디를 통해 'Beddit'에 가입할 수 있다.

A login modal window with a white background and a gray border. At the top center is the title "Login" in a bold, black font. To the right of the title is a close button with a black "X" inside a blue square. Below the title are two rounded rectangular buttons. The first button contains the Google "G" logo and the text "Continue with Google". The second button contains the text "Some Other Provider". Below these buttons is the word "OR" in a bold, black font. Underneath "OR" are two stacked input fields. The top field is labeled "email" and the bottom field is labeled "password". Below the input fields is a large, blue, rounded rectangular button with the text "Log In" in white. At the bottom of the modal, there are two lines of text: "Forgot your password? [Reset](#)" and "New here? [Sign UP](#)".

Login

 Continue with Google

Some Other Provider

OR

email

password

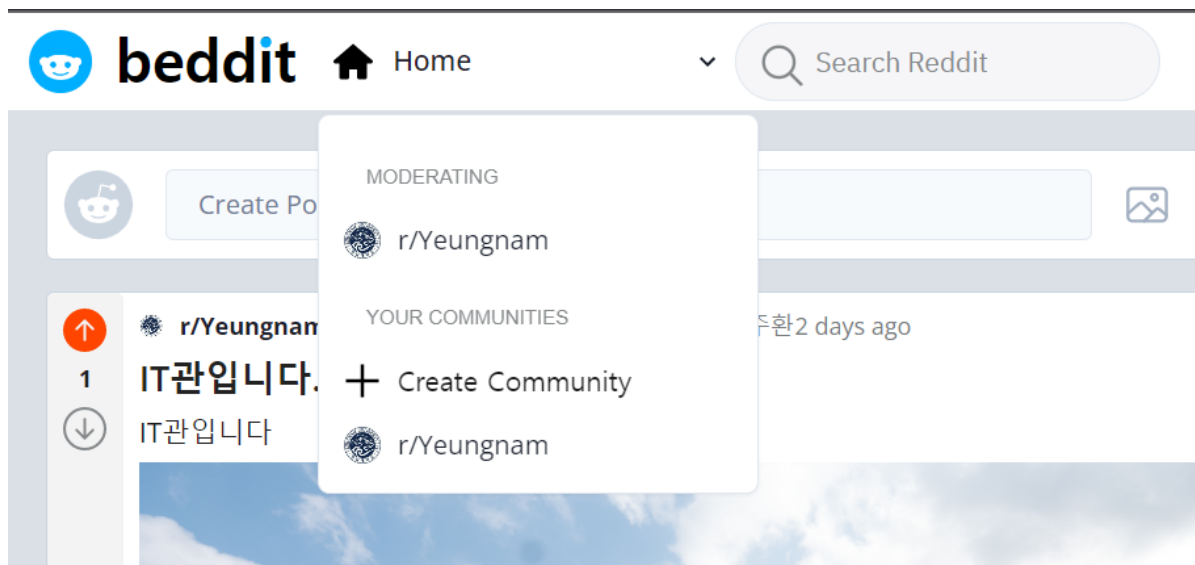
Log In

Forgot your password? [Reset](#)

New here? [Sign UP](#)

[그림 5-3] 로그인

사용자가 우측 상단에 로그인 버튼을 클릭했을 때 출력되는 화면이다. 사용자는 자신이 생성한 아이디를 통해 로그인을 하거나, 구글 아이디를 통해 로그인이 가능하다.



[그림 5-4] 커뮤니티 리스트

사용자가 로그인 한 상태에서 왼쪽 상단에 홈 버튼을 클릭했을 때 출력되는 화면이다. 사용자는 자신이 가입한 커뮤니티의 이름과 로고를 볼 수 있고, 클릭 시 해당 커뮤니티로 접속한다.

