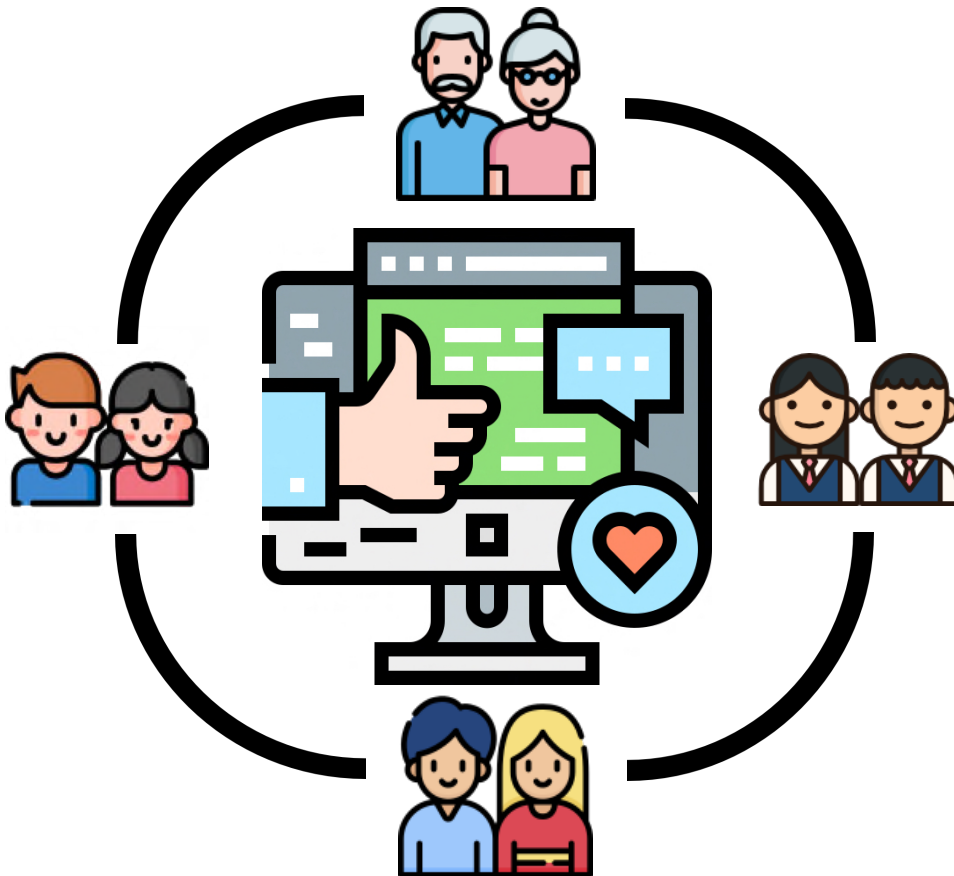


Sharing Favorite



Student No	21912058
Name	서영석
E-Mail	jrdtjr@naver.com

[Revision history]

Revision date	Version #	Description	Author
06/09/2023	1.0.0	first draft	서영석

= Contents =

1. Introduction	4
2. Class diagram	5
3. Sequence diagram	9
4. State machine diagram	19
5. Implementation requirements	20
6. Glossary	20
7. References	21

1. Introduction

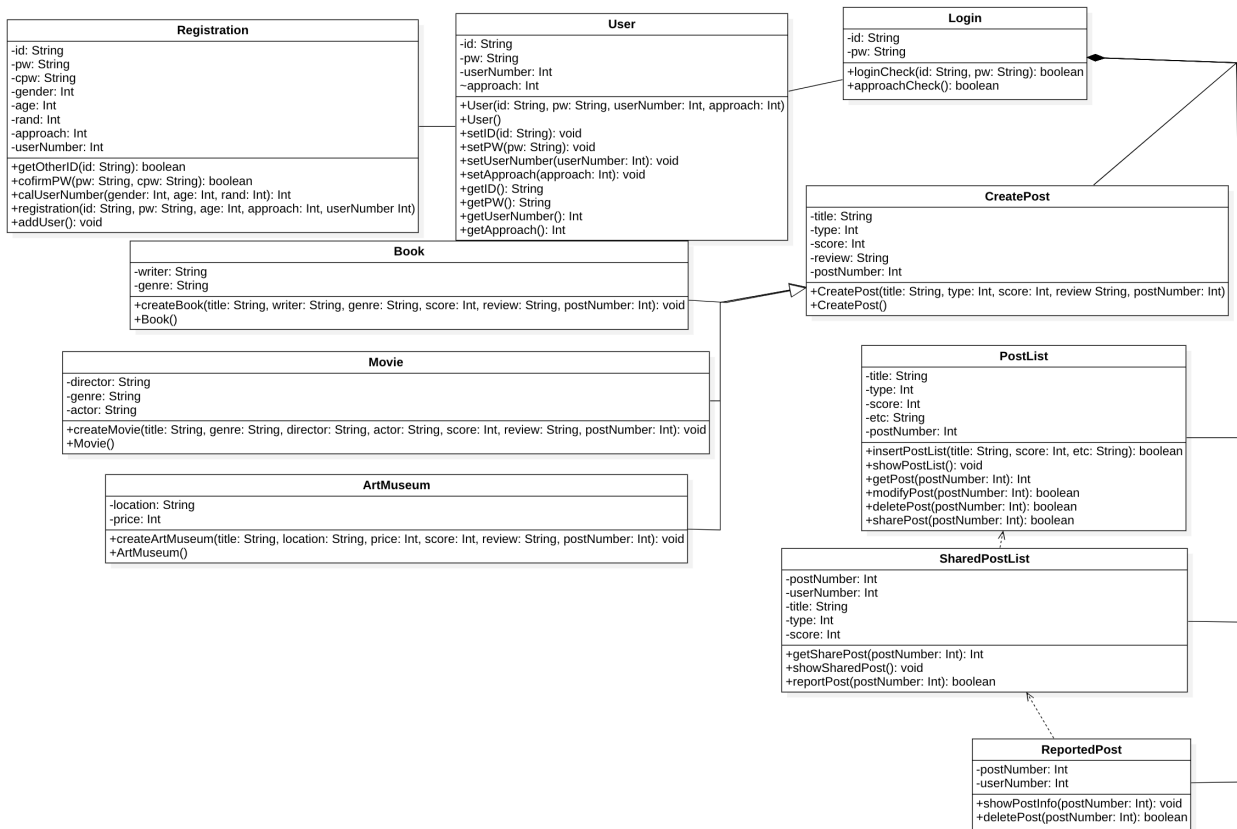
‘Sharing Favorite’는 본 프로젝트 작성자의 경험을 토대로 설계되었다. 2020년 코로나 시대로 인해, 밖에 나가는 일이 줄어들면서 집에 있는 시간이 길어졌다. 사람들과 만남을 자제하게 되면서 자연스럽게 인터넷 사용시간은 늘어나게 되었고, 인터넷으로 할 수 있는 다양한 활동들을 접하게 되었다. 기본적으로 넷플릭스, 티빙과 같은 OTT를 통해 수많은 콘텐츠를 시청하거나, eBook을 통해 책을 읽거나, 웹툰, 유튜브, 인터넷 방송 등 실내에서 혼자 다양한 콘텐츠를 소비하면서 시간을 보낼 수 있게 되었다. 이렇게 혼자서 할 수 있는 콘텐츠들이 많아짐에 따라 점점 사람들과 연락은 줄어들고 소통이 아닌 검색을 통해 콘텐츠에 대한 정보를 찾아보면서 점점 인간관계가 소홀해 짐을 느꼈고, 이를 개선할 방법으로 ‘Sharing Favorite’를 고안했다. ‘Sharing Favorite’는 각자 즐기고 있는 콘텐츠를 공유함으로써 소통의 계기를 만들어 줄 수 있다.

