



Beddit

Conceptualization Document

21812062 김주환

Revision History

Revision date	Version #	Description	Author
2023.03.15	1.0	First Draft	김주환
2023.03.18	1.1	Use case list Concept of operation	김주환
2023.03.19	1.2	Problem State Gloassary	김주환
2023.04.22	1.3	Design	김주환

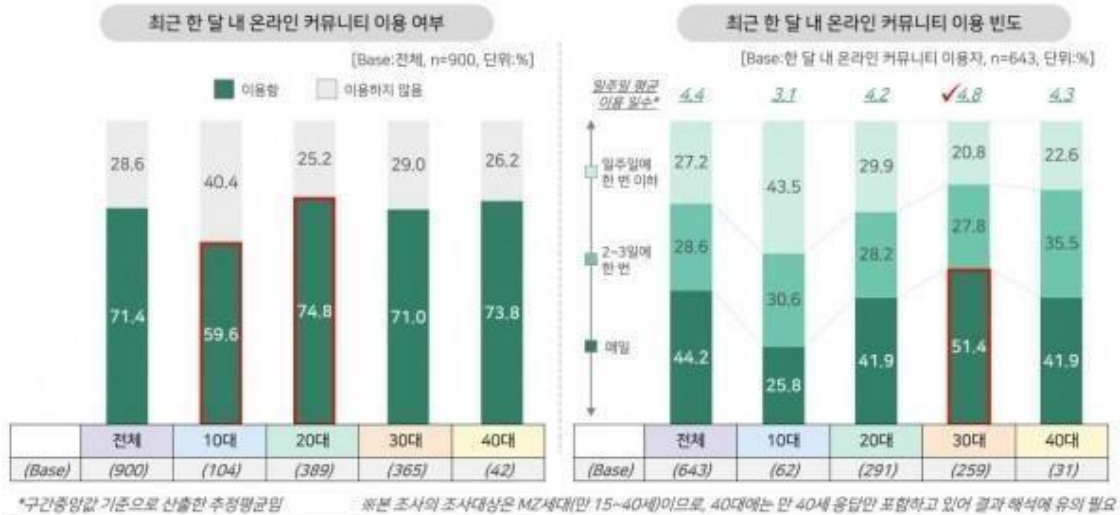
Contents

1. Business purpose.....	4
2. System Context Diagram	5
3. Use case list	7
4. Concept of operation.....	10
5. Problem statement.....	16
6. Glossary.....	17
7. References.....	18

1. Business purpose

MZ세대의 70% 이상이 온라인 커뮤니티를 이용하고 있고, 그 중 44.2%는 매일 이용 중

- MZ세대의 71.4%는 최근 한 달 내 온라인 커뮤니티를 이용한 적 있고, 온라인 커뮤니티 이용 경험자는 일주일에 평균 4.4일 이용함
- 한 달 내 온라인 커뮤니티를 이용한 비율은 20대가 74.8%로 가장 높은 반면, 10대는 59.6%로 다른 연령 대비 온라인 커뮤니티 이용률이 낮은 편
- 한편 30대 온라인 커뮤니티 이용자는 일주일에 평균 4.8일 이용 중이며, 매일 이용하는 비율이 51.4%로 전 연령대 중 가장 활발함