**Create a community** [X]

**Name**  
Community names including capitalization cannot be changed. ⓘ

r/

21 Characters remaining

**Community Type**

- ☒ **Public** Anyone can view, post, and comment to this community
- ☐ **Restricted** Anyone can view this community, but only approved users can post
- ☐ **Private** Anyone can view, post, and comment to this community

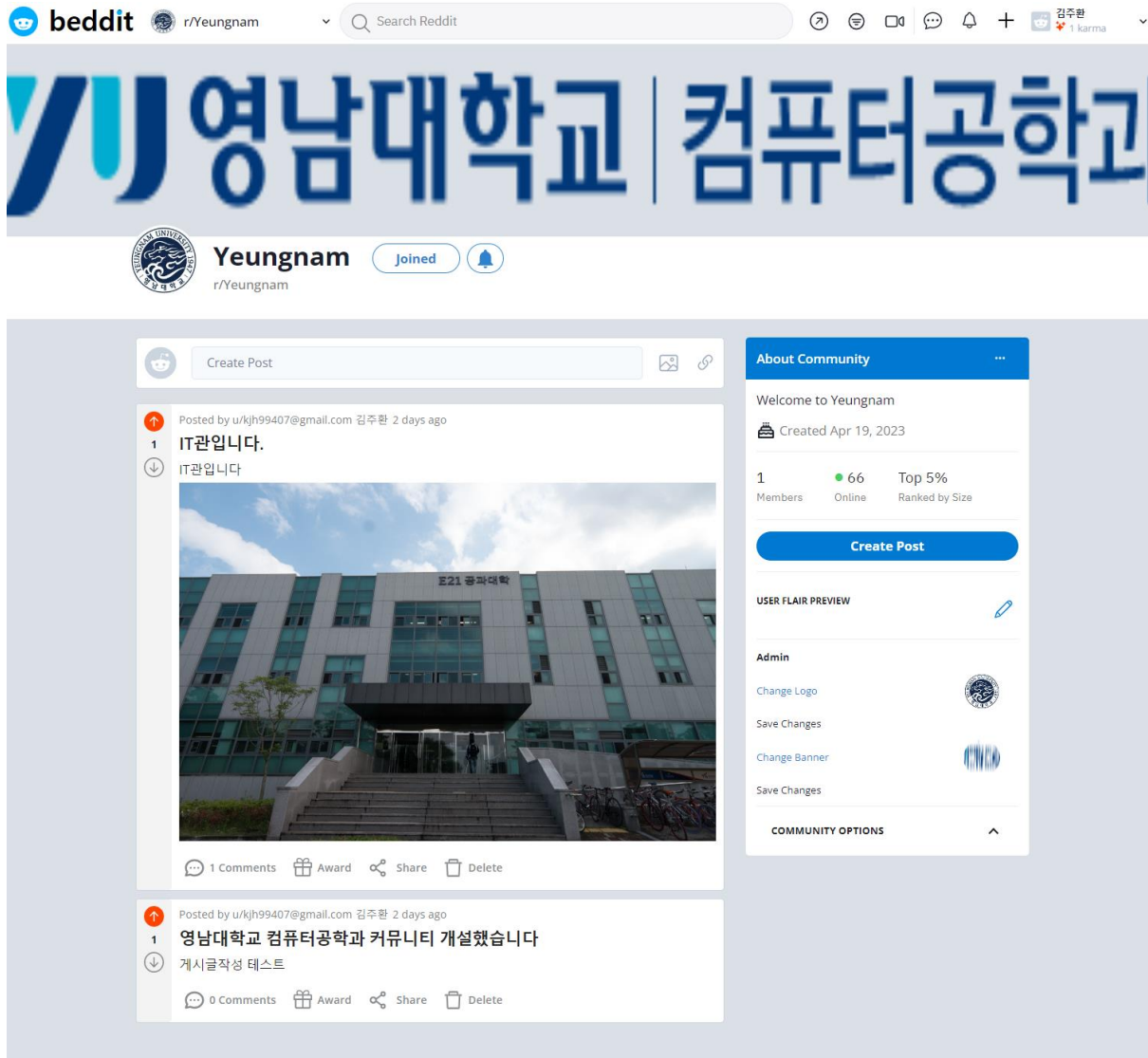
**Adult content**

☐ **NSFW** 18+ year old community

**Cancel** **Create Community**

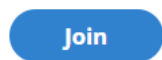
[그림 5-5] 커뮤니티 생성

사용자가 왼쪽 상단에 홈 버튼을 통하여 커뮤니티 생성 버튼을 눌렀을 때, 출력되는 화면이다. 사용자는 자신이 개설하고자 하는 커뮤니티의 이름을 입력하고 해당 커뮤니티의 이름이 존재하는지 여부를 판단한 뒤, 없다면 커뮤니티가 개설되고 해당 커뮤니티 사이트로 접속된다.



