

소프트웨어설계 프로젝트 결과보고서

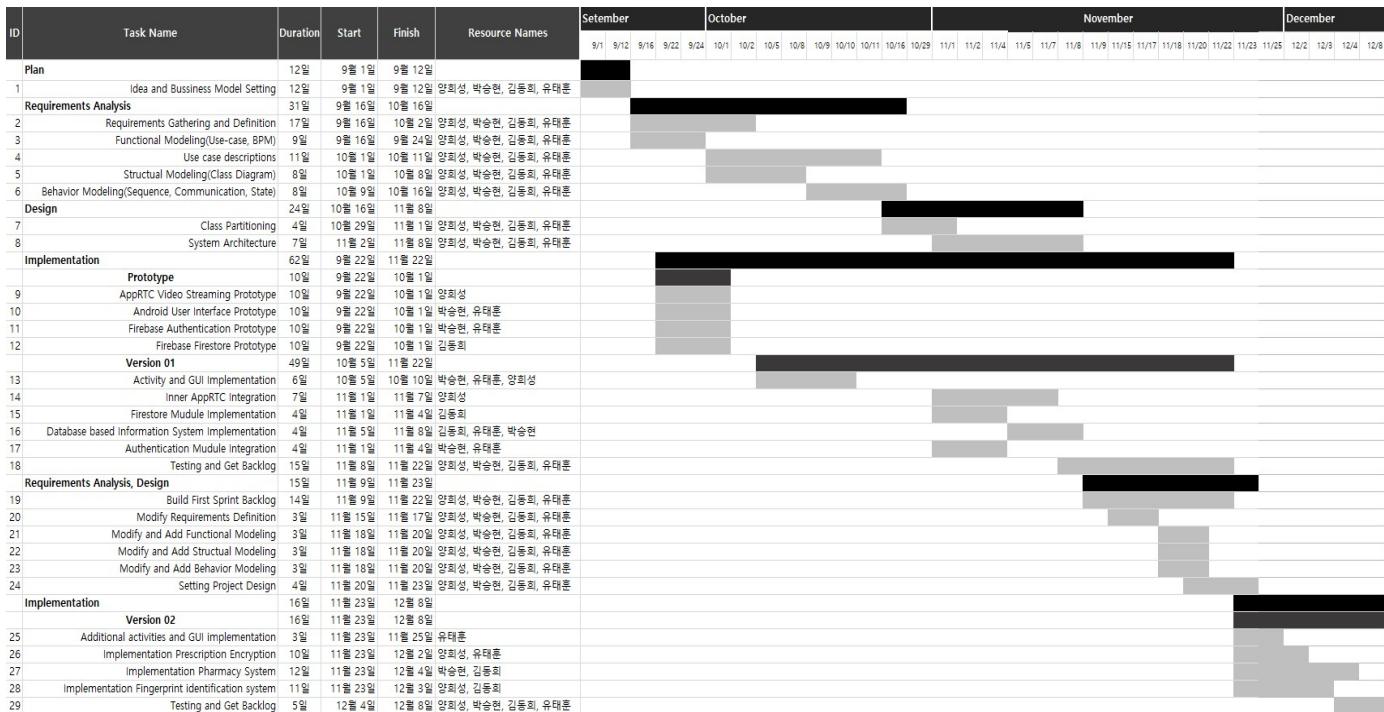
프로젝트명:		Mobile Telemedicine System			
9	조	팀원:	양희성, 김동희, 박승현, 유태훈		

1. 프로젝트 개요

전국민이 코로나19 바이러스로 인한 감염의 위험에 노출되어 인류의 적은 바이러스임을 깨닫고, 추가적으로 전염성이라는 바이러스의 속성에 반하는 언택트 시장을 겨냥한 아이디어가 없을까를 생각함. 이후 나온 솔루션이 원격 의료 서비스이고, 이 것이 현재의 정책에서는 불법으로 해석될 수 있는 여지가 많다. 하지만 현재 정부에서 4대 의료 개정안으로 내놓은 개정안 중 하나가 원격의료관련 법이다. 정책적으로 규제가 풀린다면 새로운 시장이 열릴 것이다.

우선 환자의 입장에서는 병원에 가지 않기 때문에, 감염의 위험성이 매우 감소하고, 병원에 가기 위한 시간, 교통비용 등 여러가지 추가 발생하는 비용을 줄임으로서 도움이 된다. 병원의 입장으로는 최근 병원 내원 환자 수 감소로 인한 경영난 해소, 환자 대기실 등의 공간 효율화, 의료 인력들의 감염 위험 감소, 의사들의 진료 효율 극대화의 효과가 있다.