또한 코로나의 영향이 줄어들면서 대면 활동이 많아지게 되면서 그 전보다 더 많은 사람들이 외부 활동을 적극적으로 한다. 미술관, 공연, 축제, 전시회 등 문화 활동뿐만 아니라 축구, 클라이밍, 골프 등 다양한 스포츠 또한 유행하고 있다. 미술관, 공연과 같은 문화 활동의 후기를 작성하여 비슷한 취향을 가진 사람들과 공유하며 자신의 활동 범위를 확장할 수 있으며, 장소가 필요한 스포츠 활동에 대한 공유 정보를 토대로, 보다 좋은 질의 스포츠 활동을 할 수 있다.

이 어플리케이션은 회원가입과 로그인을 통해 이용자와 포스트(후기) 공유를 통해 보다 보기 쉽게 다양한 콘텐츠를 접할 수 있으며, 또한 부적절한 콘텐츠 신고하기를 통해 삭제할 수 있다.

현재 문서는 Analysis의 다음 단계인 Design의 내용이며, 이 보고서에서는 Class Diagram, Sequence Diagram, State machine Diagram을 이용하여 해당 시스템의 전반적인 기능 구성 및 동작에 대한 모습을 시각적으로 보여준다.

2. Class diagram



1) Registration

(1) Attributes

- id:String : 사용자 아이디
- pw:String : 사용자 비밀번호
- cpw:String : 입력한 비밀번호 확인
- gender:Int : 사용자 성별
- age:Int : 사용자 나이
- rand:Int : 사용자 고유번호 뒷자리의 랜덤숫자
- approach:Int : 사용자인지 관리자인지 확인하는 숫자
- userNumber:Int : 사용자의 고유번호

(2) Methods

- +getOtherID(id:String):boolean : 사용자가 입력한 아이디가 사용할 수 있는 아이디 인지 확인
- +confirmPW(pw:String, cpw:String):boolean : 사용자가 입력한 비밀번호와 확인 비밀번호가 동일한지 확인
- +calUserNumber(gender:Int, age:Int, rand:Int):Int : 사용자의 고유번호를 계산해주

는 함수

+registration(id:String, pw:String, age:Int, approach:Int, userNumber:Int) : 회원가입
생성자

+addUser() : 유저 추가

2) Login

(1) Attributes

-id:String : 로그인 아이디 입력

-pw:String : 로그인 비밀번호 입력

(2) Methods

+loginCheck(id:String, pw:String):boolean : 아이디 비밀번호를 확인

+approachCheck():boolean : 사용자의 접근권한을 확인

3) User

(1) Attributes

-id:String : 사용자의 아이디

-pw:String : 사용자의 비밀번호

-userNumber:Int : 사용자의 고유번호

~approach:Int : 사용자의 접근권한

(2) Methods

+User(id:String, pw:String, userNumber:Int, approach:Int) : 생성자

+User() : 생성자

+setID(id:String):void : 아이디 설정

+setPW(pw:String):void : 비밀번호 설정

+setUserNumber(userNumber:Int):void : 유저 고유번호 설정

+setApproach(approach:Int):void : 접근권한 설정

+getID():String : 아이디를 얻음

+getPW():String : 비밀번호를 얻음

+getUserNumber():Int : 고유번호를 얻음

+getApproach():Int : 접근권한을 얻음

4) CreatePost

(1) Attributes

-title:String : 포스트 제목

-type:Int : 포스트 타입

-score:Int : 포스트 점수

- review:String : 포스트 후기작성
- postNumber : 포스트 고유번호

(2) Methods

- +CreatePost(title:String, type:Int, score:Int, review:String, postNumber:Int) : 생성자
- +CreatePost() : 생성자

5) Book

(1) Attributes

- writer:String : 책 작가의 이름
- genre:String : 장르

(2) Methods

- +createBook(title:String, writer:String, genre:String, score:Int, review:String, postNumber:Int):void : 포스트 작성하기
- +Book(): 생성자

6) Movie

(1) Attributes

- director:String : 영화 감독의 이름
- genre:String : 장르
- actor:String : 주연, 조연 배우의 이름

(2) Methods

- +createMovie(title:String, genre:String, director:String, actor:String, score:Int, review:String, postNumber:Int):void : 포스트 작성하기
- +Movie() : 생성자

7) ArtMuseum

(1) Attributes

- location:String : 장소, 위치 정보
- price:Int : 관람비

(2) Methods

- +createArtMuseum(title:String, location:String, price:Int, score:Int, review:String, postNumber:Int):void : 포스트 작성하기
- +ArtMuseum() : 생성자

8) PostList

(1) Attributes

- title:String : 포스트 제목
- type:Int : 포스트 타입
- score:Int : 포스트 점수
- etc:String : review의 일부
- postNumber:Int : 포스트의 고유번호

(2) Methods

- +insertPostList(title:String, score:Int, etc:String):boolean : 사용자가 작성한 포스트를 한줄로 간략하게 만들어서 저장한다.
- +showPostList():void : 리스트 출력
- +getPost(postNumber:Int):Int : 리스트에서 포스트를 클릭하면 해당 포스트를 자세히 볼 수 있다.
- +modifyPost(postNumber:Int):boolean : 포스트 수정하기
- +deletePost(postNumber:Int):boolean : 포스트 삭제하기
- +sharePost(postNumber:Int):boolean : 포스트 공유하기

9) SharedPostList

(1) Attrttributes

- postNumber:Int : 포스트 고유번호
- userNumber:Int : 유저 고유번호
- title:String : 포스트 제목
- type:Int : 포스트 타입
- score:Int : 포스트 점수

(2) Methods

- +getSharePost(postNumber:Int):Int : 사용자가 공유한 포스트에 대한 고유번호를 받는다.
- +showSharedPost():void : 사용자들이 공유한 포스트들에 대한 정보를 출력한다.
- +reportPost(postNumber:Int):boolean : 부적절한 포스트를 관리자에게 신고한다.

10) ReportedPost

(1) Attrttributes

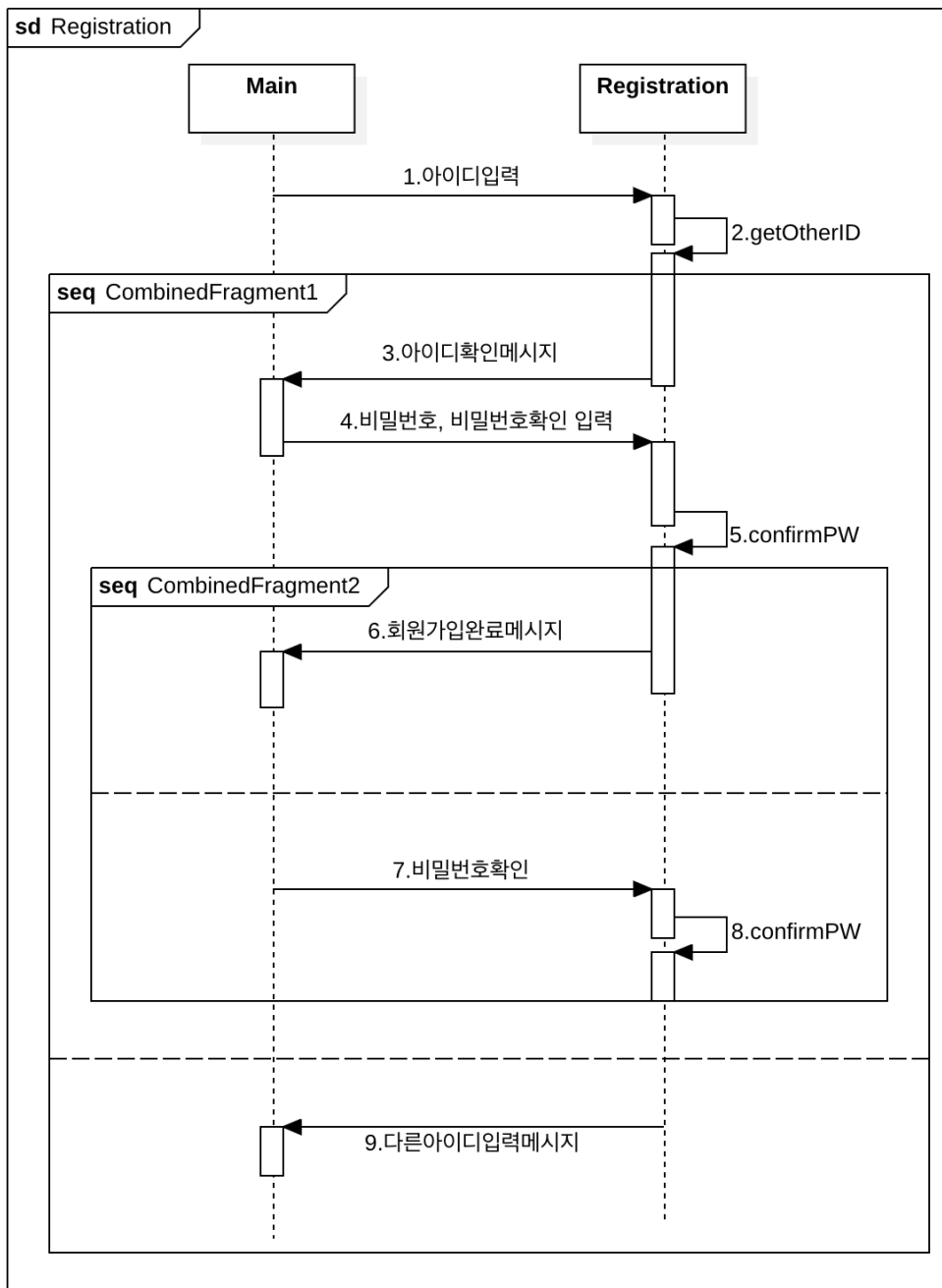
- postNumber: Int : 신고받은 포스트 고유번호
- userNumber: Int : 해당 포스트를 작성한 유저의 고유번호

