



Beddit

Analysis Document

21812062 김주환

Revision History

Revision date	Version #	Description	Author
2023.05.15	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	9
4. User Interface prototype 오류! 책갈피가 정의되어 있지 않습니다.	
5. Glossary	오류! 책갈피가 정의되어 있지 않습니다.
6. References	22

1. Introduction

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

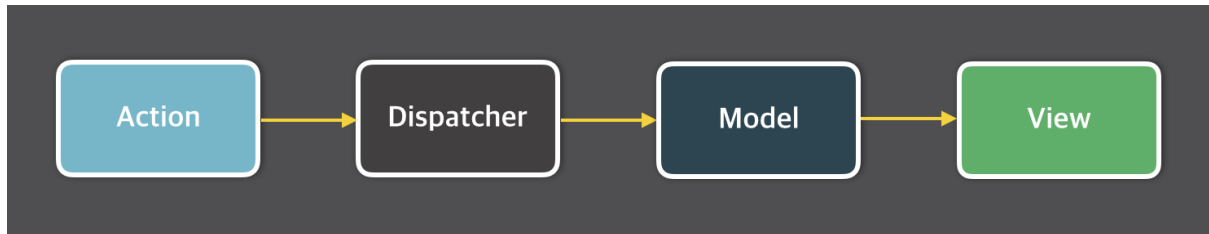
또한 User는 커뮤니티 그룹이 세분화되어 이전과 다르게 자신이 관심있는 공통 주제에 대한 소통을 세분화되어 있는 커뮤니티에서 같은 집단에 속해져 있는 듯한 느낌을 받으며 유대 관계 형성에 도움을 받기도 한다. 기업이 서비스와 함께 커뮤니티를 제공해야 하는 이유는 고객의 체류 시간, 특정 목적이 없어도 해당 앱을 습관적으로 접속한다. SNS를 하듯, 특별한 이유 없이 해당 앱을 자주 방문하고 오래 머물 수 있다.

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.

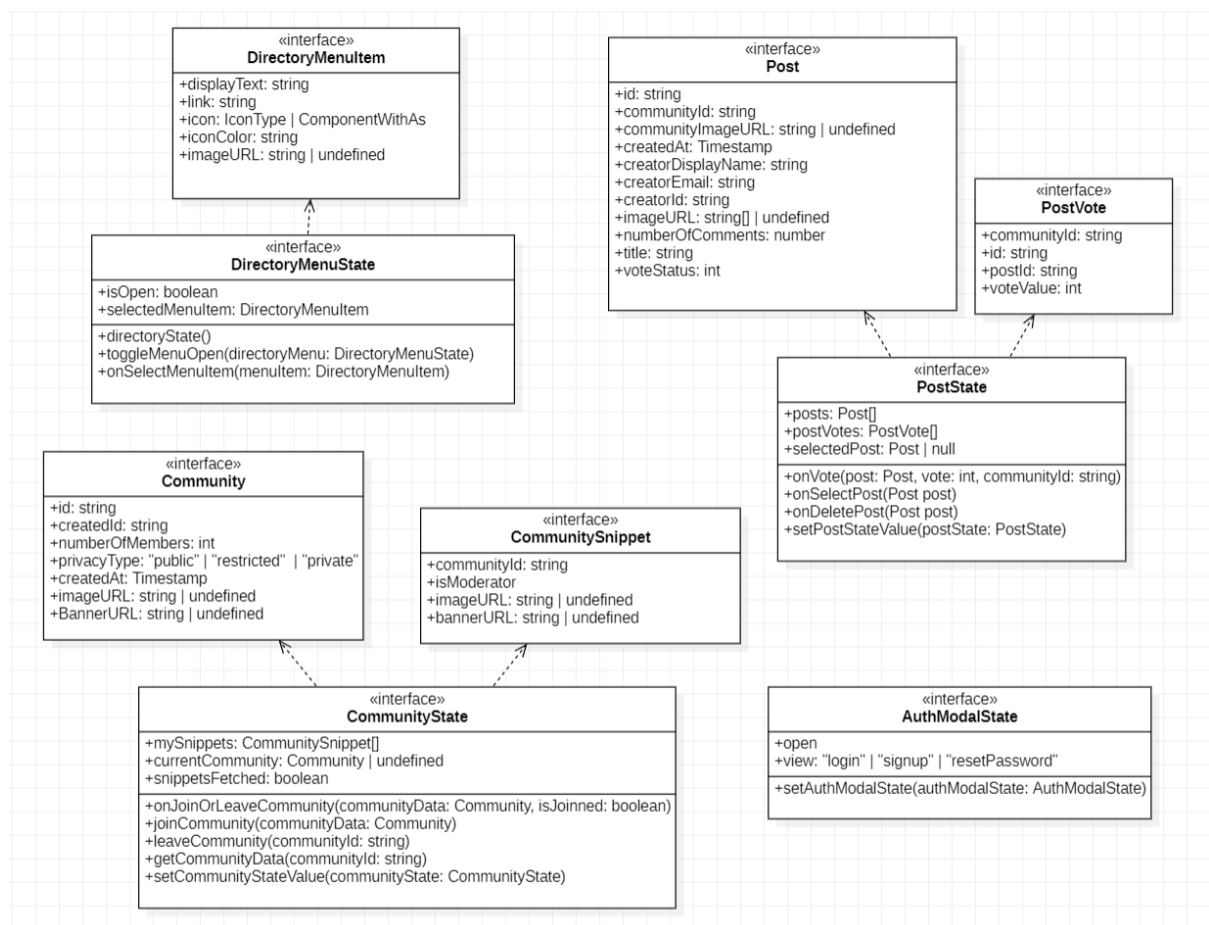
2. Class Diagram

본 프로젝트는 MVC 디자인 패턴이 아닌 Flux 디자인 패턴으로 구현 되어있고 사용자의 Action에 따라 Dispatcher가 Store(Model)의 상태를 변경하고 View에 반영하는 방식입니다.