[그림 5-6] YU 커뮤니티

‘Beddit’에 개설된 커뮤니티에 접속하면, 해당 커뮤니티의 로고와 배너를 볼 수 있으며, 해당 커뮤니티에서 올라온 게시글 들을 확인할 수 있다. 커뮤니티 로고와 제목 옆에



Join 버튼을 통해 해당 커뮤니티를 가입할 수 있다. 커뮤니티에 이미 가입

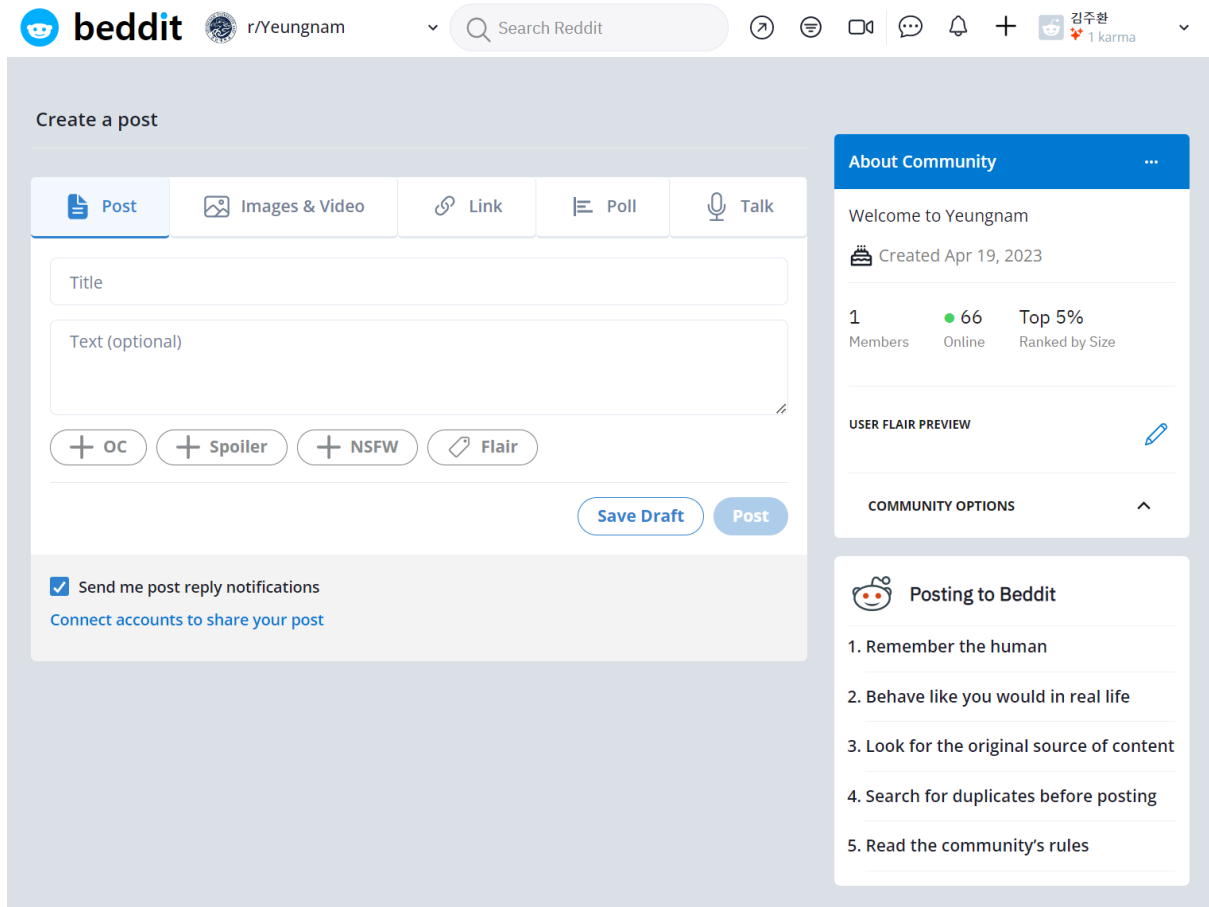
된 유저라면



가 나타난다. 커뮤니티 탈퇴를 원한다면 Joined 버튼을

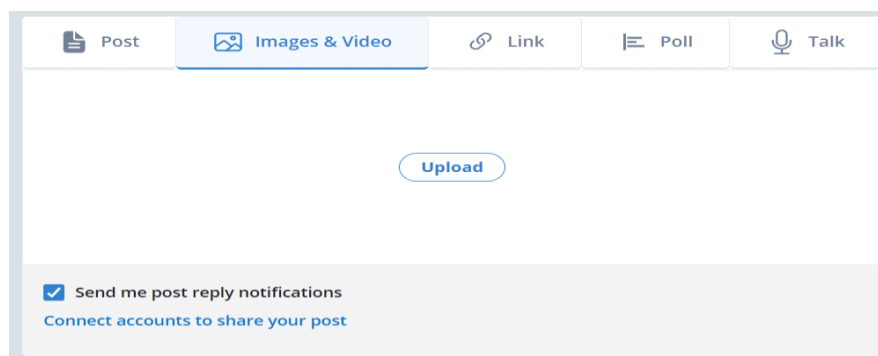
눌러 탈퇴가 가능하다. 관리자라면 Admin 권한이 있어 로고와 배너를 변경할 수 있다.