(2) Methods

- +showPostInfo(postNumber: Int):void : 해당 포스트에 대한 정보를 출력한다.
- +deletePost(postNumber:Int):boolean : 포스트를 삭제한다.

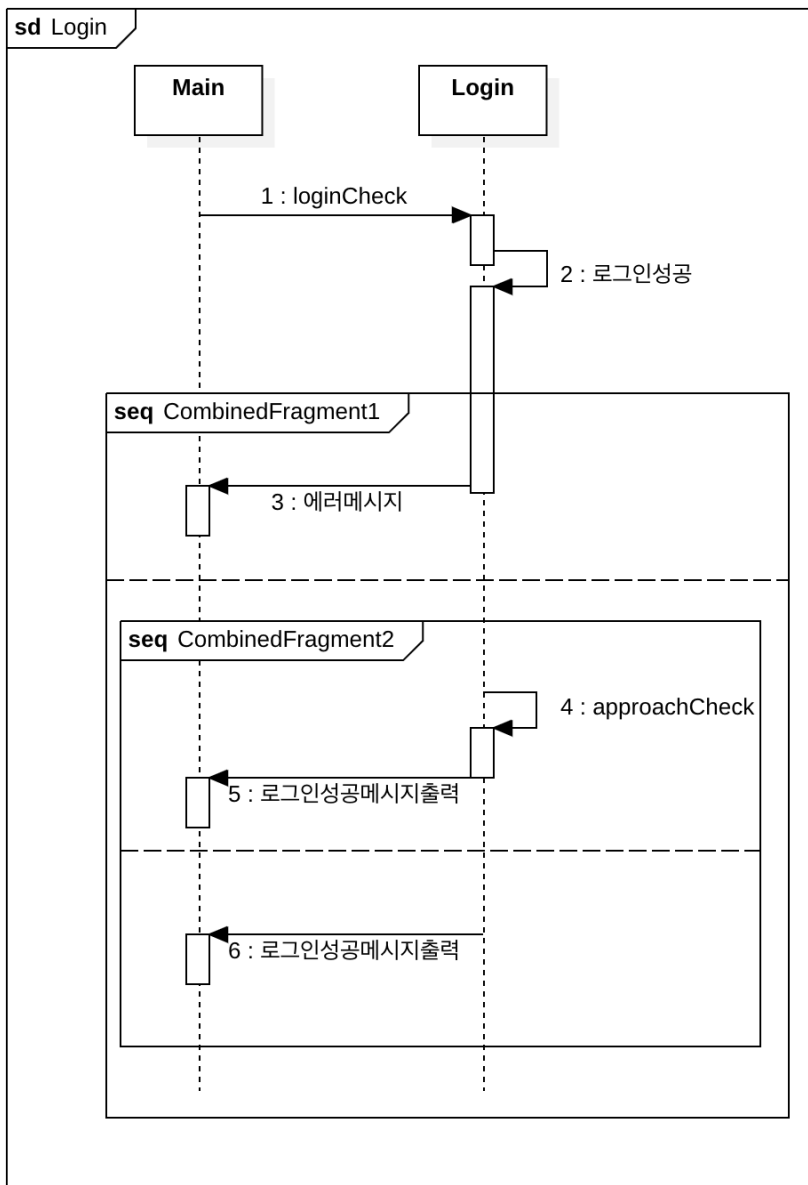
3. Sequence diagram

1) 회원가입



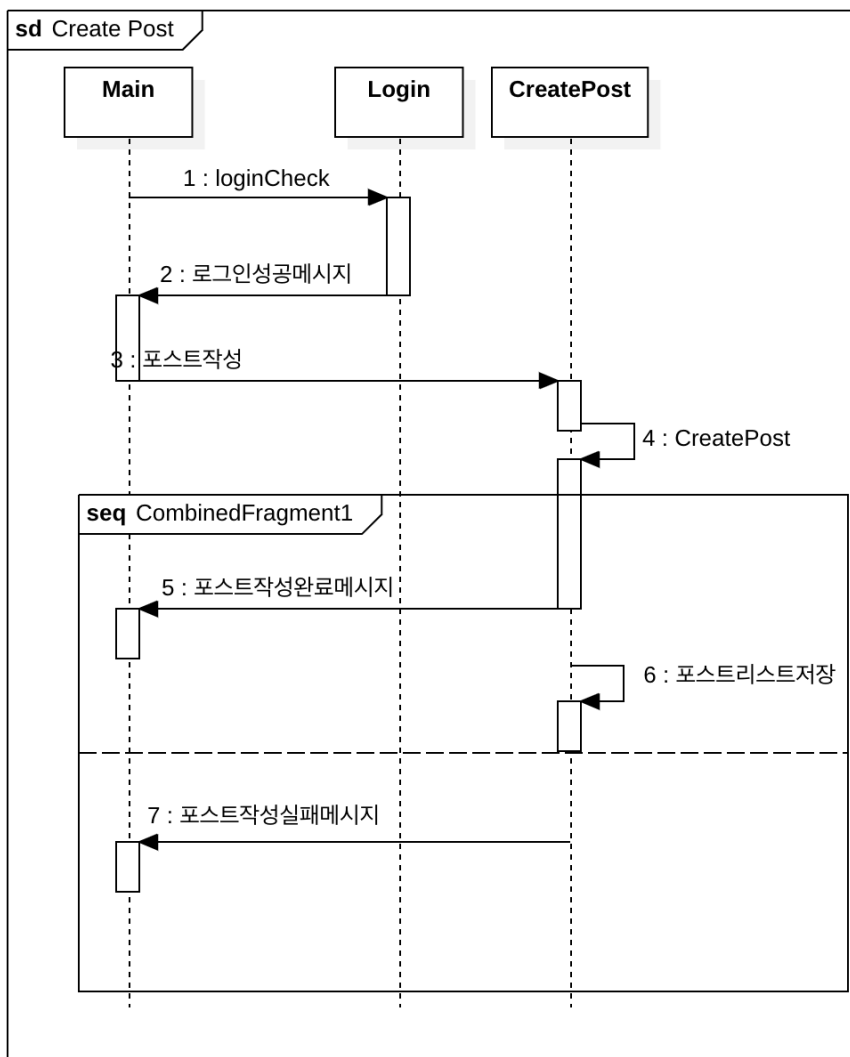
메인화면에서 회원가입을 위해 회원가입 페이지에 들어가서 회원가입을 한다. 회원가입 시 아이디를 입력했을 때 중복 아이디가 없을 시 아이디 확인 메시지를 출력한다. 비밀번호와 해당 비밀번호와 똑같은 비밀번호를 입력 후, 두 비밀번호가 일치하는지 확인하고 일치한다면 회원가입 완료메시지를 보낸다. 아이디 확인 시 이미 존재하는 아이디라면 다른 아이디를 입력하라는 메시지를 출력하고, 비밀번호와 비밀번호 확인의 비밀번호가 다르다면 비밀번호를 확인하라는 메시지를 출력한다.