따라서 해당 프로젝트의 Flux 패턴을 Class diagram으로 표현하기 위해서는 Store(Model)의 State 상태를 정의하는 Interface를 Class diagram으로 나타내었습니다. 해당 Class diagram의 Methods에 정의된 것은 파라미터로 전달된 State를 이전에 저장된 State에서 새롭게 업데이트 및 저장할 State로 Dispatch 하는 기능을 담고 있습니다. (해당 Method는 실제로는 Custom hook의 함수로서 인터페이스 외부에 함수의 형태로 따로 구현되어 있습니다.) 로그인, 회원가입은 컴포넌트 함수로 이루어져 있어서 클래스 다이어그램으로 나타내는데 한계가 있음으로 따로 class diagram에 나타내진 못했습니다.

[그림 2-1] class diagram



1) Community

Attributes
Id : 커뮤니티가 가지는 고유한 값 (Identity) createdId : 커뮤니티를 개설한 유저의 UID 값 numberOfMembers : 커뮤니티에 가입된 멤버 수 privacyType : 커뮤니티의 공개 여부 createdAt : 커뮤니티가 생성된 날짜와 시각 imageUrl : 커뮤니티가 가지는 이미지 링크 주소 BannerURL : 커뮤니티가 가지는 배너 링크 주소
Methods

2) CommunitySnippet

Attributes
communityId : 커뮤니티가 가지는 고유한 값 (Identity) isModerator : 커뮤니티를 개설한 유저의 UID 값 imageUrl : 커뮤니티가 가지는 이미지 링크 주소 BannerURL : 커뮤니티가 가지는 배너 링크 주소
Methods

3) CommunityState

Attributes
mySnippets : 사용자가 가입한 커뮤니티 목록 상태 CommunitySnippet 배열 currentCommunity : 사용자가 어느 커뮤니티에 접속하고 있는지 상태를 나타내는 Community 객체 snippetsFetched : 사용자가 커뮤니티에 로그인을 하고 커뮤니티 상태가 업데이트 되었는지 여부
Methods
onJoinOrLeaveCommunity(communityData: Community, isJoined: boolean) : 매개변수로 입력받은 커뮤니티에 가입했는지 여부를 확인후 가입이 안되어 있다면 가입을 하고, 가입이 되어 있다면 탈퇴를 수행한다. joinCommunity(communityData: Community) : 매개변수로 입력받은 커뮤니티에 가입을 수행한다. leaveCommunity(communityData: Community) : 매개변수로 입력받은 커뮤니티에 탈퇴를 수행한다. getCommunityData(communityId: string) : 매개변수로 입력받은 커뮤니티 ID를 통해 데이터베이스에서 해당 커뮤니티의 정보를 불러오고 그 정보로 사용자의 currentCommunity 상태를 업데이트한다.

setCommunityStateValue(communityState: CommunityState) : 매개변수로 입력받은 커뮤니티 상태를 사용자의 CommunityState로 업데이트 한다.

4) Post

Attributes
<p>Id : 게시글이 가지는 고유한 값 (Identity)</p> <p>communityId : 게시글이 올려진 커뮤니티의 Id 값</p> <p>communityImageUrl : 게시글이 올라간 커뮤니티의 이미지 URL 주소</p> <p>createdAt : 게시글이 생성된 날짜</p> <p>title: 게시글의 제목</p> <p>body: 게시글의 내용</p> <p>numberOfComments: 해당 게시글의 작성된 댓글 수</p> <p>voteStatus: 사용자의 게시글 좋아요(투표) 여부</p> <p>imageUrl: 게시글에 첨부된 이미지 링크 주소</p> <p>creatorDisplayName : 게시글을 작성한 작성자의 닉네임 또는 이름</p> <p>creatorEmail : 게시글을 작성한 작성자의 이메일 주소</p> <p>creatorId : 게시글을 작성한 작성자의 UID</p>
Methods

5) PostVote

Attributes
<p>Id : PostVote가 가지는 고유한 값 (Identity)</p> <p>communityId : 게시글이 올려진 커뮤니티의 Id 값</p> <p>postId : 게시글이 가지는 고유한 값 (Identity)</p> <p>voteValue : 사용자가 해당 게시글에 대한 투표 여부를 나타낸다.</p>
Methods

6) PostState

Attributes
<p>posts : 게시글의 상태를 저장하기위한 Post 배열</p> <p>postVotes : 게시글의 투표 상태를 저장하기위한 PostVote 배열 속성</p> <p>selectedPost : 현재 선택된 게시글이 있는지 여부를 나타내는 상태 속성</p>
Methods
<p>setPostStateValue(postState: PostState) : 매개변수로 입력받은 게시글 상태를 사용자의</p>

PostState로 업데이트 한다.