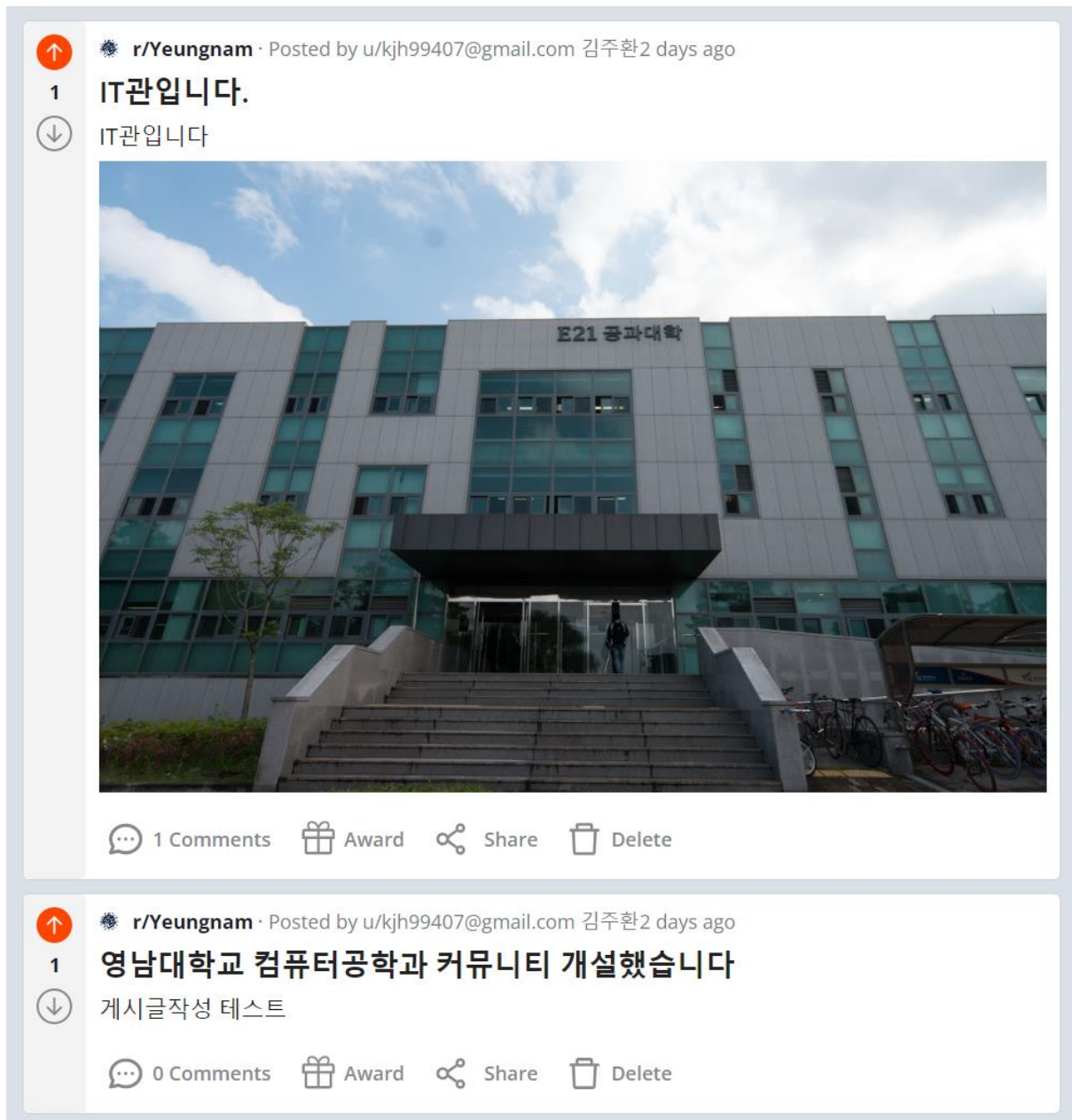


[그림 5-7] 게시글 작성

커뮤니티에서 게시글 작성 버튼을 눌렀을 때, 나오는 화면이다. 위의 버튼들 중 Post에서는 해당 게시글의 제목과 내용을 입력할 수 있고, Images & Video 에서는 이미지 업로드가 가능하다.