2. 프로젝트 관리



양희성 : System Analysis, Project Manager, Android Staff, Developer

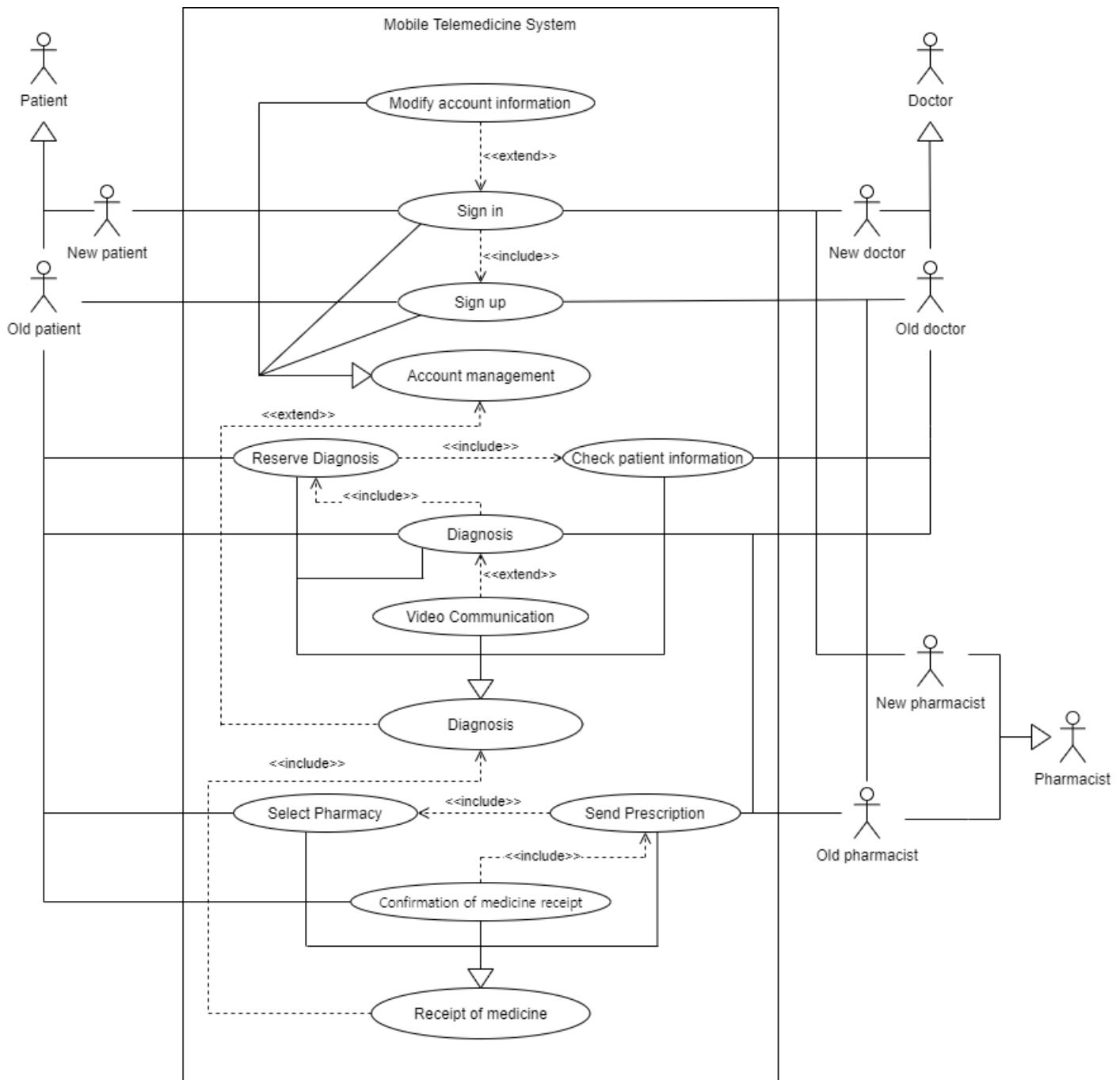
김동희 : System Analysis, Project Manager, Back-end Develop Staff, Infrastructure Staff, Developer