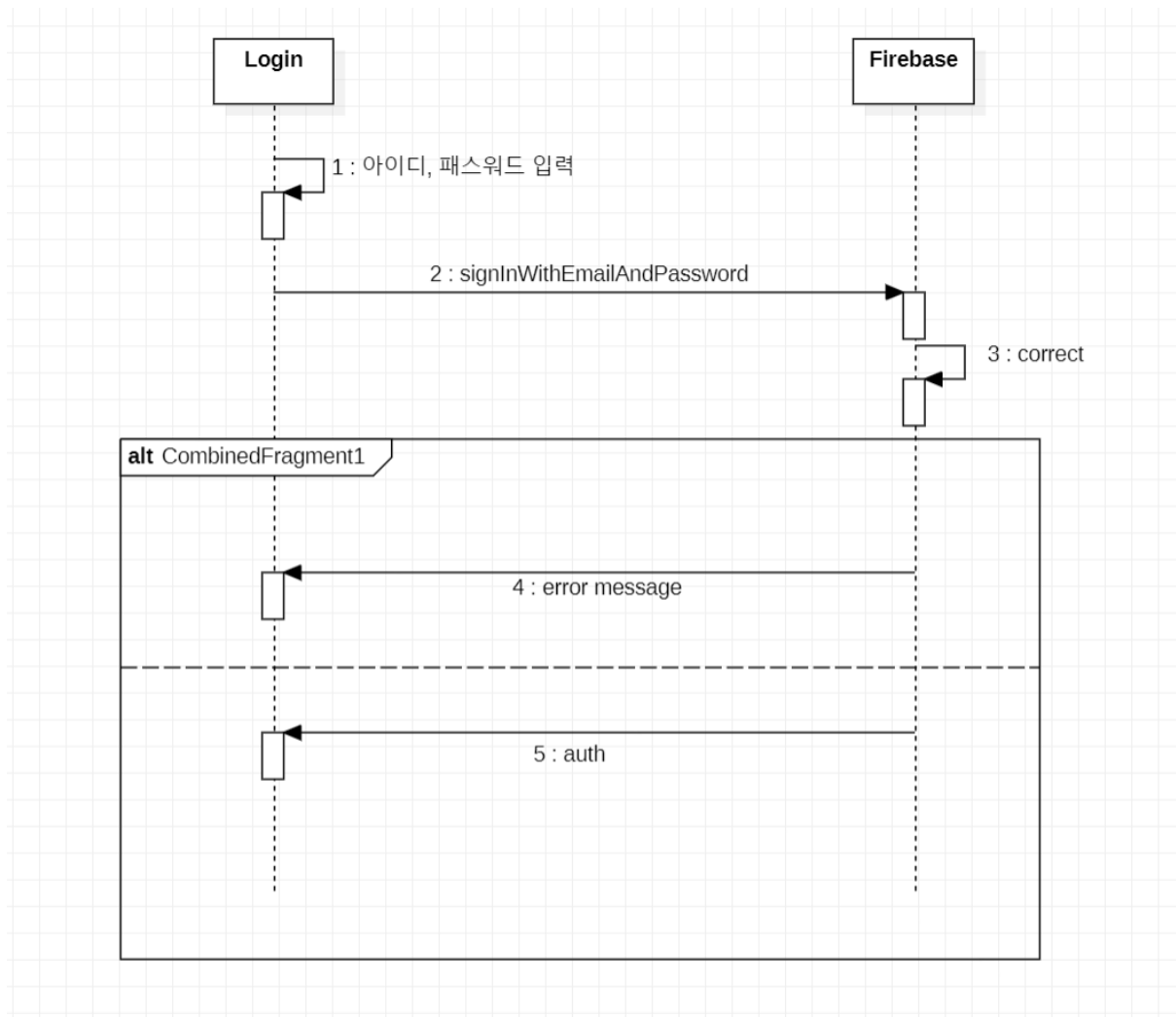
onVote(post: Post) : 게시글에 투표 기능을 제공

onSelectPost(post: Post) : 게시글의 상세 페이지 보기

onDeletePost(post: Post) : 게시글 삭제 기능을 제공

3. Sequence Diagram