2) 로그인



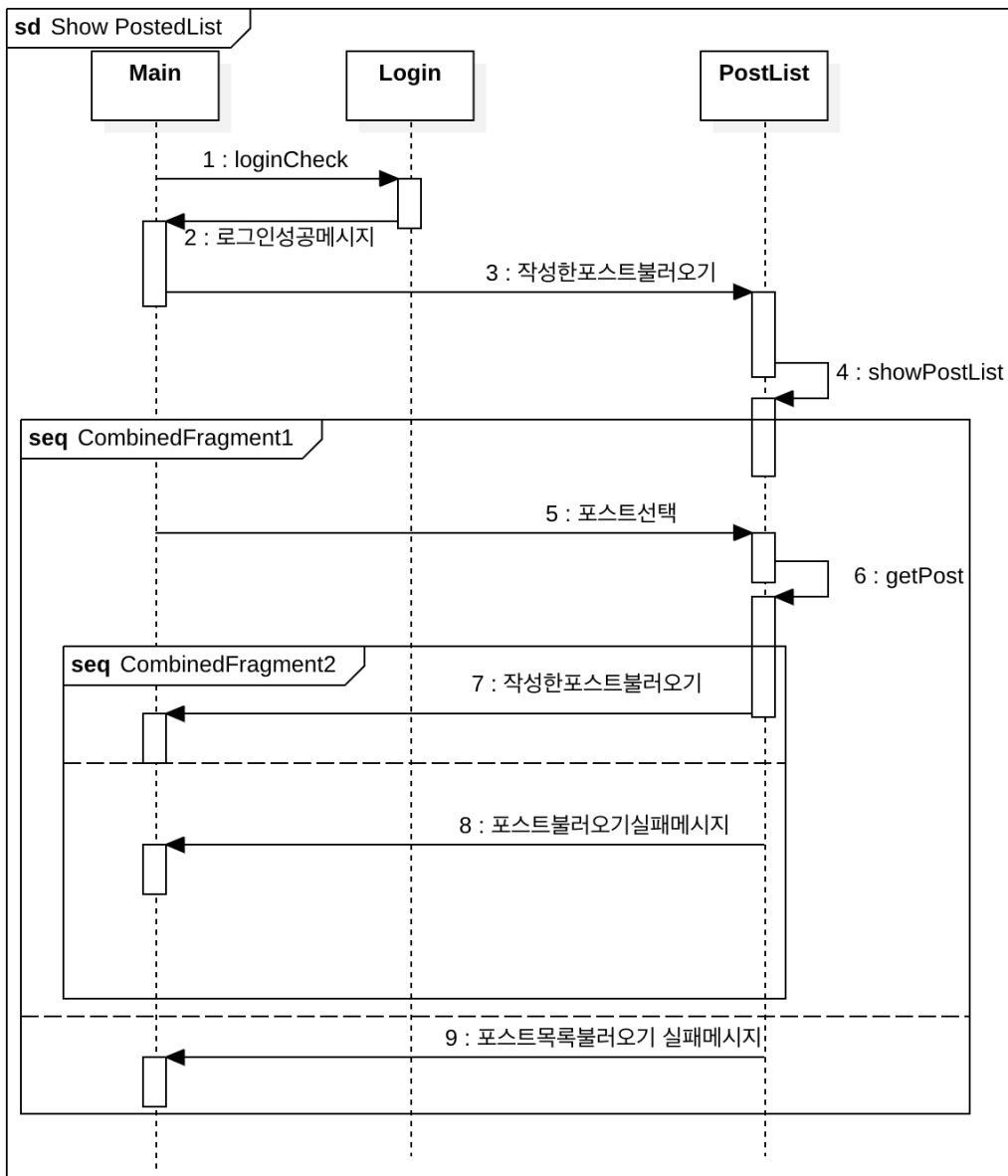
어플리케이션 접속 후 로그인을 하기 위해서 아이디 비밀번호를 입력하고 아이디에 해당하는 비밀번호를 확인 후 일치한다면 해당 아이디의 접근권한을 확인 후 해당 기능을 수행할 수 있는 권한을 가진 아이디로 로그인을 한다. 로그인 실패 시 에러메시지를 출력한다.

3) 포스트 작성



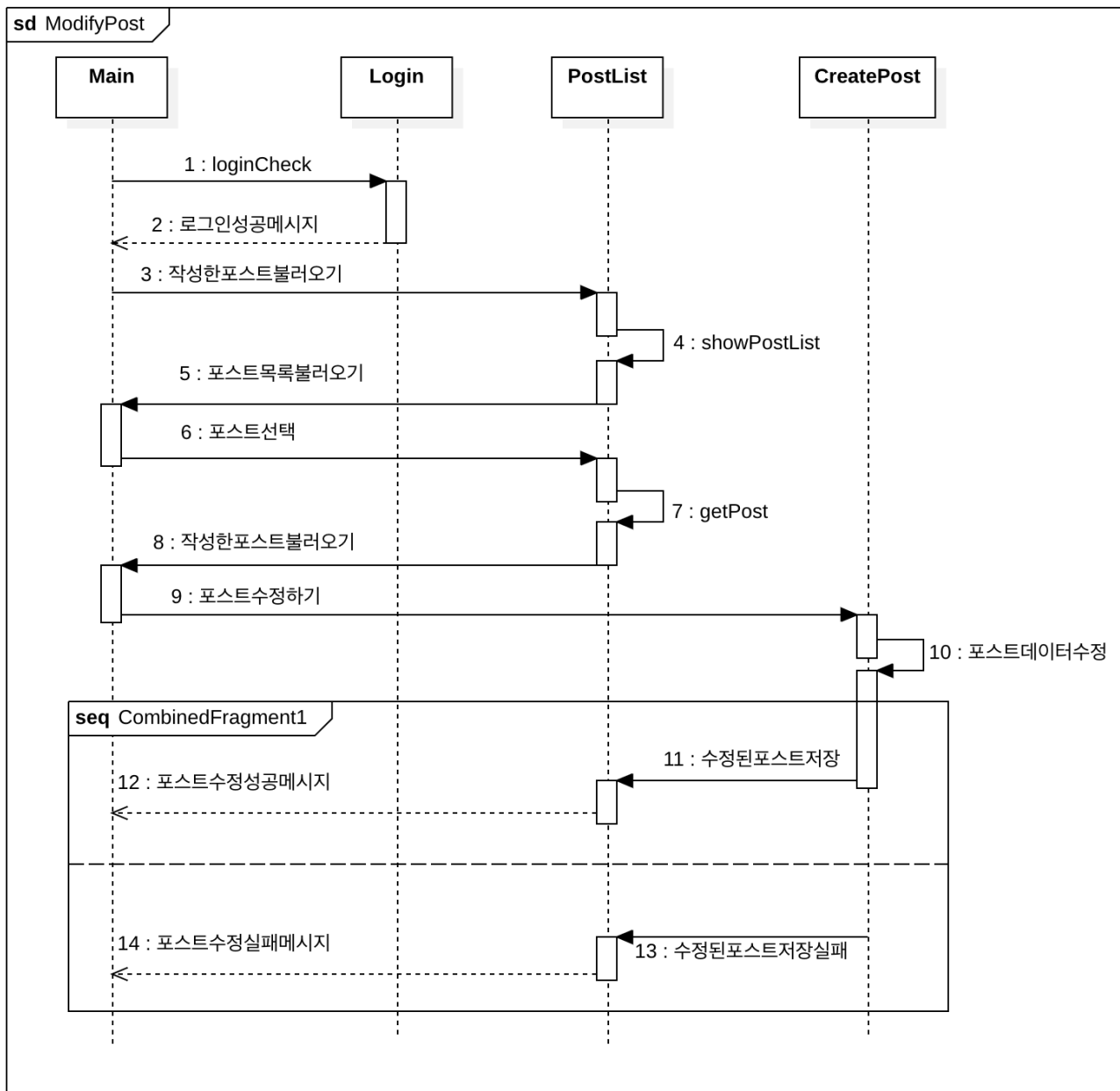
포스트 작성을 위해서 로그인을 완료한 뒤, 포스트를 작성하기 위해 포스트 작성 클래스를 실행한다. 해당 포스트에 해당하는 양식을 불러온 뒤 양식에 맞는 데이터를 입력 후 포스트 작성을 완료하면, 포스트가 제대로 작성되었다면 포스트 작성 완료 메시지를 출력 후, 사용자의 포스트 리스트에 작성한 포스트를 저장한다. 포스트 작성 실패 시 사용자에게 포스트 작성에 실패 했다는 메시지를 출력한다.