박승현 : System Analysis, Project Manager, Business Modeling Staff, Developer

유태훈 : System Analysis, Project Manager, Front-end Develop Staff, Developer

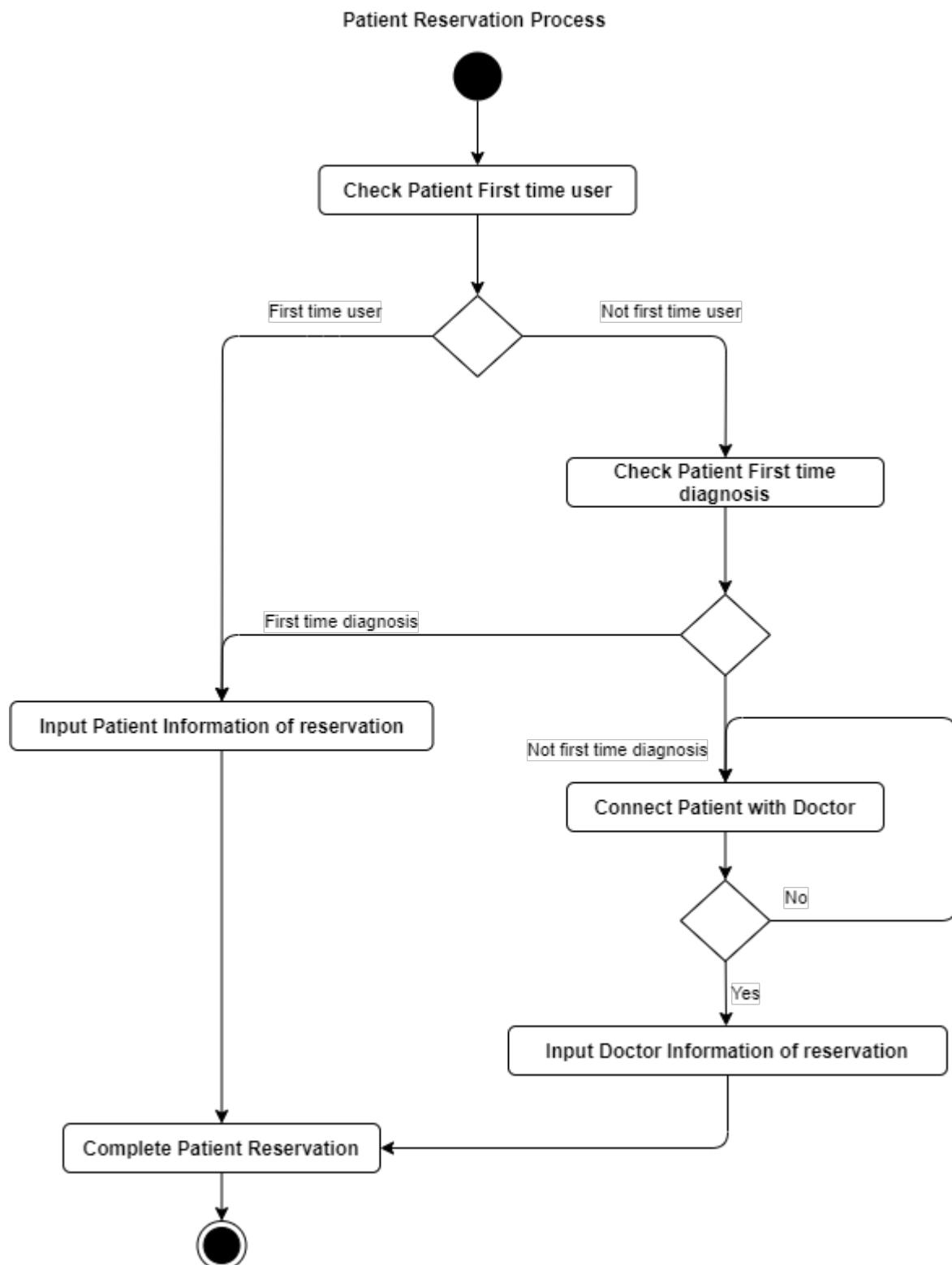
3. 시스템 기능 및 동작

3.1 Use-case Diagram (Version.02 추가 내용)

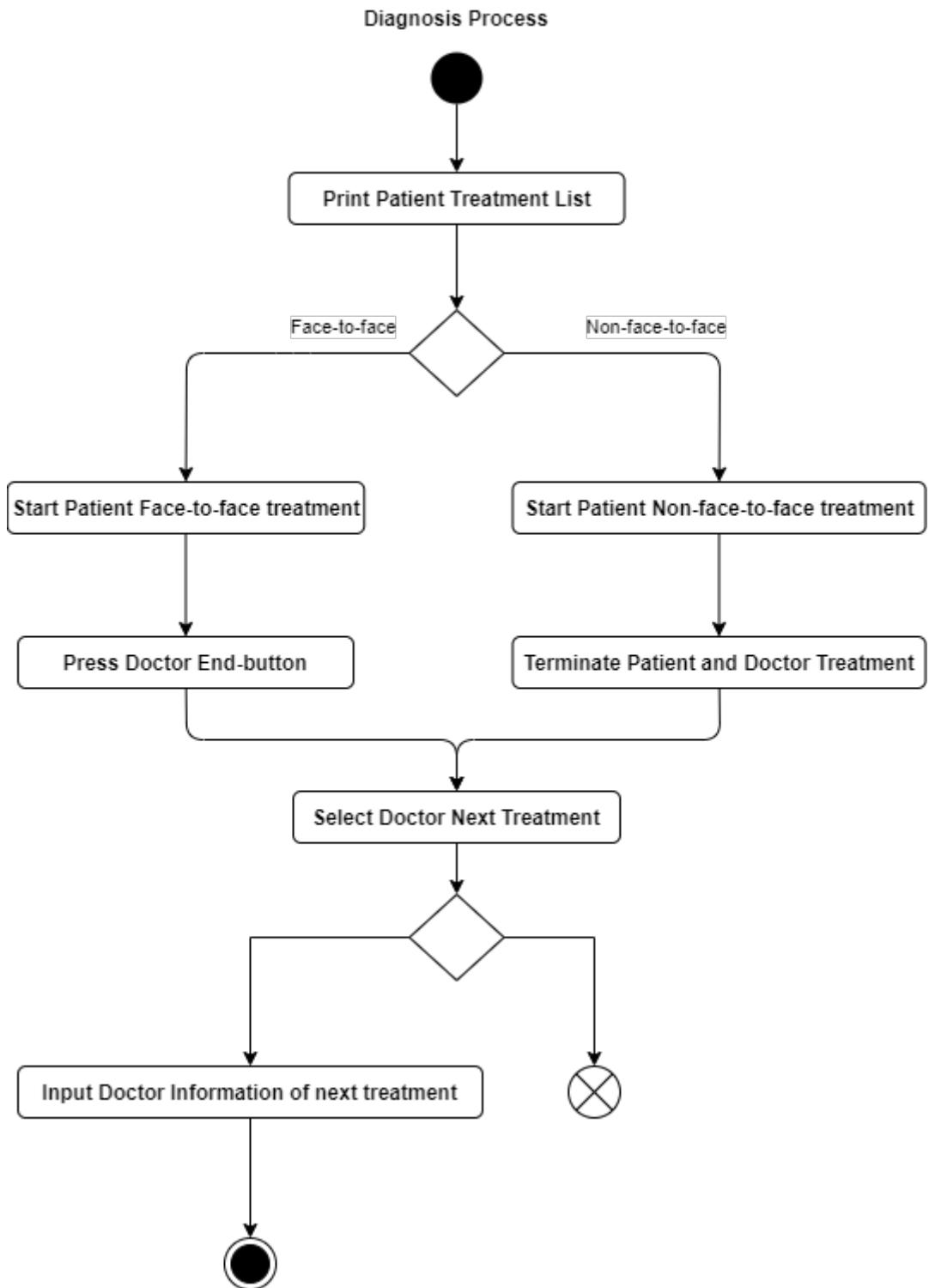


3.2 Activity Diagram

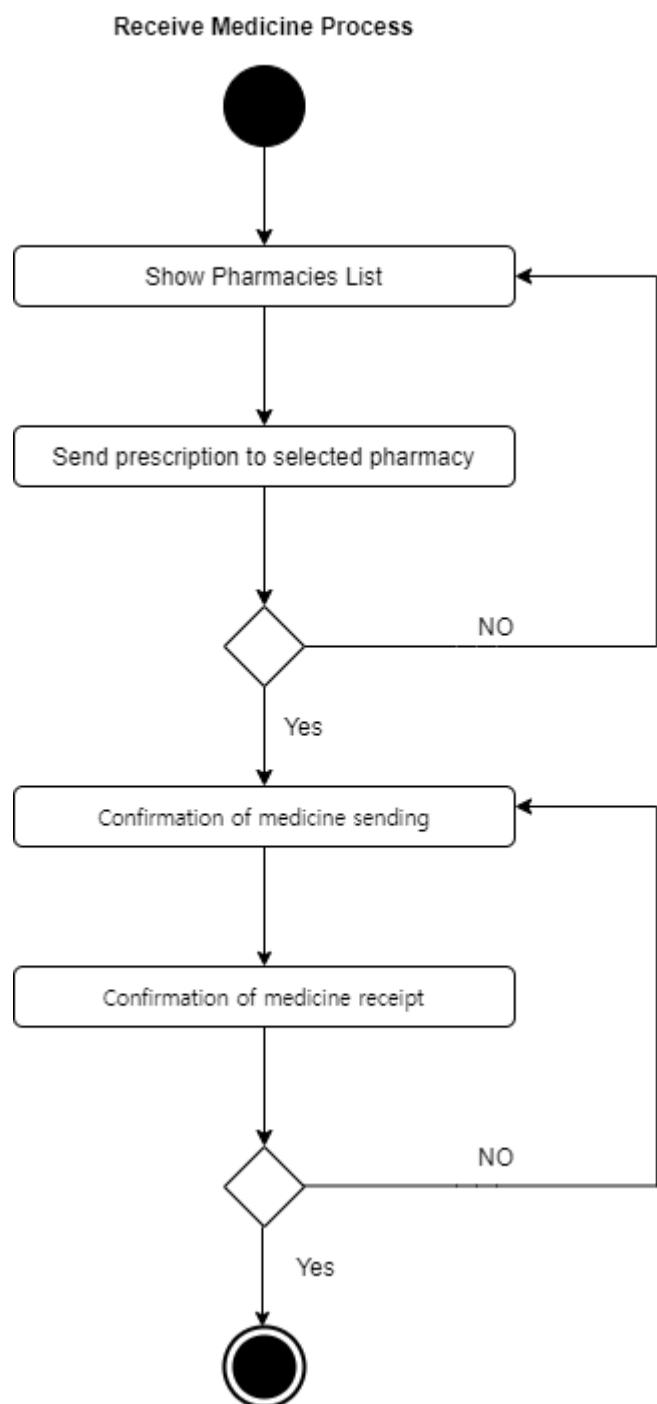
3.2.1. Patient Reservation



3.2.2 Diagnosis



3.2.3 Receipt of Medicine (Version.02 추가 내용)