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

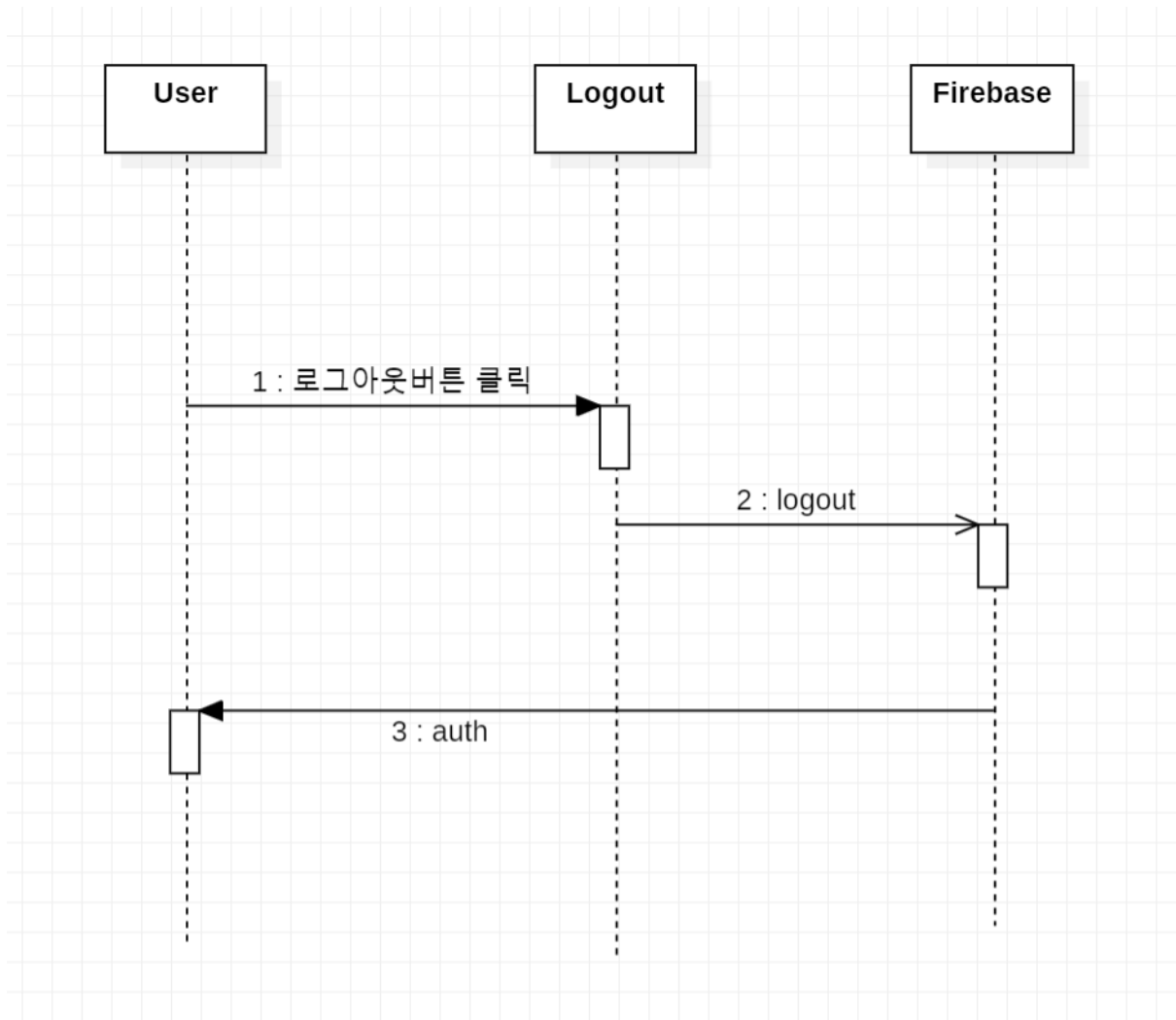


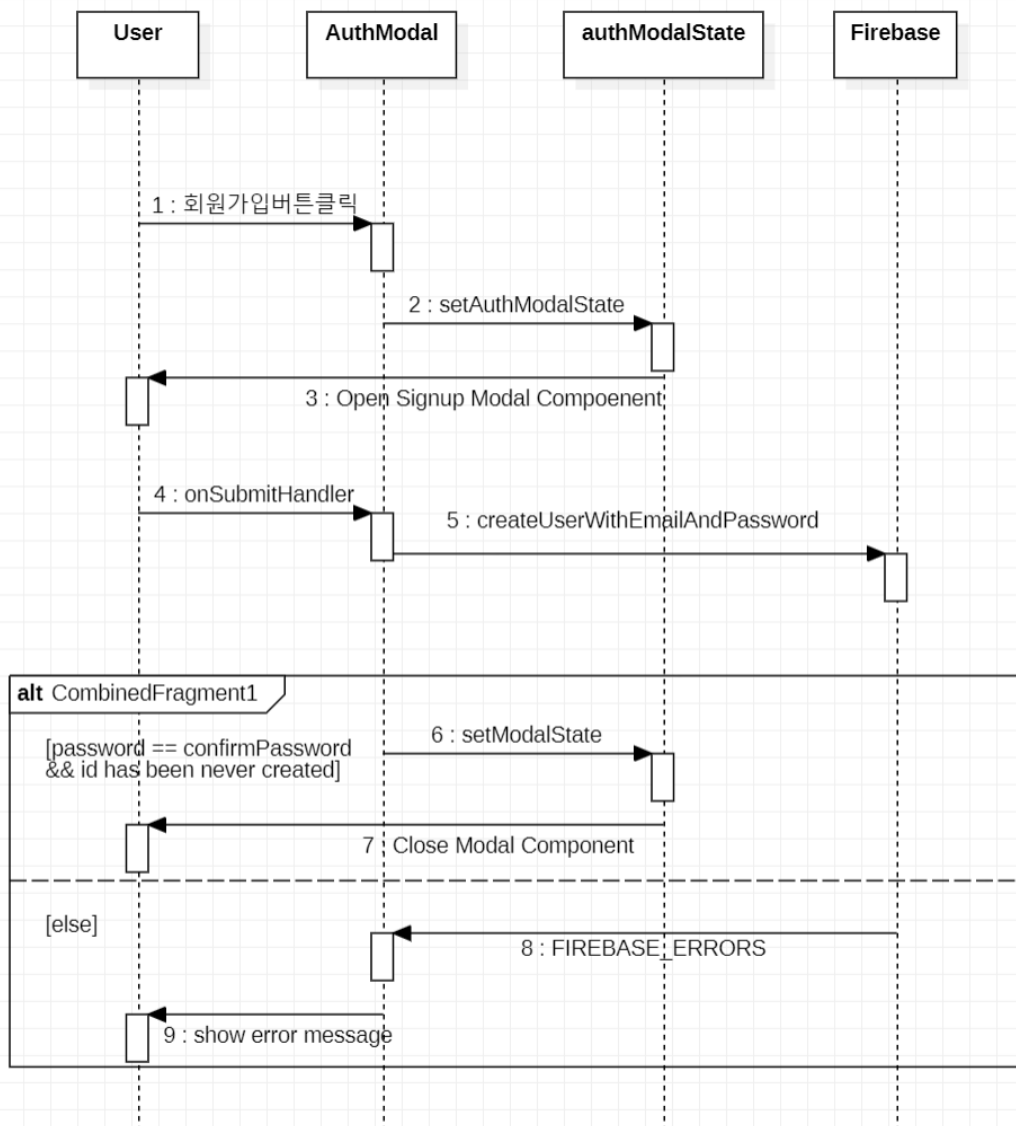
1) Login

생성된 커뮤니티의 정보를 담고 있는 인터페이스입니다.