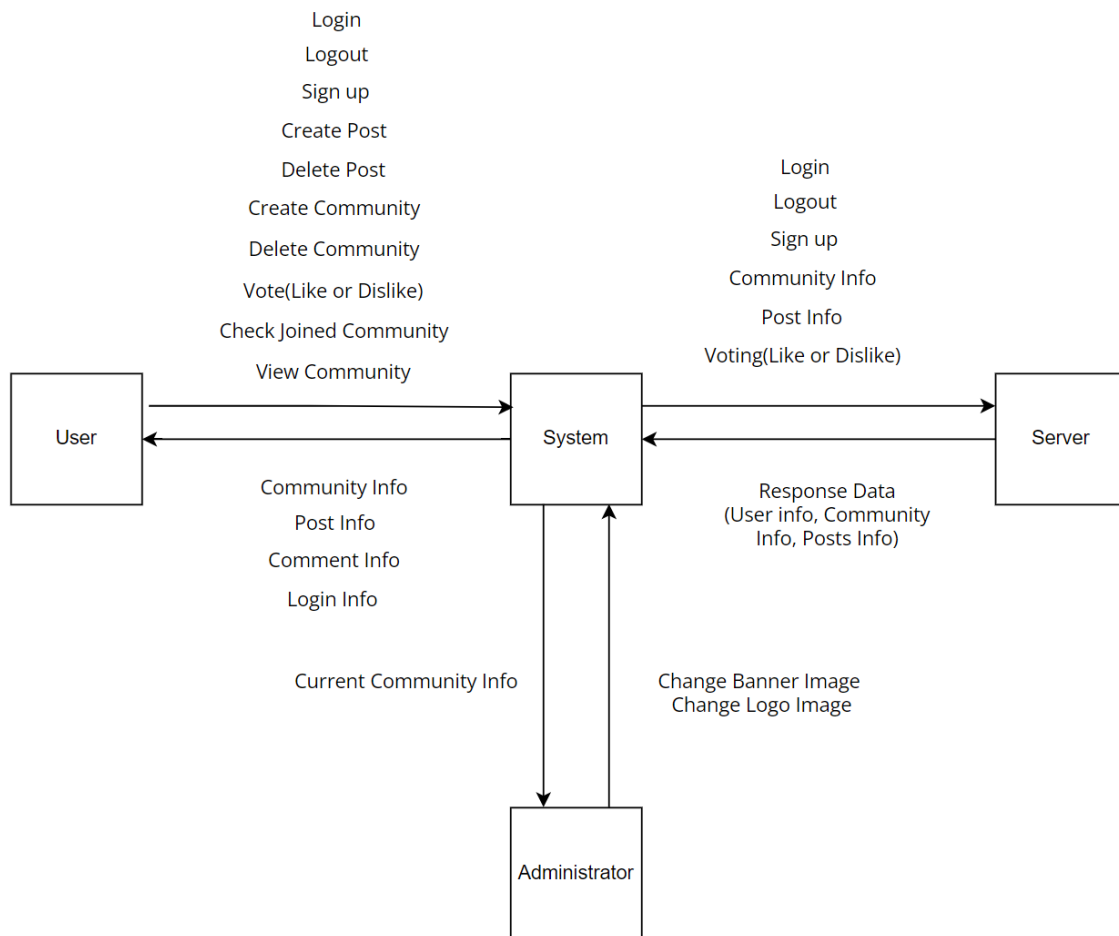


[그림 1-1]

전세계적으로 MZ 세대의 온라인 커뮤니티의 이용률은 매우 높다. 2000 년대 초반만 하더라도 커뮤니티는 공론장의 역할로서 서로 토론을 위한 역할을 했었다 하지만. 오늘날 여러 대형 커뮤니티(Reddit, DC inside, Naver Cafe) 등은 서로의 관심사가 맞는 사람들끼리 모여서 토론보다는, 유희를 즐기는 장으로 변화하였다 다시 말해. 놀거리를 찾는 MZ 세대가 모인 곳이라고 할 수 있다.

이러한 커뮤니티의 이용률이 높은 것을 바탕으로 MZ 세대를 타겟으로 한 광고를 게시하거나 해당, 커뮤니티의 브랜드 이미지를 높일 수 있는 기대효과를 가져올 수 있다. 뿐만 아니라 커뮤니티, 웹페이지는 사용자가 특정 관련 주제에 대한 새로운 아이디어를 제시하고 피드백을, 주고받는 공간이 될 수 있으므로 이를 통해 새로운 제품이나 서비스를 개발할 때 유용한 아이디어를 발굴해낼 수 있다.

2. System Context Diagram



[그림 2-1] use context diagram

- ① Login : 로그인
- ② Logout : 로그아웃
- ③ Sign up : 회원가입
- ④ Create Post : 게시글 작성
- ⑤ Delete Post : 게시글 삭제
- ⑥ Create Community : 커뮤니티 개설

- ⑦ Join Community : 커뮤니티 가입
- ⑧ Leave Community : 커뮤니티 탈퇴
- ⑨ Create Comment : 댓글 작성
- ⑩ Delete Comment : 댓글 삭제
- ⑪ Check Joined Community : 가입된 커뮤니티 확인
- ⑫ Voting : 게시글의 투표(좋아요 및 싫어요)
- ⑬ View Community : 개설된 특정 커뮤니티 접속
- ⑭ Change Banner, Logo Image : 배너와 로고 이미지를 변경

3. Use case list

1) Login	
Actor	User
Description	유저가 아이디와 비밀번호를 입력하여 로그인을 한다. 시스템은 서버에게서 해당 정보를 받아 로그인 성공 여부 유저에게 반환한다.