4. 시스템 아키텍처

4.1 Hardware :

개발 :

- Samsung notebook(Window10)
- Macbook pro (Mac OS Catalina)
- Android AVD(Pixel 2)
- Samsung Galaxy S8(Android 9 Pie)

시연 :

- Samsung Galaxy S20(Android 10)
- Samsung Galaxy Note 10(Android 10)

4.2 Software :

개발 :

- Google Firebase Console
- Android Studio

협업 및 환경 :

- Github Desktop
- Cloud Firestore
- Firebase Auth
- Firebase Realtime Database
- AppRTC

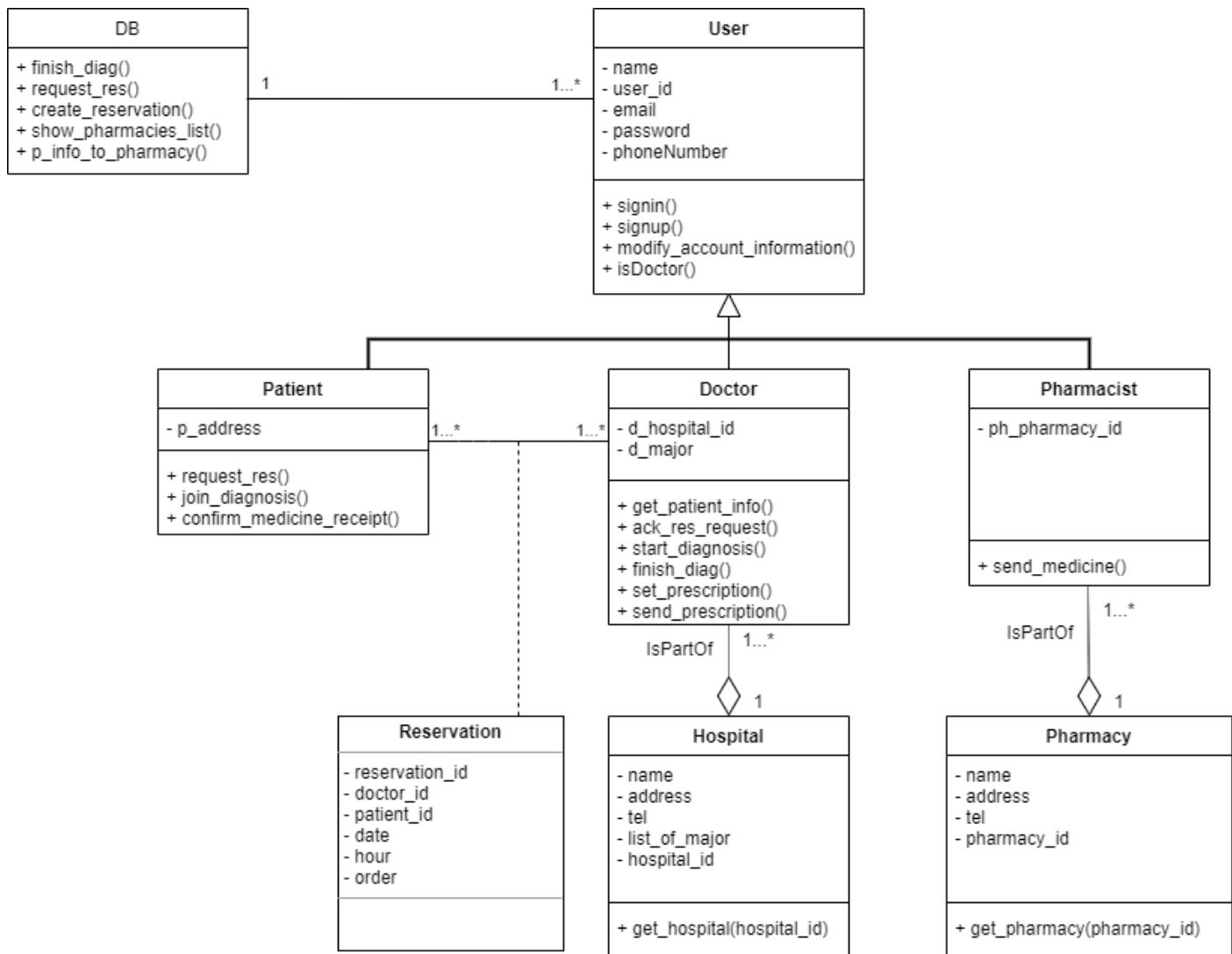
4.3 System Model :