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

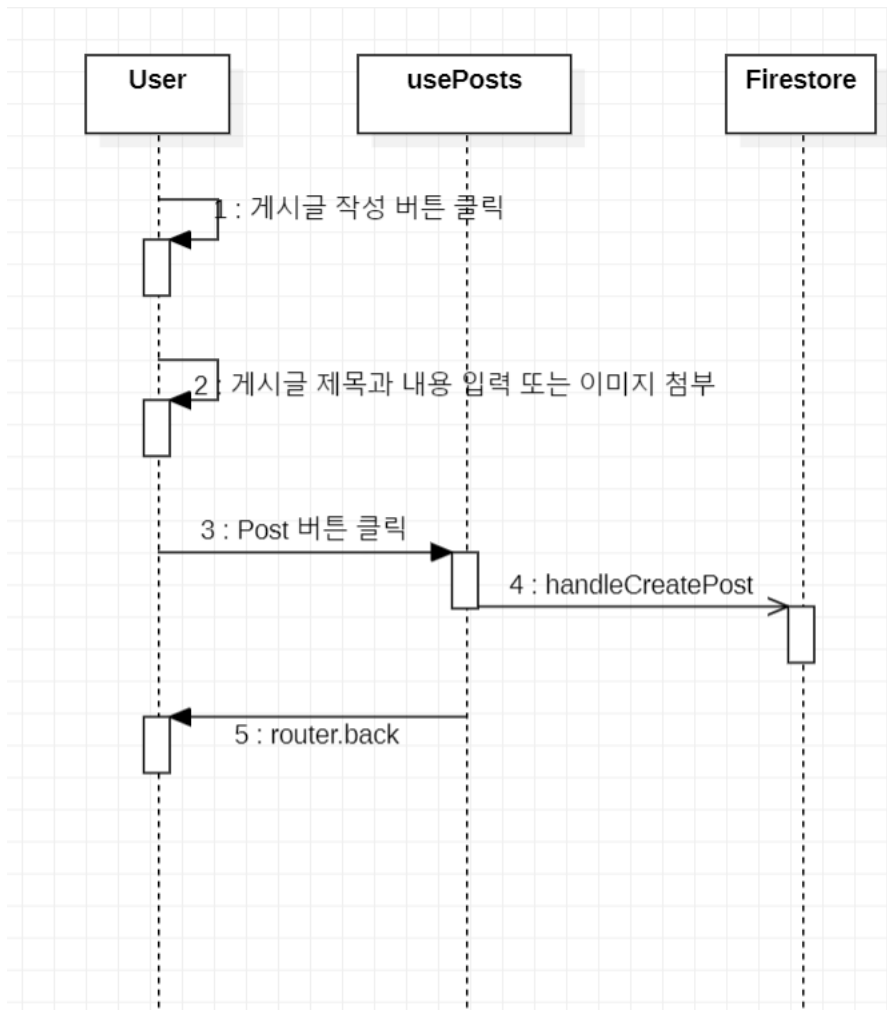
2) Logout





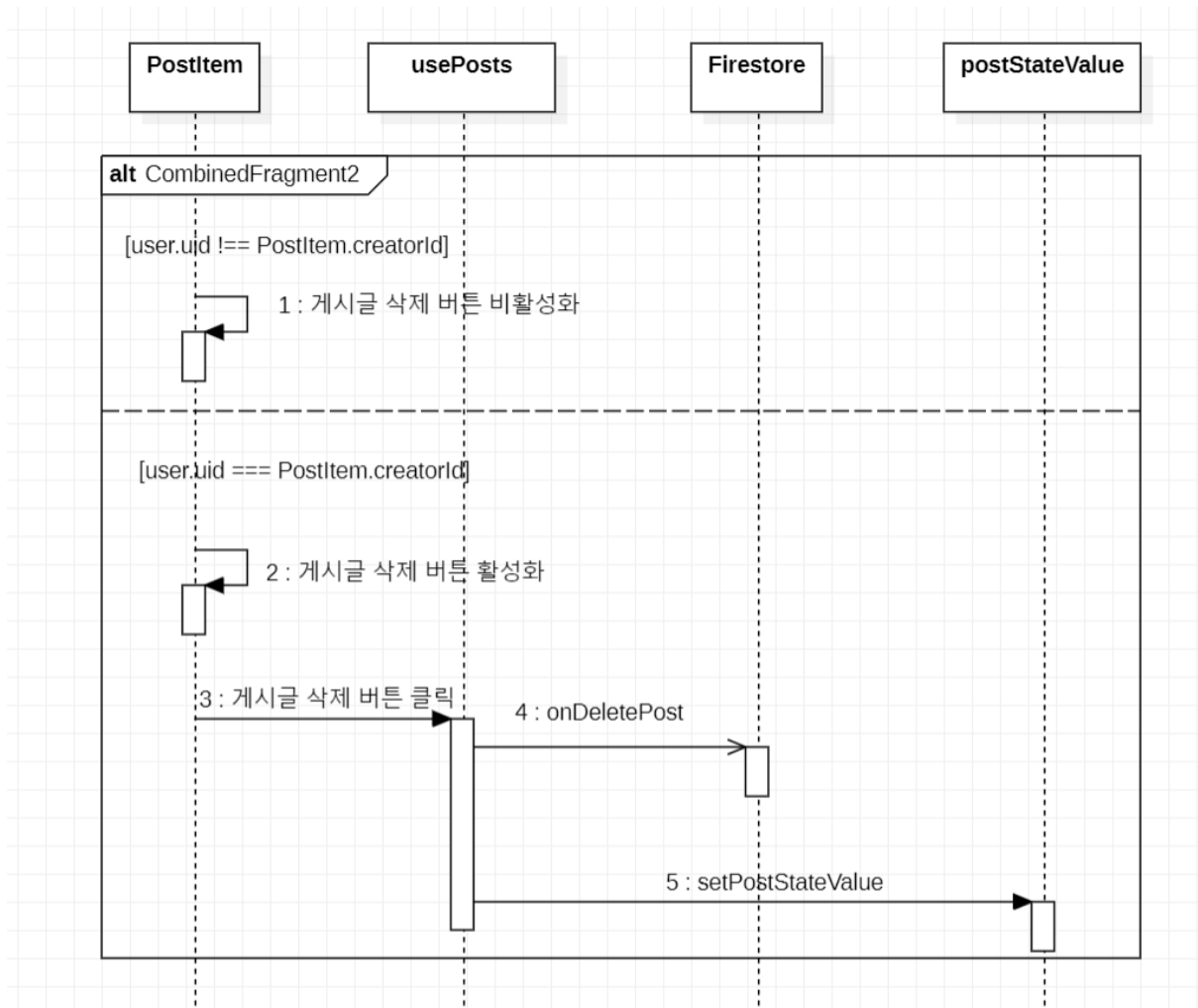
3) Signup

유저가 회원가입 버튼을 클릭한 뒤에 AuthModal 컴포넌트가 authModalState의 상태 속성 중에 view를 “login”으로 상태를 변경하고 유저가 회원가입 Modal에서 아이디와 비밀번호, 비밀번호 확인 입력을 하고 제출을 하게 되면 AuthModal은 Firebase로 유저가 입력한 입력 값을 전달하게 된다. 만약 해당하는 아이디가 존재하거나 비밀번호를 다르게 입력했을 경우 Firebase로부터 오류 메시지를 전달받게 되고, 그렇지 않다면 정상적으로 회원가입이 성공하고 AuthModal 컴포넌트는 authModalState의 상태 속성 중에 open 값을 false로 하여 해당 창을 닫는다.