2) Logout	
Actor	User
Description	유저가 로그아웃을 한다. 시스템은 해당 유저의 로그인 상태를 조정한다.

3) Sign up	
Actor	User
Description	유저가 아이디와 비밀번호를 입력하여 회원가입을 한다. 시스템은 서버에게 해당 ID 와 비밀번호를 전송하여 서버는 해당 정보를 확인후 승인여부를 전송한다.

4) Create Post	
Actor	User
Description	유저가 특정 커뮤니티에 게시글을 업로드한다. 유저가 특정 커뮤니티에 게시글을 업로드한다.

5) Delete Post	
Actor	User
Description	유저가 자신이 쓴 게시글을 삭제한다.

6) Create Community	
Actor	User
Description	유저가 자신이 원하는 커뮤니티를 생성한다.

7) Join Community	
Actor	User
Description	유저가 자신이 선택한 커뮤니티에 가입한다.

8) Leave Community	
Actor	User
Description	유저가 탈퇴를 원하는 커뮤니티를 탈퇴한다.

9) Create Comment	
Actor	User
Description	유저가 선택한 게시글의 댓글을 등록한다.

10) Delete Comment	
Actor	User
Description	유저가 자신이 작성한 게시글의 댓글을 삭제한다.

11) Check Joined Community	
Actor	User
Description	유저가 자신이 가입한 커뮤니티에 대한 목록을 메뉴 리스트에서 확인한다.

12) Voting	
Actor	User
Description	유저가 특정 게시글의 좋아요 및 싫어요를 투표한다.

13) View Community	
Actor	User
Description	유저가 다른 유저가 생성한 커뮤니티를 접속한다.

13) Change Banner, Logo Image	
Actor	User
Description	커뮤니티 관리자가 자신의 커뮤니티의 배너와 로고 이미지를 변경한다.

4. Concept of operation

1) Login	
Purpose	커뮤니티를 이용하기 위해 등록된 사용자인지 확인
Approach	사용자가 웹사이트에 접속 후 ID, PW를 입력하고 로그인을 요청하면 서버에서 회원 정보를 조회 후 로그인 성공 및 실패 여부를 확인한다.
Dynamics	사용자가 웹사이트를 처음 실행하고 로그인을 하지 않은 상태에서 커뮤니티 게시물 작성 및 투표, 커뮤니티 가입 등을 할 경우 와 사용자가 직접 로그인 버튼을 누른 경우
Goals	로그인을 한 유저에게 커뮤니티의 기능을 제공하게 한다.

2) Logout	
Purpose	다른 유저로 커뮤니티를 이용하고 싶은 경우 다른 아이디로 로그인을 가능하도록 유저를 로그아웃
Approach	사용자가 커뮤니티에 접속 후 로그아웃 버튼을 누르게 되면, 서버에서 해당 유저의 로그인 상태를 조정한다.
Dynamics	사용자가 커뮤니티에 접속 후 로그인 된 상태에서 로그아웃을 요청한 경우
Goals	로그인을 한 유저가 다른 아이디로 로그인 할 수 있도록 한다.

3) Sign up	
Purpose	커뮤니티의 기능을 이용하기 위해 유저 등록
Approach	사용자가 커뮤니티에 접속 후 회원가입 버튼을 누르고, ID와 PW를 입력하면 서버에서 해당 유저의 회원가입 승인 여부를 반환한다.
Dynamics	사용자가 커뮤니티에 접속 후 로그인이 안된 상태에서 회원가입을 요청한 경우
Goals	로그인을 안 한 사용자가 회원 가입을 통해 커뮤니티의 기능을 제공받는다.

4) Create Post	
Purpose	사용자가 커뮤니티에 게시글을 업로드 할 수 있게 한다.
Approach	사용자가 특정 커뮤니티에 접속 후 해당 커뮤니티에 글 제목과 내용을 포함한 게시글을 업로드하면, 서버에서 해당 유저의 게시글 업로드 성공 여부를 반환한다.
Dynamics	사용자가 로그인 하고 커뮤니티에 접속 후 게시글 제목과 내용 작성 후 업로드를 요청한 경우
Goals	로그인 한 사용자의 게시글 업로드를 통해 커뮤니티에 게시글이 업로드 된다.