5. 소프트웨어 상세 설계

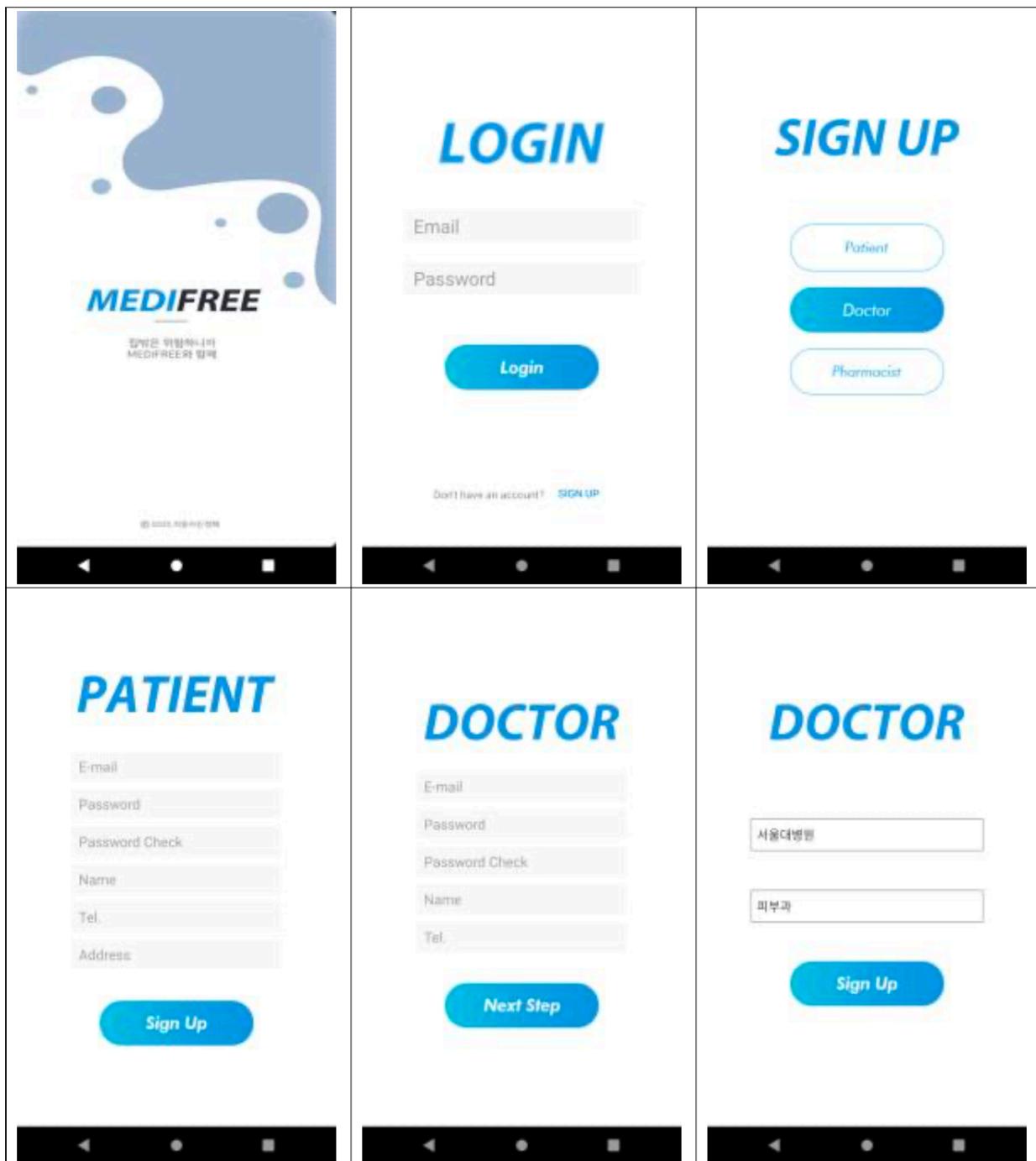
5.1 정적 모델

5.1.1 Class Diagram (Version.02 추가 내용)



5.2 GUI 설계 (Version.02 추가 내용 존재)

1) 계정 회원가입 및 로그인



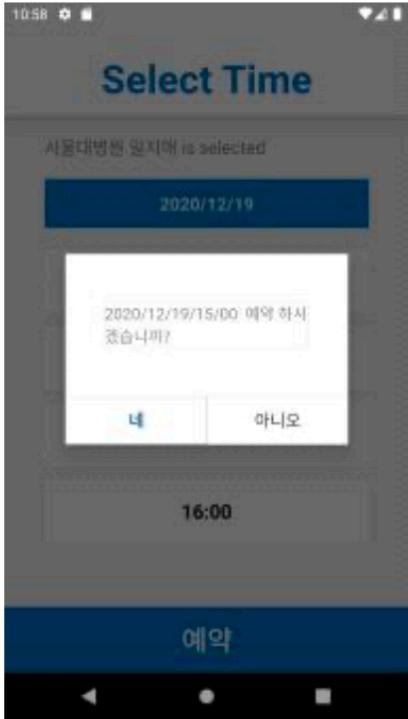
2) 환자 부분

2-1 승인된 예약 화면



2-2 예약하기

1. 예약 버튼을 눌러 병원 리스트를 확인한다.	2. 선택한 병원의 희망 진료과를 선택한다.	3. 해당 진료과의 의사를 선택한다.
<p>The screenshot shows a list of hospitals: 서울대병원, 영남대병원, 동산병원, 경북대병원, and 연세세브란스병원. Each item has a circular icon with a hospital building and a name.</p>	<p>The screenshot shows a list of medical specialties under the heading "서울대병원 is selected": 피부과, 산부인과, 정형외과, 내과, 비뇨기과, and 신경외과.