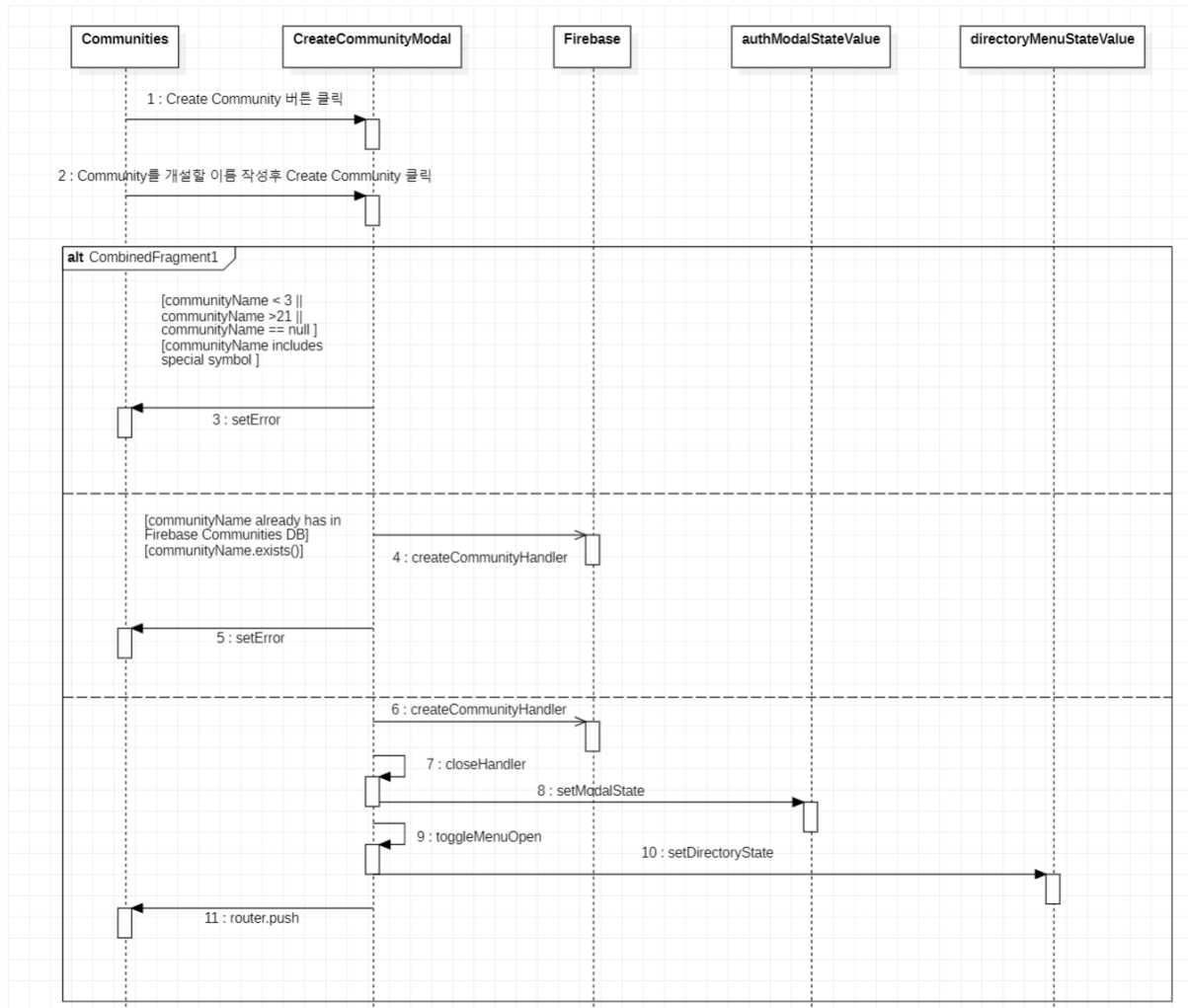
4) CreatePost

유저가 게시물 작성 버튼을 클릭하고 게시물 제목과 내용 또는 이미지를 첨부하고 Post 버튼을 클릭하여 게시물을 제출하게 되면 usePost 컴포넌트에서 handleCreatePost 비동기 함수를 통해 Firestore에 해당 게시물의 제목과 내용을 데이터를 저장하게 되고 usePost컴포넌트는 유저를 router.back 함수를 통해 이전화면으로 되돌아가도록 한다.



5) DeletePost

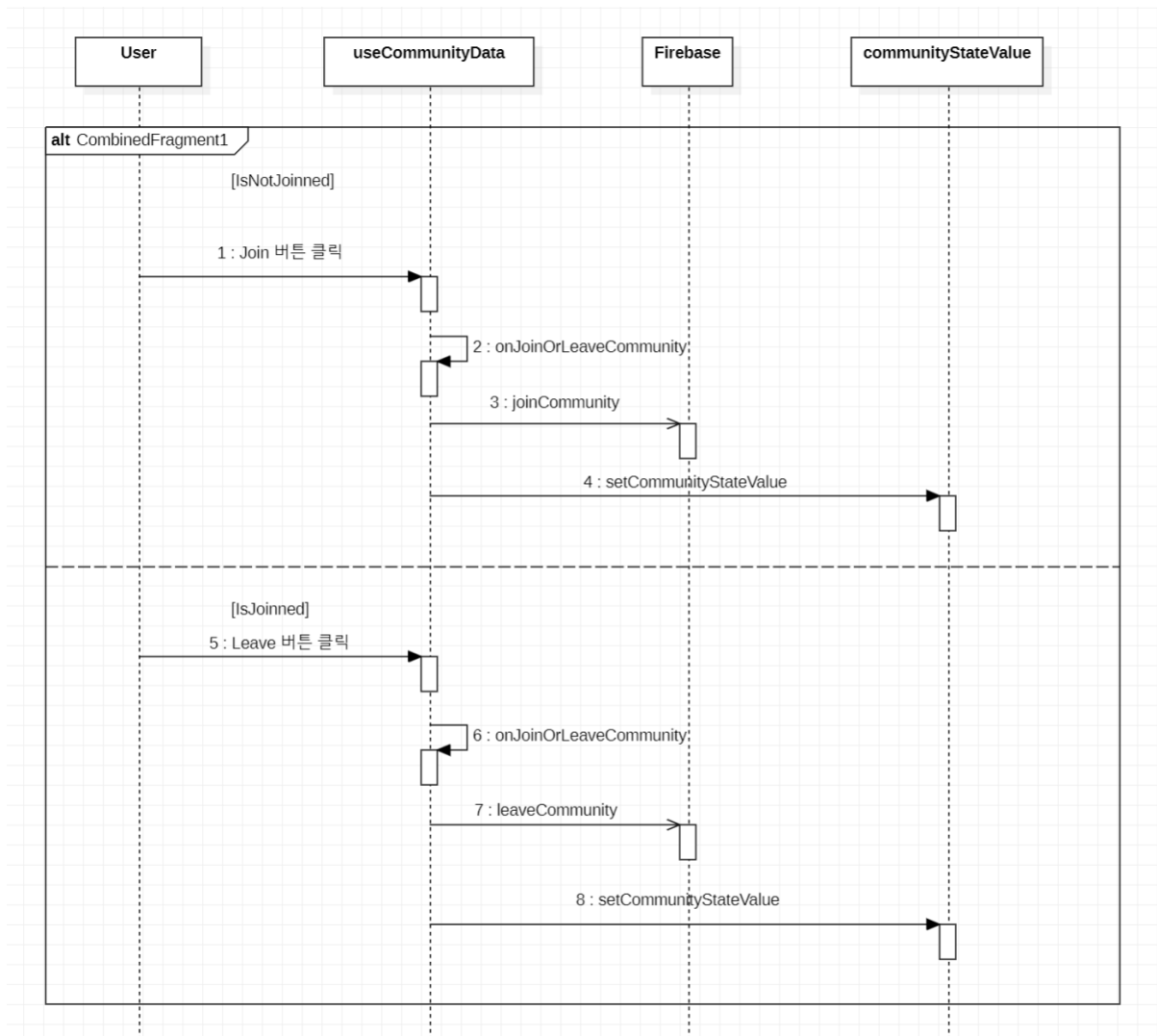
유저의 uid가 postStateValue의 posts의 creatorId 와 같다면 삭제버튼이 활성화가 되며 유저가 삭제 버튼을 클릭 시에 usePost 컴포넌트에서 onDeletePost 비동기 함수가 실행되고 해당 함수는 Firestore에서 해당되는 게시글의 id와 같은 게시글을 삭제합니다.