5) Delete Post	
Purpose	사용자 본인이 커뮤니티에 올린 게시글을 삭제할 수 있게 한다.

Approach	사용자가 특정 커뮤니티에 접속 후 해당 커뮤니티에 업로드 되어있는 게시글 중에 자신이 작성한 게시글을 삭제버튼을 눌러 삭제하면, 서버에서 해당 커뮤니티의 게시글 데이터를 삭제한다.
Dynamics	사용자가 로그인 하고 커뮤니티에 접속 후 자신이 게시한 게시글의 삭제요청을 한 경우
Goals	로그인 한 사용자가 자신이 작성한 게시글 들 중 표시되길 원하지 않는 게시글이 있다면 삭제한다.

6) Create Community	
Purpose	사용자가 생성하고 싶은 커뮤니티를 생성하게 한다.
Approach	사용자가 상단에 있는 메뉴아이템에서 Create Community 버튼을 눌러 커뮤니티 이름을 작성 후 커뮤니티를 생성한다.
Dynamics	사용자가 로그인을 하고 자신이 원하는 커뮤니티 이름을 입력 후 커뮤니티 생성을 요청을 한 경우
Goals	로그인 한 사용자가 자신이 원하는 커뮤니티를 생성할 수 있게 한다.

7) Join Community	
Purpose	사용자가 가입하고 싶은 커뮤니티를 가입하게 한다.
Approach	사용자가 특정 커뮤니티에 접속 후 해당 커뮤니티에 상단에 있는 Join 버튼을 눌러 커뮤니티에 가입을 한다.

Dynamics	사용자가 로그인을 하고 커뮤니티에 접속 후 자신이 원하는 커뮤니티에 가입을 요청한 경우
Goals	로그인 한 사용자가 자신이 원하는 커뮤니티에 가입하게 한다.

8) Leave Community	
Purpose	사용자가 탈퇴하고 싶은 커뮤니티를 탈퇴하게 한다.
Approach	사용자가 특정 커뮤니티에 접속 후 해당 커뮤니티에 상단에 있는 Leave 버튼을 눌러 커뮤니티를 탈퇴한다.
Dynamics	사용자가 로그인을 하고 자신이 탈퇴하고자 하는 커뮤니티에 접속 후 탈퇴 요청을 한 경우
Goals	로그인 한 사용자가 자신이 탈퇴하고자 하는 커뮤니티를 탈퇴하게 한다.

9) Create Comment	
Purpose	사용자가 등록하고 싶은 댓글을 등록하게 한다.
Approach	사용자가 게시글을 클릭하여 세부 게시글 정보로 이동한 뒤 댓글 내용을 입력 후 등록버튼을 누른다.
Dynamics	사용자가 로그인을 하고 게시글의 댓글을 등록 요청한 경우
Goals	로그인 한 사용자가 자신이 원하는 게시글의 댓글을 등록하게 한다.

10) Delete Comment	
Purpose	사용자가 등록하고 싶은 댓글을 등록하게 한다.
Approach	사용자가 게시글을 클릭하여 세부 게시글 정보로 이동한 뒤 댓글 내용을 입력 후 등록버튼을 누른다.
Dynamics	사용자가 로그인을 하고 게시글의 댓글을 등록 요청한 경우
Goals	로그인 한 사용자가 자신이 원하는 게시글의 댓글을 등록하게 한다.

11) Check Joined Community	
Purpose	사용자가 자신이 가입한 커뮤니티의 목록을 확인하게 한다.
Approach	사용자가 상단에 있는 메뉴아이템을 클릭한다.
Dynamics	사용자가 메뉴아이템을 클릭하여 메뉴리스트를 요청한 경우
Goals	로그인 한 사용자가 자신이 가입한 커뮤니티의 목록을 확인할 수 있게 한다.

12) Voting	
Purpose	사용자가 특정 게시글의 투표를 할 수 있게 한다.
Approach	사용자가 게시글 왼쪽에 Up 버튼과 Down 버튼을 눌러 게시글 투표를 할 수 있다.

Dynamics	사용자가 로그인을 하고 특정 게시글의 투표 요청을 한 경우
Goals	로그인 한 사용자가 자신이 원하는 게시글의 좋아요 및 싫어요 라는 의견을 투표로 통해서 할 수 있게 한다.