4) 작성한 포스트 리스트 목록



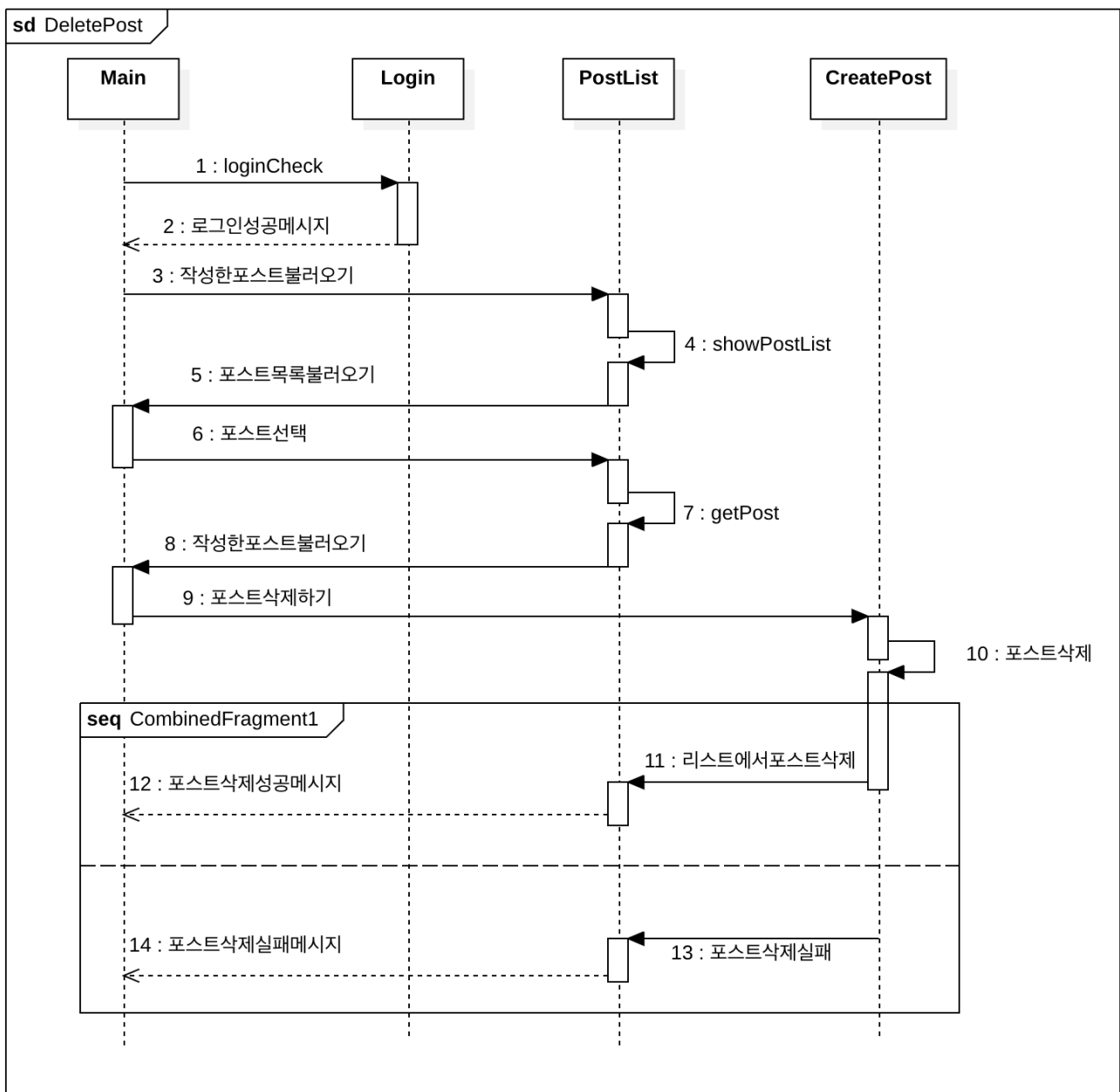
사용자가 자신이 작성한 포스트 리스트를 보기위해 로그인을 하고, 작성한 포스트를 불러온다. 작성한 포스트 리스트를 문제없이 불러왔다면, 사용자에게 리스트를 보여준다. 리스트 불러오기를 성공한다면 사용자가 해당 포스트를 자세히 보기위해 해당 포스트를 클릭하면 클래스에서 포스트에 대한 자세한 정보를 불러온다. 불러오기에 성공한다면 사용자에게 포스트를 보여준다. 포스트 목록 및 포스트 불러오기를 실패했다면 해당 기능을 수행하지 못했다는 실패 메시지를 출력한다.