</p>	<p>The screenshot shows a list of doctors under the heading "피부과": 일지매, 홍길동, Dr.Suh, 이몽룡, 성춘향, and 김두한.</p>

4. 진료 희망 시간을 선택한다.	4-1. 희망 날자를 고른다.	4-2 예약 하기 버튼을 누른다.
		

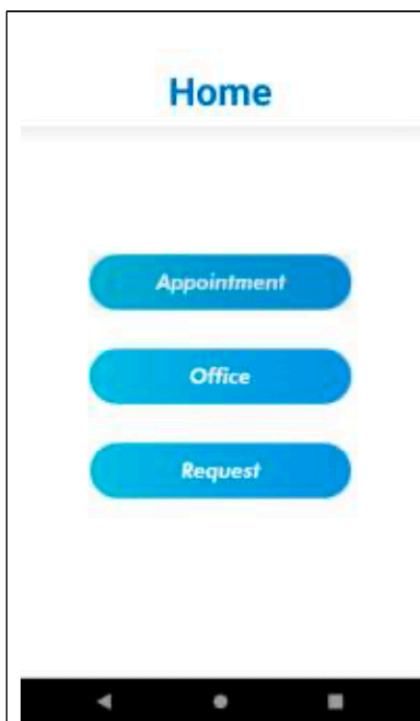
5. “네”를 누르면 진료 예약 요청을 보냄.


2-2-1 예약하기 예외처리

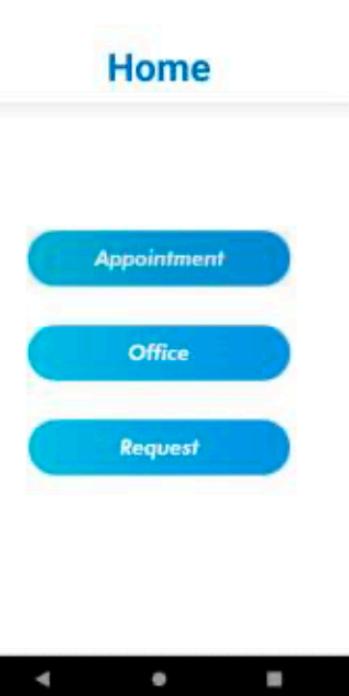
- 시간을 고르지 않고 예약 클릭 시 실패	- 과거 날자를 고르면 예약 클릭 실패

3) 의사 부분

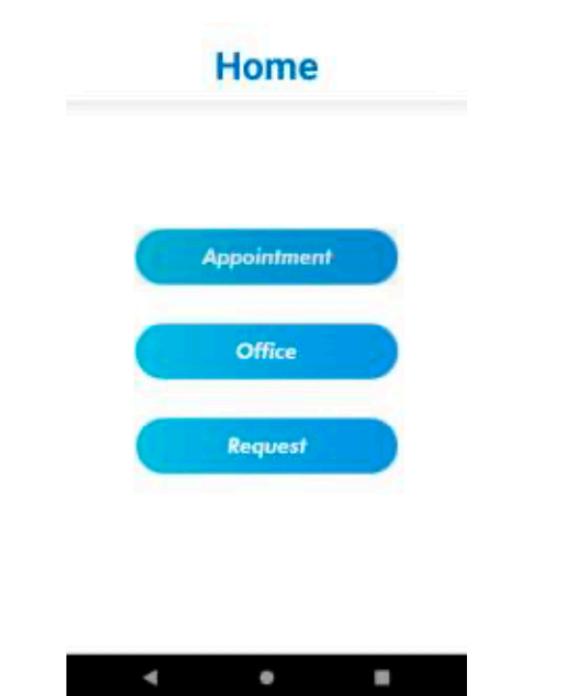
3-1 의사 기본 페이지.



3-2 승인된 예약 확인하기.

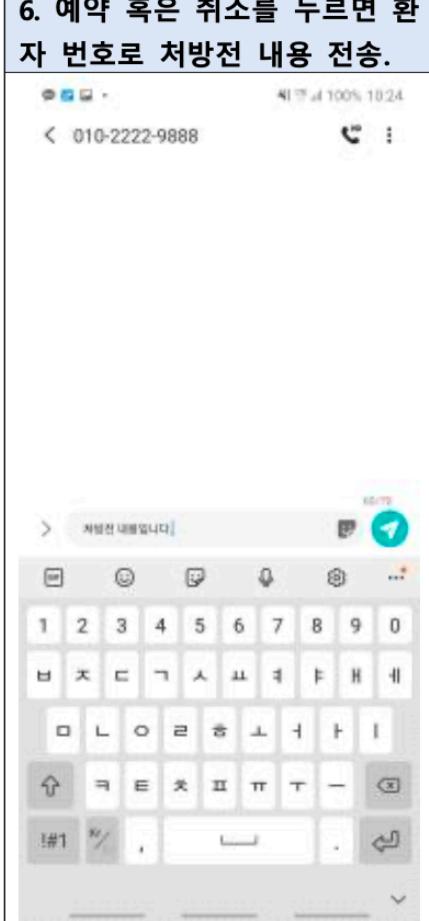
1. Appointment 버튼을 클릭한다.	2. 예약 시간 중 보고 싶은 시간을 선택한다.	3. 예약 된 환자의 리스트를 볼 수 있다.
		