13) View Community	
Purpose	사용자가 등록하고 싶은 댓글을 등록하게 한다.
Approach	사용자가 게시글을 클릭하여 세부 게시글 정보로 이동한 뒤 댓글 내용을 입력 후 등록버튼을 누른다.
Dynamics	사용자가 로그인을 하고 게시글의 댓글을 등록 요청한 경우
Goals	로그인 한 사용자가 자신이 원하는 게시글의 댓글을 등록하게 한다.

14) Change Banner, Logo Image	
Purpose	커뮤니티 관리자가 자신의 커뮤니티 이미지를 변경하게 한다.
Approach	로그인을 한 커뮤니티 관리자가 자신의 커뮤니티 오른쪽 중간에 이미지 변경 버튼을 누르면 이미지를 선택하고 변경할 수 있습니다.
Dynamics	커뮤니티 관리자가 로그인을 하고 커뮤니티 이미지 변경을 요청했을 경우
Goals	관리자가 자신이 원하는 이미지로 커뮤니티 이미지를 변경할 수 있게 한다.

5. Problem statement

Beddit은 해외 대형 커뮤니티인 **Reddit**을 예시로 삼아서 사용자들이 누구나 쉽게 자신의 커뮤니티를 만들고, 각자의 관심사에 따라 다양한 주제에 대해 이야기하고 의견을 공유할 수 있는 웹사이트입니다.

Problem#1

해당 웹사이트의 인증, 데이터베이스, 스토리지를 **FireBase** 라는 해외 플랫폼 업체를 이용하여 구현할 것이다. 간혹 해당 업체의 **API** 응답속도가 느려질 경우가 있다. 해당 경우를 대비하기위해 대안이 될 수 있는 **MongoDB**에 대해 조금 더 공부할 필요가 있어 보인다. 그러나 이 프로젝트의 규모는 작고 빠른 시일내에 **NoSQL**을 사용하여 구축하기 위해 **Firestore**를 채택하여 개발할 것이다.

Problem#2

해당 프로젝트를 구현하기 위해 해외 대형 커뮤니티 **Reddit**을 참고 삼아 진행하였다.

Reddit의 UI는 매우 정교하고 세련되었다. 해당 UI를 그대로 구현하기 위해

Chakra UI 와 **Material UI**, **Bootstrap** 중에 하나의 웹 프레임워크

라이브러리를 이용할 생각이다. 해당 라이브러리에 대한 충분한 사전 지식이 필요할 것으로 생각된다.

해당 프로젝트 **Beddit** 은 **Reddit** 과 유사하게 만드는 것이 목표이고, 실제 상업적으로 이용할 목적은 없습니다.

6. Glossary

용어	설명
Reddit	Reddit 은 인터넷에서 가장 인기 있는 사이트 중 하나로, 사용자들이 올리는 게시물들을 공유하고 댓글을 달아 의견을 나눌 수 있는 소셜 뉴스 웹사이트입니다. 2005 년에 설립되었으며, 온라인 커뮤니티와 포럼을 결합한 형태로 운영됩니다.
Firebase	Firebase는 구글에서 개발한 클라우드 기반 모바일 및 웹 애플리케이션 개발 플랫폼입니다. 앱 개발자들이 서버나 인프라를 관리하는 부담 없이 고품질의 애플리케이션을 더욱 쉽게 개발할 수 있도록 도와줍니다.
UI	UI는 "User Interface"의 약자로, 사용자 인터페이스를 의미합니다. 사용자 인터페이스는 사용자가 디지털 기기나 소프트웨어를 사용하는 데 필요한 일련의 요소와 상호 작용을 말합니다. 즉, UI는 사용자와 컴퓨터 또는 다른 기기 사이의 상호작용을 가능케 하는 기술적 요소를 포함한 모든 디자인 요소를 포함합니다.
NoSQL	NoSQL은 "Not only SQL"의 약자로, 관계형 데이터베이스(RDBMS)가 아닌 다른 형태의 데이터 저장 기술을 의미합니다.

7. References

[그림 1]

<http://www.casenews.co.kr/news/articleView.html?idxno=4272>

[Firebase]

<https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%9D%B4%EC%96%B4%EB%B2%A0%EC%9D%B4%EC%8A%A4>