5) 포스트 수정



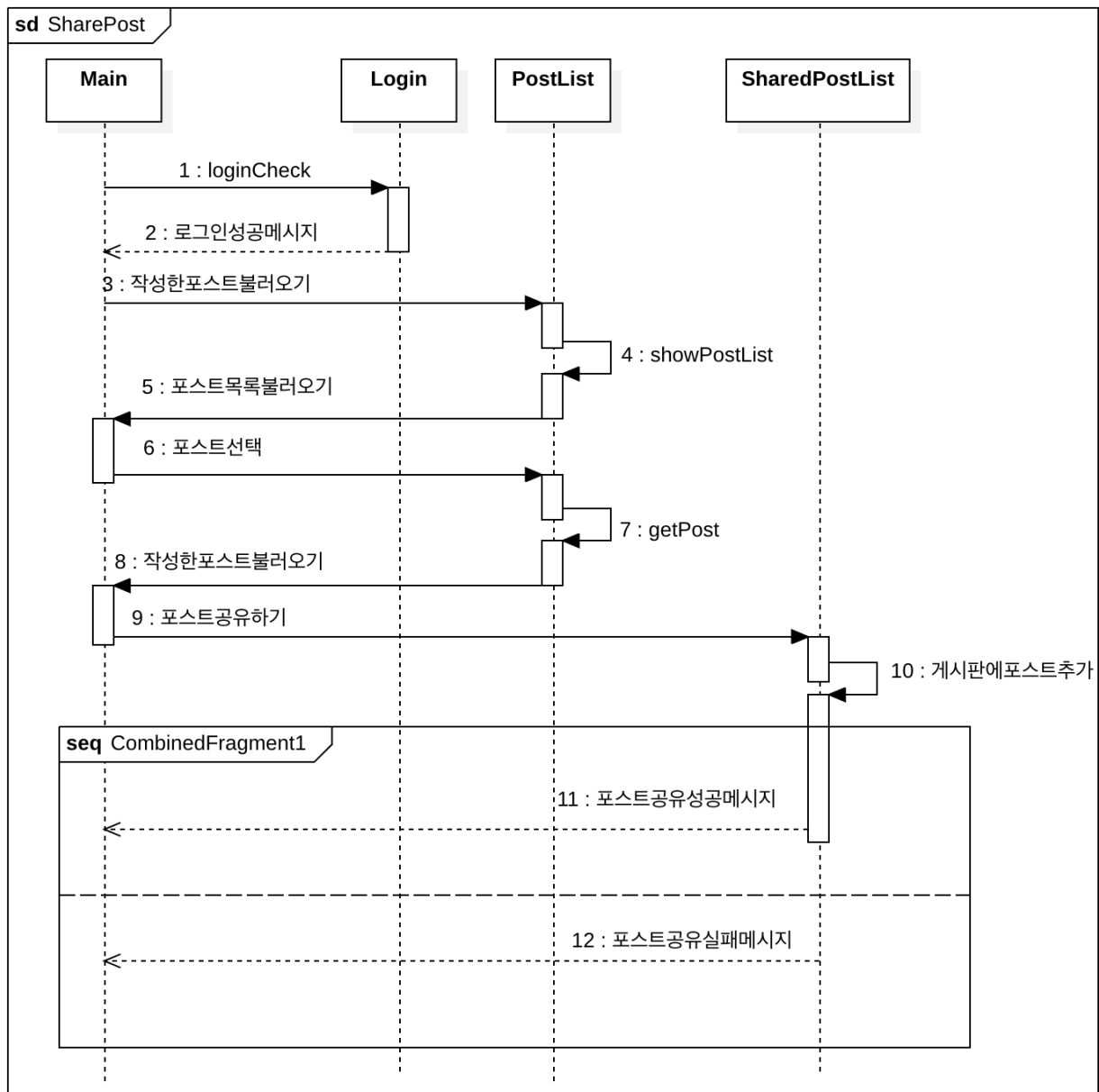
포스트 수정을 하기위해서 사용자는 로그인 후 자신이 작성한 포스트 리스트 중 한 가지 포스트를 선택한다. 포스트를 수정하기위해 수정 버튼을 클릭하여 포스터를 수정 할 수 있는 상태로 변경한다. 사용자의 요구에 맞게 해당 데이터를 수정 후 수정 한 결과를 새로 저장하고, 수정된 포스트를 사용자의 포스트 리스트 목록에 반영한다. 포스트 수정에 성공했다면 성공 메시지를 출력하고, 수정된 포스트 저장에 실패했다면, 실패메시지를 출력한다.

6) 포스트 삭제



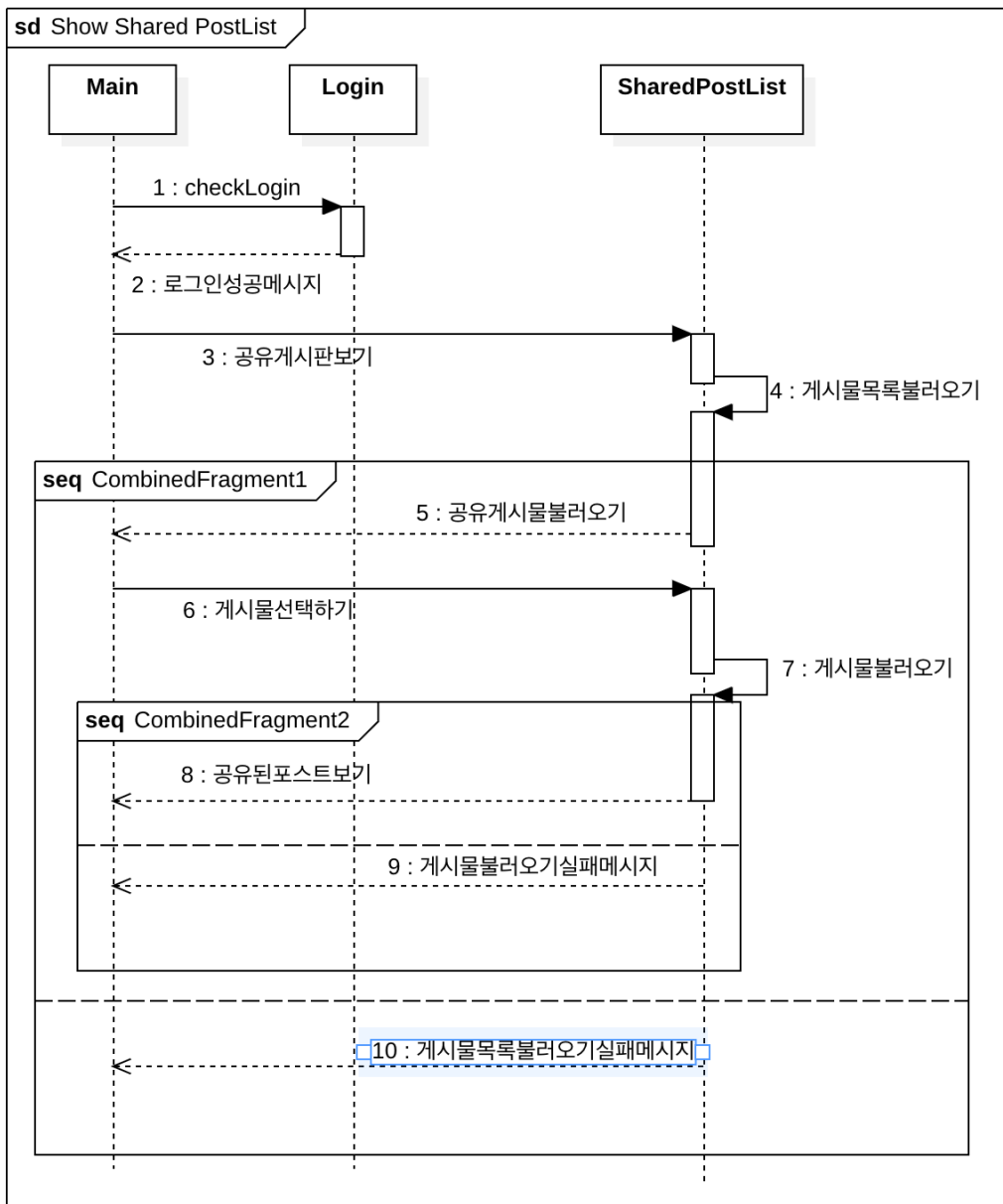
포스트를 삭제하기 위해 사용자는 로그인 후, 자신이 작성한 포스트 리스트에서 해당 포스트를 선택한다. 포스트 선택 후 삭제버튼을 누르면 사용자가 작성한 리스트에서 해당 리스트를 삭제한다. 삭제에 실패했다면 포스트 삭제 실패 메시지를 출력한다.