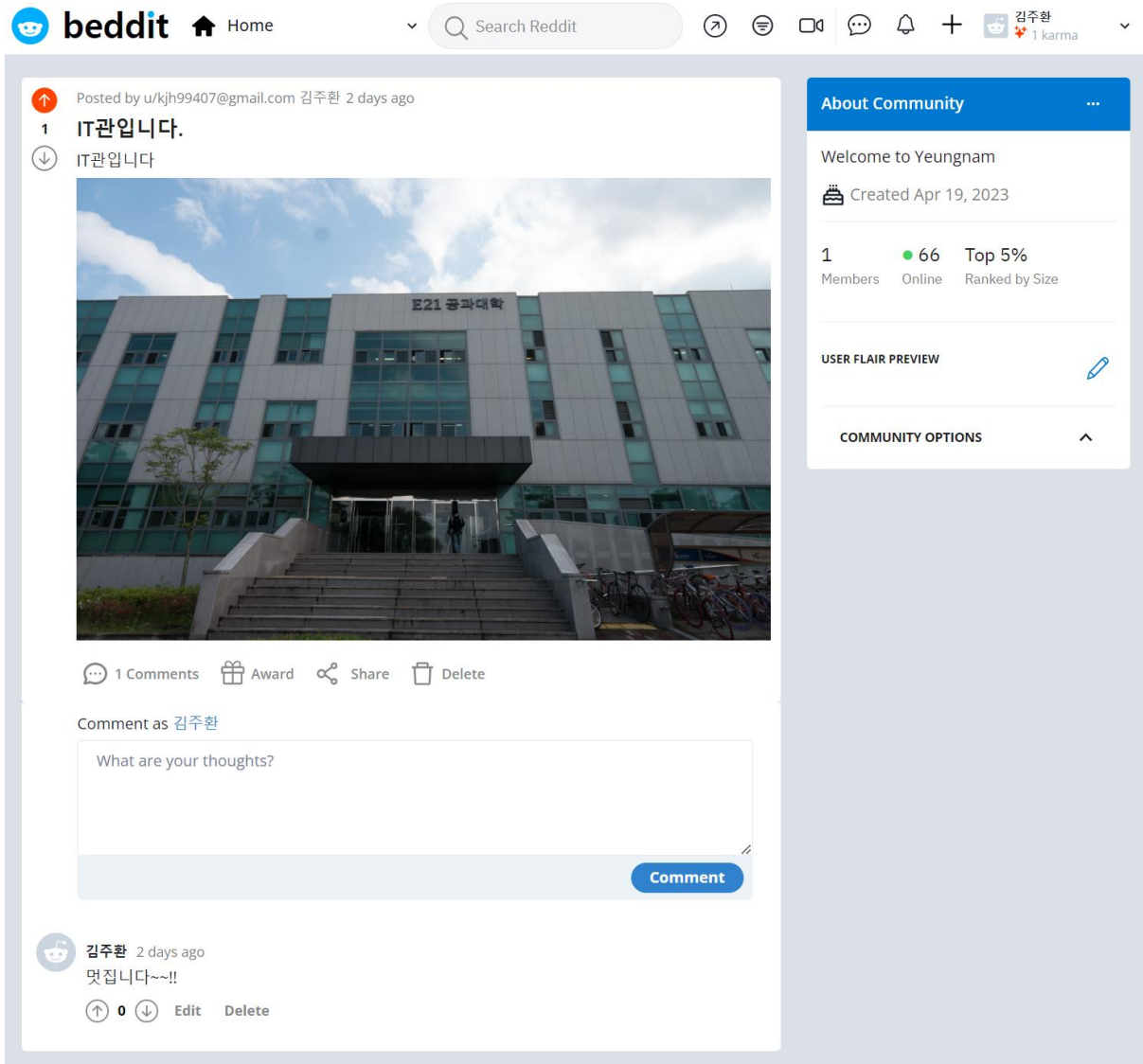


[그림 5-8] 이미지 업로드



[그림 5-9] 게시글

홈 화면 및 커뮤니티에서 게시글을 확인할 수 있다. 해당 게시글에는 게시글의 제목과 내용, 이미지, 작성한 유저의 아이디와 이름이 출력되고 게시글 작성 시점이 얼마나 지났는지 출력된다. 또한 해당 게시글에 댓글이 달렸다면 댓글의 개수 또한 출력된다. 로그인한 유저가 개설한 게시글이라면 Delete 버튼이 활성화된다. 해당 버튼을 통해 유저는 자신이 개설한 게시글을 삭제할 수 있다. 또한 유저는 게시글의 투표 (좋아요 및 싫어요)도 가능하다. 투표를 하게 되면 해당 버튼이 활성화되고 숫자가 변경된다.



[그림 5-10] 게시글 상세 보기

홈 화면 및 커뮤니티에서 게시글을 확인하고 해당 게시글을 클릭하면 게시글 상세 보기 화면을 출력한다. 게시글에 달린 댓글을 확인할 수 있고 로그인되어 있는 상태라면 textarea와 comment 버튼을 통해 게시글의 내용을 입력 후 등록할 수 있다.

## 5. Glossary

Terms	Description
Beddit	유저가 커뮤니티 기능(커뮤니티 개설, 게시글 작성, 가입 )을 이용할 수 있도록 도와주는 시스템
유저 (User)	'Beddit' 에 회원 가입을 한 사용자
관리자 (Administrator)	자신의 커뮤니티를 개설하여 관리하는 유저
커뮤니티	유저가 개설한 커뮤니티
DB 서버 (Firebase)	유저의 정보 및 커뮤니티 정보, 게시글 정보를 저장하기 위한 서버

## 6. References

[Firebase Authentication]

<https://firebase.google.com/docs/auth>

[React의 상태 값과 관리]

<https://naon.me/posts/til104>

[Typescript의 Type 정의]

<https://www.typescriptlang.org/ko/docs/handbook/utility-types.html>