6) Create Community

사용자가 현재 접속해 있는 Directory를 나타내는 인터페이스입니다.

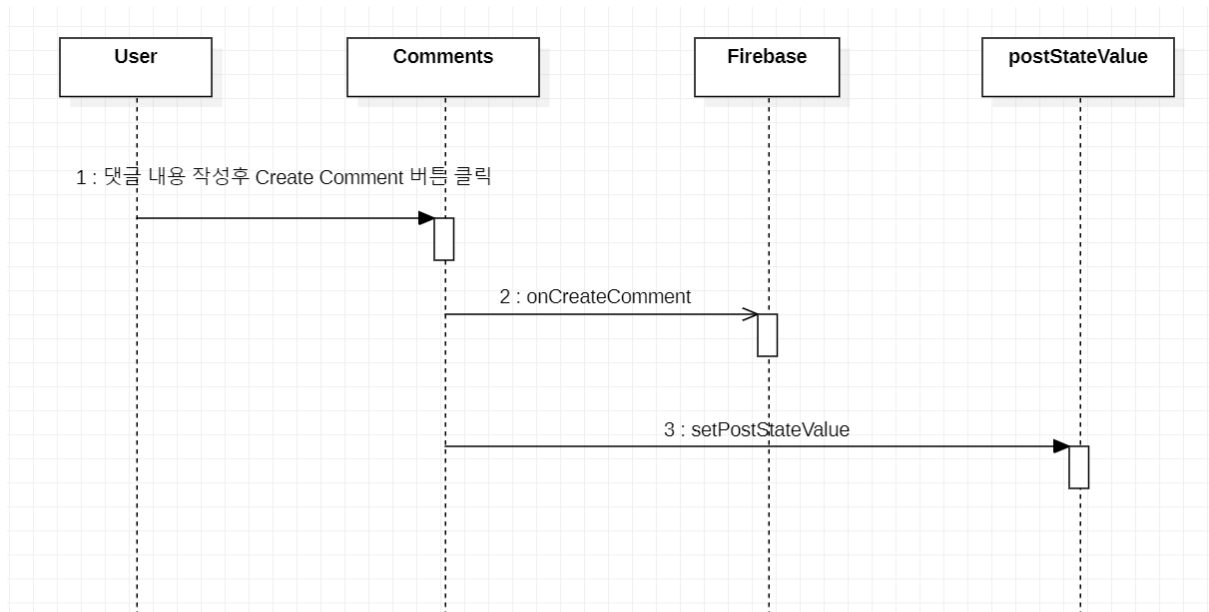
해당 디렉토리의 이름(커뮤니티 제목), 디렉토리의 주소(커뮤니티 주소), 아이콘(커뮤니티 로고) 정보를 담고 있습니다.