7) 포스트 공유



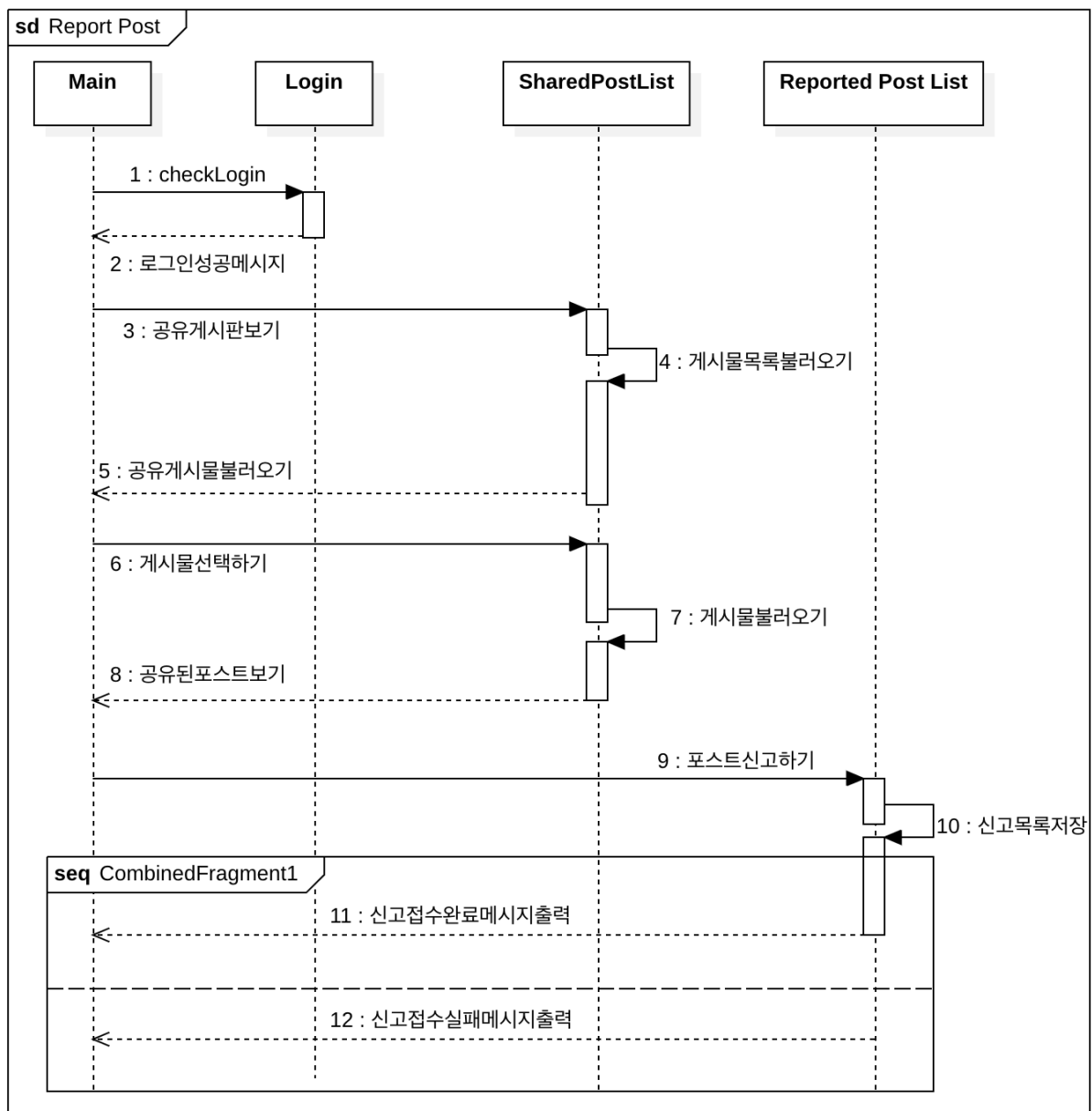
포스트를 공유하기 위해서 사용자가 로그인 후, 사용자의 포스트 리스트에서 해당 포스트를 선택한다. 해당 포스트 불러오기를 성공한 후 공유하기 버튼을 누르면, 해당 포스트가 공유 게시판 목록으로 업로드된다. 업로드에 성공하면 포스트 공유를 성공했다는 메시지를 출력하고, 포스트 공유에 실패했다면 실패 메시지를 출력한다.

8) 공유된 포스트 리스트 목록



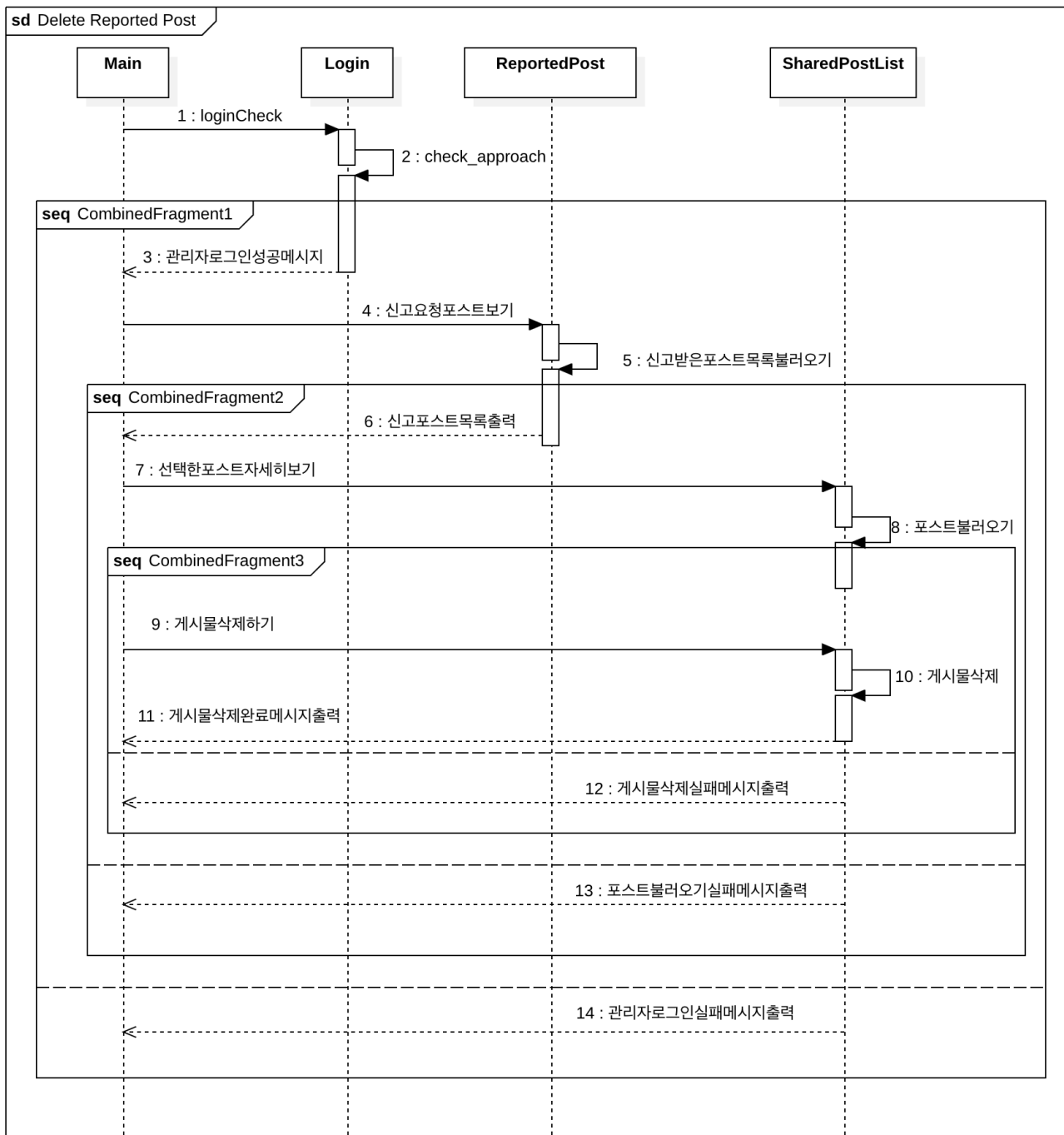
다른 사용자들이 공유한 포스트 리스트를 보기 위해 사용자는 로그인 후, 공유 게시판 기능을 수행한다. 사용자가 공유 게시판의 게시물들에 대한 정보를 확인 후 해당 포스트를 자세히 보고 싶다면 포스트 클릭을 통해 게시물을 불러온다. 포스트 불러오기를 성공한다면 공유된 포스트를 볼 수 있고, 각각 기능에 대한 불러오기를 실패한다면 실패 메시지를 출력한다.