3-3 예약 요청 확인하기.

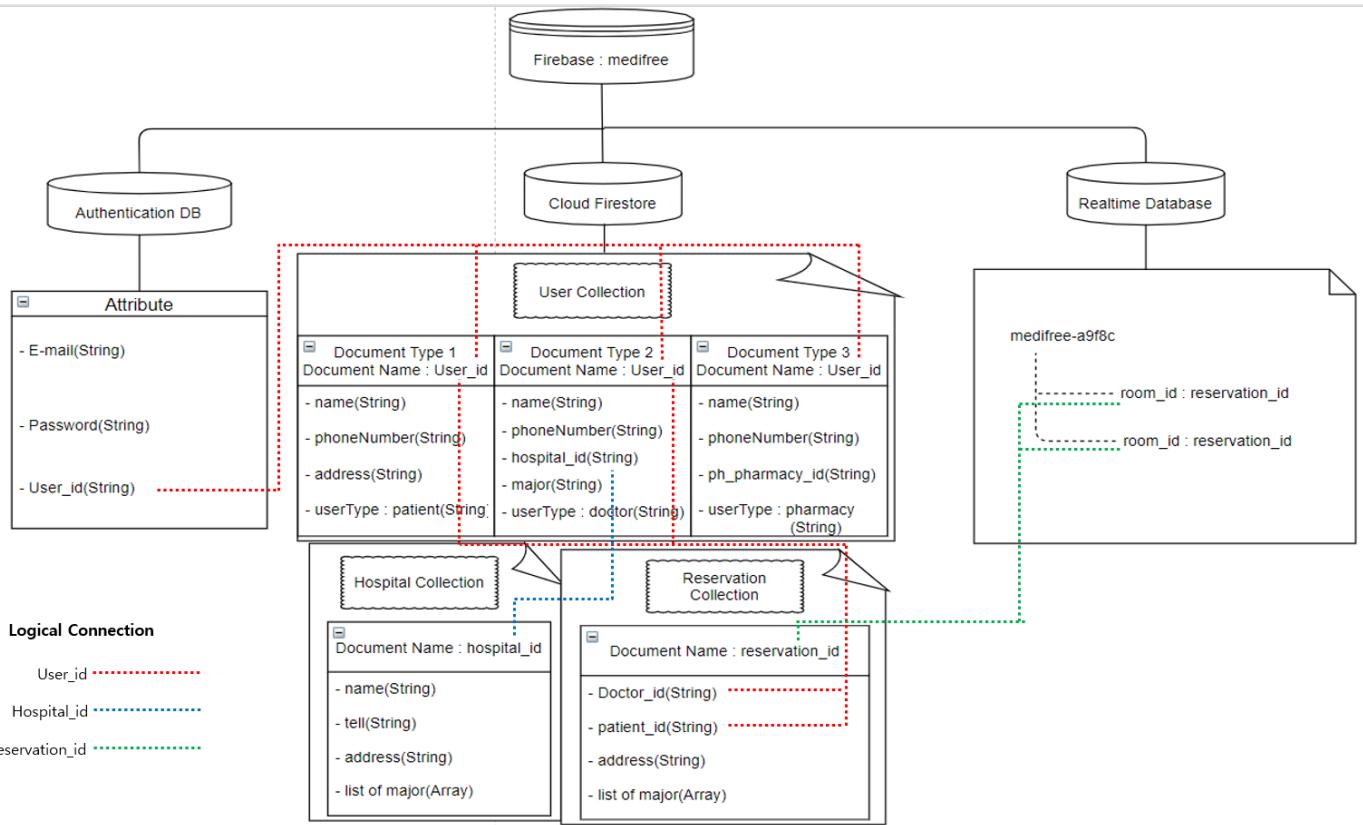
1. Request 버튼을 클릭한다.	2. 진료 요청 목록을 보고 수락 및 거절을 클릭하면 서버로 전송된다.
	

4) 화상 진료 부분

1. (의사화면) 진료 대상 선택 후 진료 시작 버튼 클릭	1. (환자화면) 진료실 버튼 클릭	2. (의사화면) 화상통화 시작
2. (환자화면) 화상통화 시작	3. (의사화면) 진료 후 처방전 작성	4. (의사화면) 전송 버튼 클릭 했을 때 다음 예약 물어 봄.

5. 다음 예약 날짜 일시 선택	5. 다음 예약 날짜 일시 선택	5. 다음 예약 날짜 일시 선택
<p>Next Appointment</p> <p>YY/MM/DD 09:00</p> <p>예약 취소</p>	 <p>2020/12/18 15:00</p> <p>예약 취소</p>	<p>Next Appointment</p> <p>2020/12/18 15:00</p> <p>예약 취소</p>
<p>6. 예약 혹은 취소를 누르면 환자 번호로 처방전 내용 전송.</p> <p>010-2222-9888</p> 		

5.3 데이터 설계



Medifree는 관계형 데이터베이스가 아닌, NoSQL기반의 Firebase를 통해 구현되었다. 따라서 관계형 데이터베이스 기반의 요소가 아닌 Firebase의 FireStore의 구성요소인 Collection, Document, field의 형태로 데이터베이스에 대한 설계를 하게 되었다.