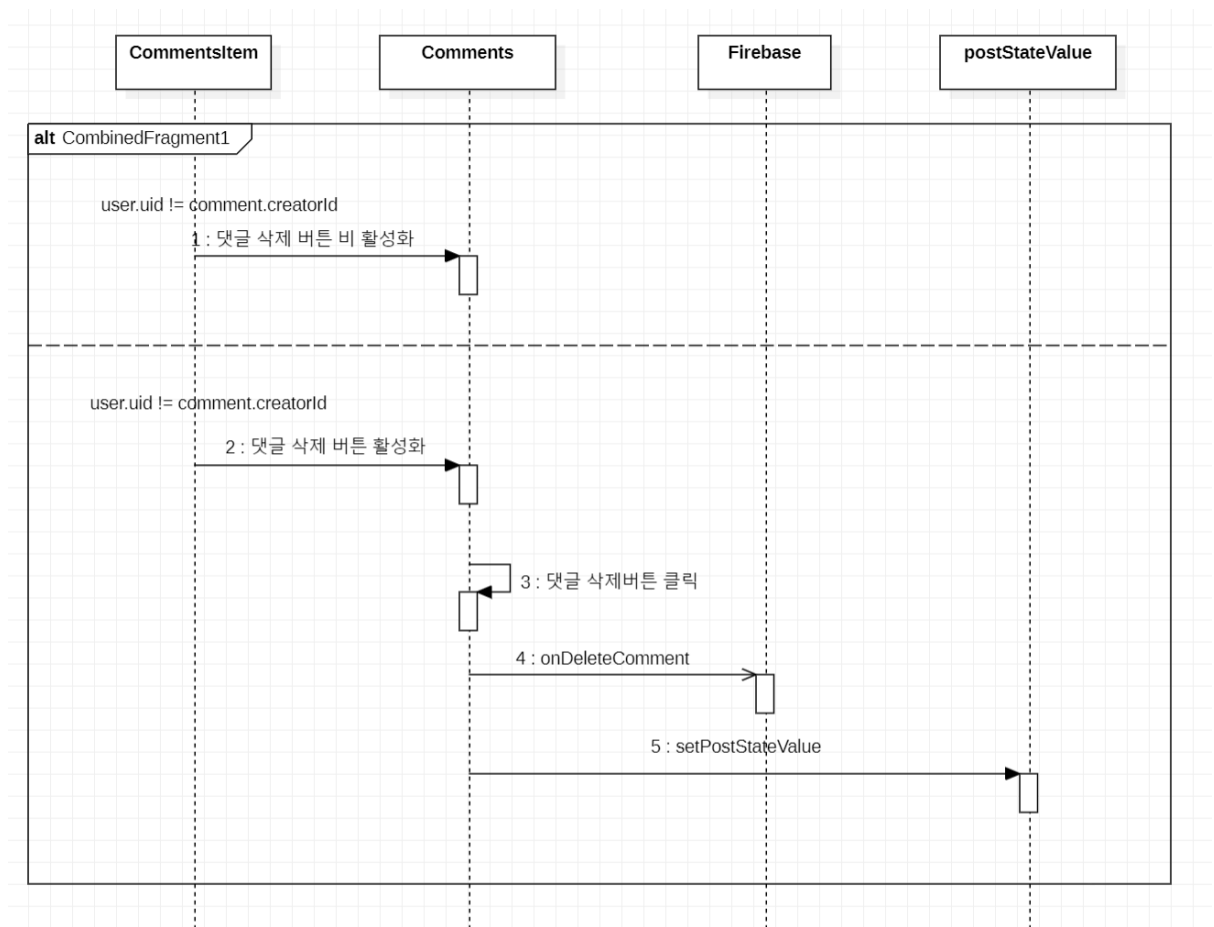
7) Join Community or Leave Community

유저 정보를 담고 있는 인터페이스입니다.

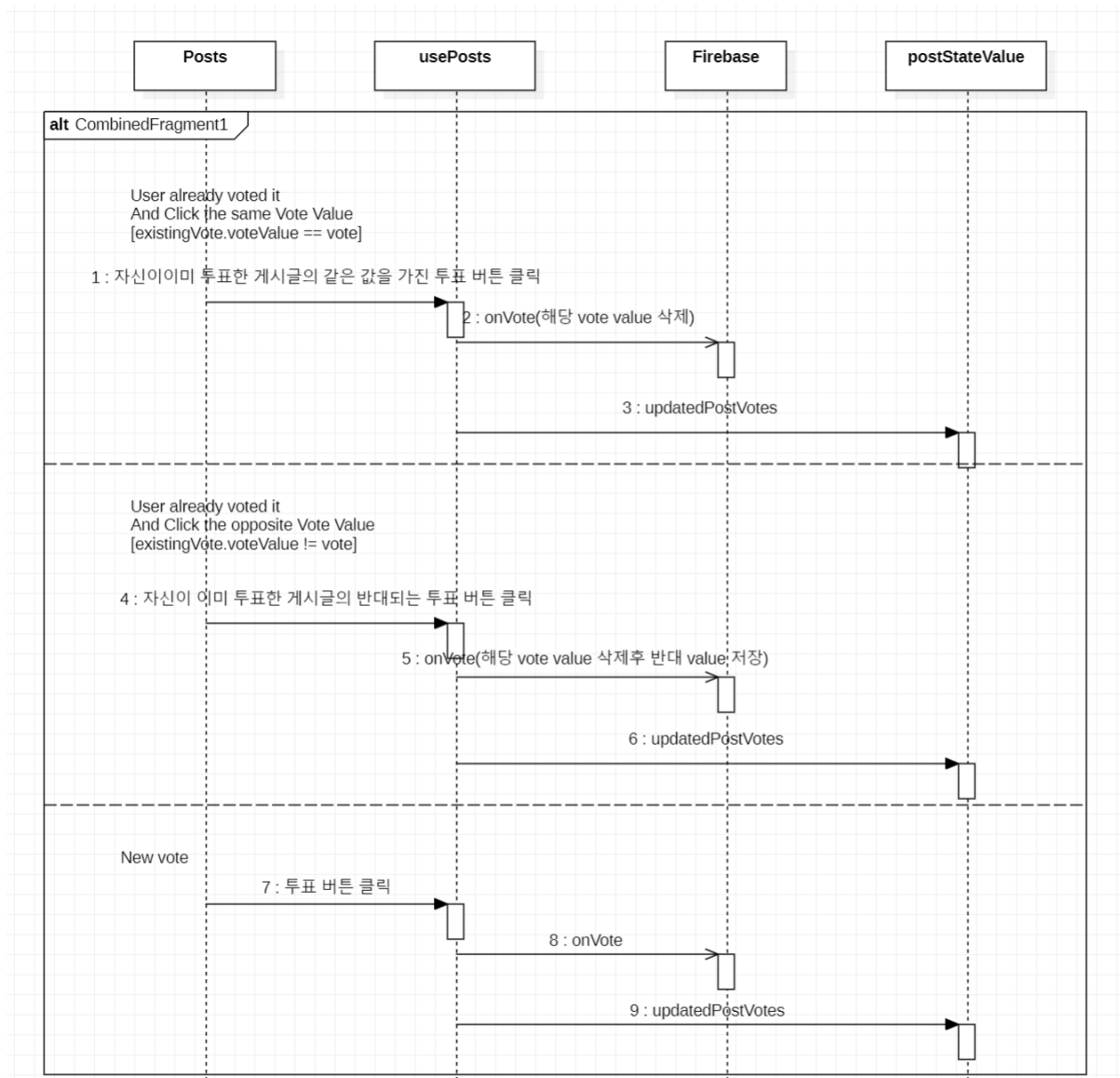
이메일, 닉네임 정보가 담겨 있습니다.



8) Create Comment



9) Delete Comment



10) Voting

4. State machine diagram

5. Implementation requirements

1) 하드웨어 요구사항

1. CPU : Intel Pentium 4 이상
2. RAM : 1GB 이상
3. HDD / SSD : 1GB 이상

2) 소프트웨어 요구사항

1. OS : Windows 7 이상

6. Glossary

7. References

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>