9) 포스트 신고



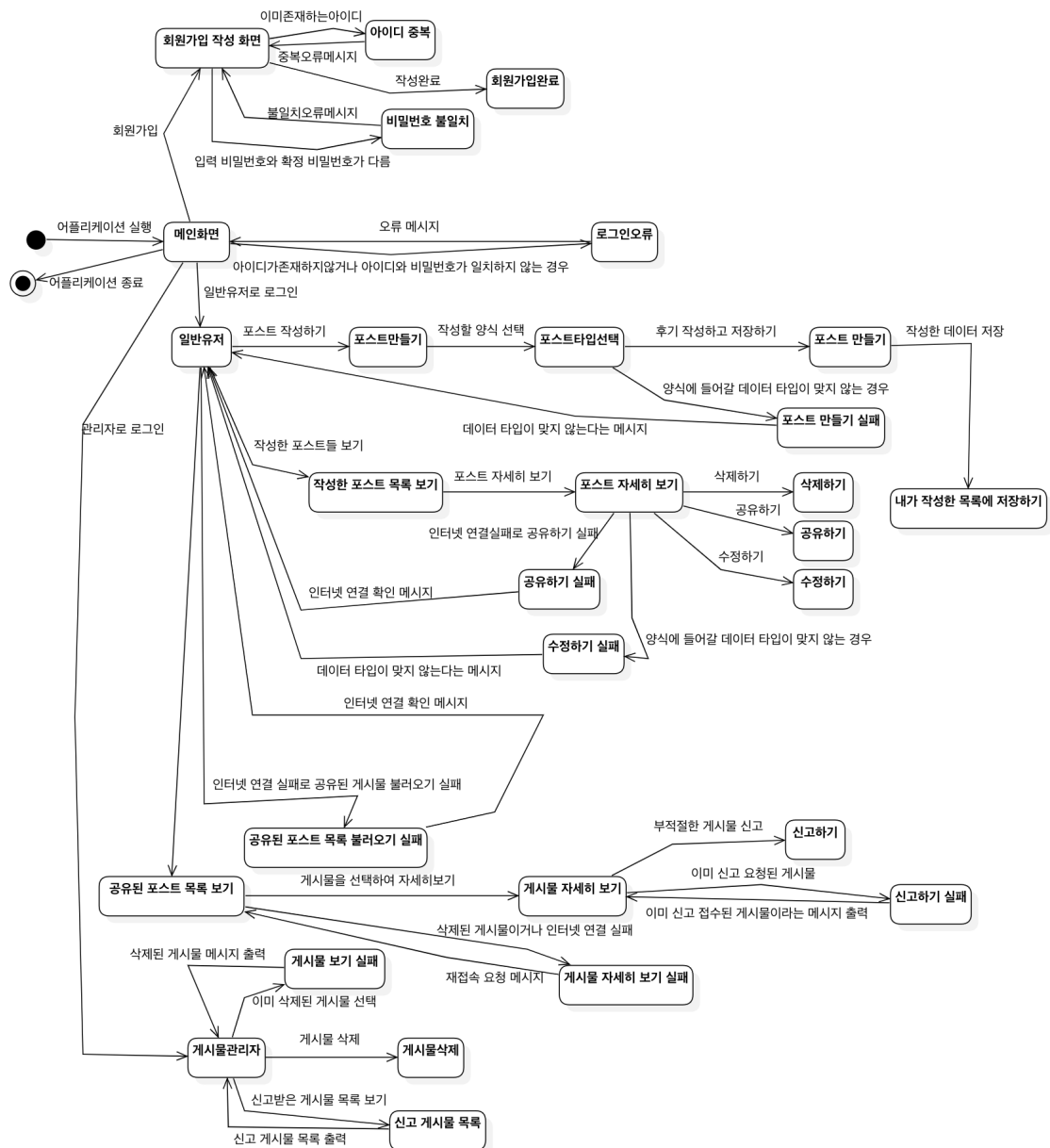
사용자가 공유 게시판을 보던 중 해당 포스트가 부적절하다고 생각하는 포스트를 발견했을 때 포스트를 신고한다. 포스트 신고하기 기능을 수행하면 해당 게시물에 대한 정보를 관리자의 신고 목록에 저장한다. 신고 목록 저장에 성공한다면 신고 접수 완료 메시지를 출력하고, 신고 접수에 실패한다면 실패 메시지를 출력한다.

10) 신고요청 받은 포스트 삭제하기



관리자가 신고받은 포스트를 보기위해 로그인을 한다. 로그인 시 아이디, 비밀번호를 입력 후 해당 아이디가 관리자 권한으로 설정되어있다면, 관리자 로그인이 성공했다는 메시지를 출력한다. 메시지 출력 후 신고 받은 목록 기능을 수행하기 위해 신고접수목록을 출력한다. 신고 목록이 성공적으로 출력되고, 자세한 내용을 보기위해 해당 포스트를 선택할 시, 공유 게시판의 해당 게시물으로 바로 접속한다. 접속 성공 후 게시물을 삭제한다. 게시물 삭제 및 불러오기를 실패한다면 실패메시지를 출력한다. 관리자 권한의 아이디지만, 권한을 사용 못하면 관리자 로그인 실패 메시지를 출력한다.

4. State machine diagram



어플리케이션을 실행 한 뒤 메인화면에서 다양한 기능을 수행한다. 회원가입을 통해 유저 및 관리자를 등록하고, 로그인을 통해 권한에 맞는 기능을 사용할 수 있다. 유저 권한을 가진 사용자는 자신이 포스트 작성하기, 작성한 포스트 리스트 보기, 공유된 포스트 목록 보기 및 자신이 작성한 포스트에 대한 공유, 수정, 삭제를 할 수 있고, 공유된 게시물을 신고할 수 있다. 관리자는 사용자들이 신고 접수한 게시물들을 확인하고 해당 포스트를 확인 후 게시물을 삭제할 수 있다.

5. Implementation requirements

- 1) Xcode 14.3.1버전 사용
- 2) iPhone 12 모델기기로 테스트

테스트는 위의 두가지 기기에서 실행했으며, 아이폰에서만 실행 가능하고, MacOS의 Xcode에서 소스코드를 실행할 수 있다.

6. Glossary

TERMS	Description
Class Diagram	UML에서 클래스 다이어그램은 구조적 다이어그램의 여섯 가지 유형 중 하나이다. 클래스 다이어그램은 오브젝트 모델링 프로세스의 근간이 되며 시스템의 정적 구조를 모델링한다. 시스템의 복잡도에 따라, 단일 클래스 다이어그램을 사용하여 전체 시스템을 모델링하거나 여러 클래스 다이어그램을 사용하여 시스템의 컴포넌트를 모델링할 수 있다
Attribute	UML 모델에서 속성은 클래스류 인스턴스에 속하는 정보, 데이터 또는 특성을 나타낸다. 클래스류에는 속성이 많이 있거나 전혀 없을 수 있다.
Method	메소드는 객체의 속성인 함수이다. 어떠한 특정 작업을 수행하기 위한 명령문의 집합
Sequence Diagram	시퀀스 다이어그램은 상호작용의 오브젝트 간 메시지 시퀀스를 설명하는 UML(Unified Modeling Language) 다이어그램이다.
State Machine Diagram	상태 머신 다이어그램은 객체의 행동 다이어그램 중 하나이다. 객체의 상태와 상태의 변화를 도식화 한 다이어그램이다.

7. References

- [1] [Design] Example 1~6
- [2] 수업 강의 자료
- [3]~[7] : Glossary 정의
- [3] <https://www.ibm.com/docs/ko/rsm/7.5.0?topic=structure-class-diagrams>.
- [4] <https://www.ibm.com/docs/ko/rsas/7.5.0?topic=classifiers-attributes>
- [5] <https://developer.mozilla.org/ko/docs/Glossary/Method>
- [6] <https://www.ibm.com/docs/ko/rsas/7.5.0?topic=uml-sequence-diagrams>
- [7] <https://sabarada.tistory.com/192>