Firebase에 대해서 간단히 설명을 하자면, 첫 번째로 medifree의 모든 data가 주로 저장되는 Cloud FireStore DB가 있다. FireStore는 Collection안의 Document로 이루어져 있으며, 이는 window filesystem의 폴더와 file의 관계와 비슷하다고 볼 수 있다. 따라서 document안에 data가 각각에 field에 맞춰서 다양한 data type으로 저장되어 있다. 두 번째로는 Authentication이다. Firebase에서 자동으로 login과 signup 기능을 구현해주고 따로 오류처리를 해주는 (비밀번호의 길이, email의 중복성 등) 편리함이 있어 사용하였다. 여기서 회원가입을 할 때 email마다 user_id를 발급해주며 이것은 random 한 String의 조합으로 이루어져 있기 때문에 정보보호 측면에서도 의미가 있어 이 user_id를 사용하기로 결정했다. 마지막으로 realtime database이다. realtime database는 Json형식으로 data를 실시간으로 생성, 삭제가 가능하다. 이와 관련된 시스템으로는 medifree의 화상통화 모듈의 database로 사용하였다.

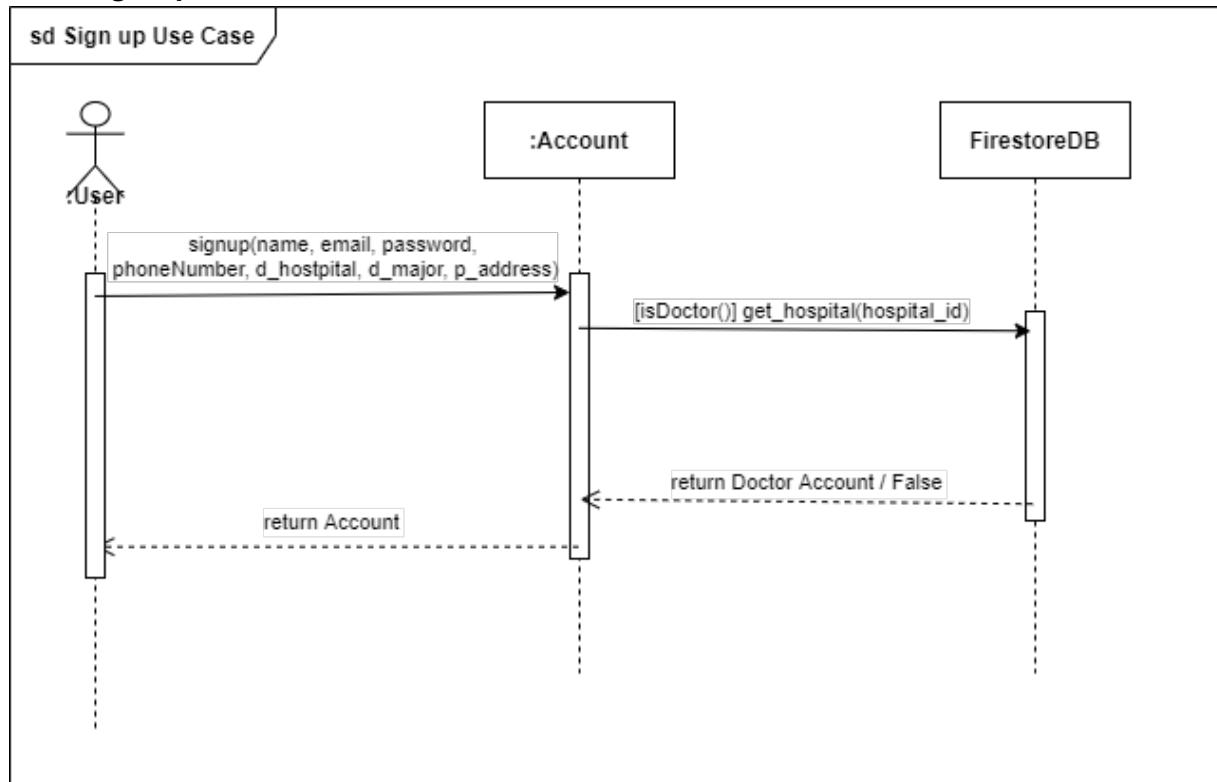
FireStore은 MySQL과 같은 DBMS처럼 FK, PK의 개념이 없다. 따라서 FK가 없어서 따로 다른 collection를 참조하지 못한다. 그래서 코드 내부적으로 업데이트를 해줘야한다. 따라서 medifree는 각 collection 문서를 생성할 때 자동으로 생성되는 해당 문서의 id를 재사용하여 논리적 connection이 이루어지게 하였다.

- Authentication DB
 - ✓ E-mail : user의 로그인시 입력하게 되는 email이다.
 - ✓ password : 로그인시 입력하게 되는 password이다.
 - ✓ user_id(UUID) : e-mail마다 생성되게되는 String의 여러조합으로 이루어진 식별자다.
- Cloud FireStore
 - ✓ User Collection : User들의 개인정보를 저장하는 Collection이다. 문서의 이름은 Authentication에서 발급한 user_id로 설정하고 문서 내부의 field 중 userType을 통해서 이 문서가 Doctor인지 patient인지 Pharmacy인지 구별하게 된다.
 - ✓ Hospital Collection
 - document이름은 문서를 생성할때 자동으로 지정해주는 id이고 이것이 hospital의 id가 된다.
 - 각 문서의 field는 Hospital의 정보를 담고 있다.
 - Doctor document의 field중 hospital_id은 hospital 문서의 id를 사용하여 논리적으로 연결되게 했다.
 - ✓ Reservation collection
 - Hospital 문서와 같이 reservaiion 문서의 문서이름은 FireStore에서 자동생성해주는 id이다.
 - Reservation의 field로 patient와 doctor의 UID를 넣음으로써 User Collection과 논리적으로 연결하였다.
 - Reservation_id는 화상모듈의 room id로 사용한다.
- Realtime Database
 - ✓ 화상을 진행할 때, 즉 비대면 진료를 직접 진행할때 reservation_id를 이름으로 통신을 하는 하나의 방이 생성된다.
 - ✓ reservation_id로 논리적인 연결을 하기 때문에, 해당 id의 문서의 date field를 참조하여서 시간이 되었을 때, room을 자동 생성해주고 화상통화가 끝났을 때 해당 room을 삭제 해준다.

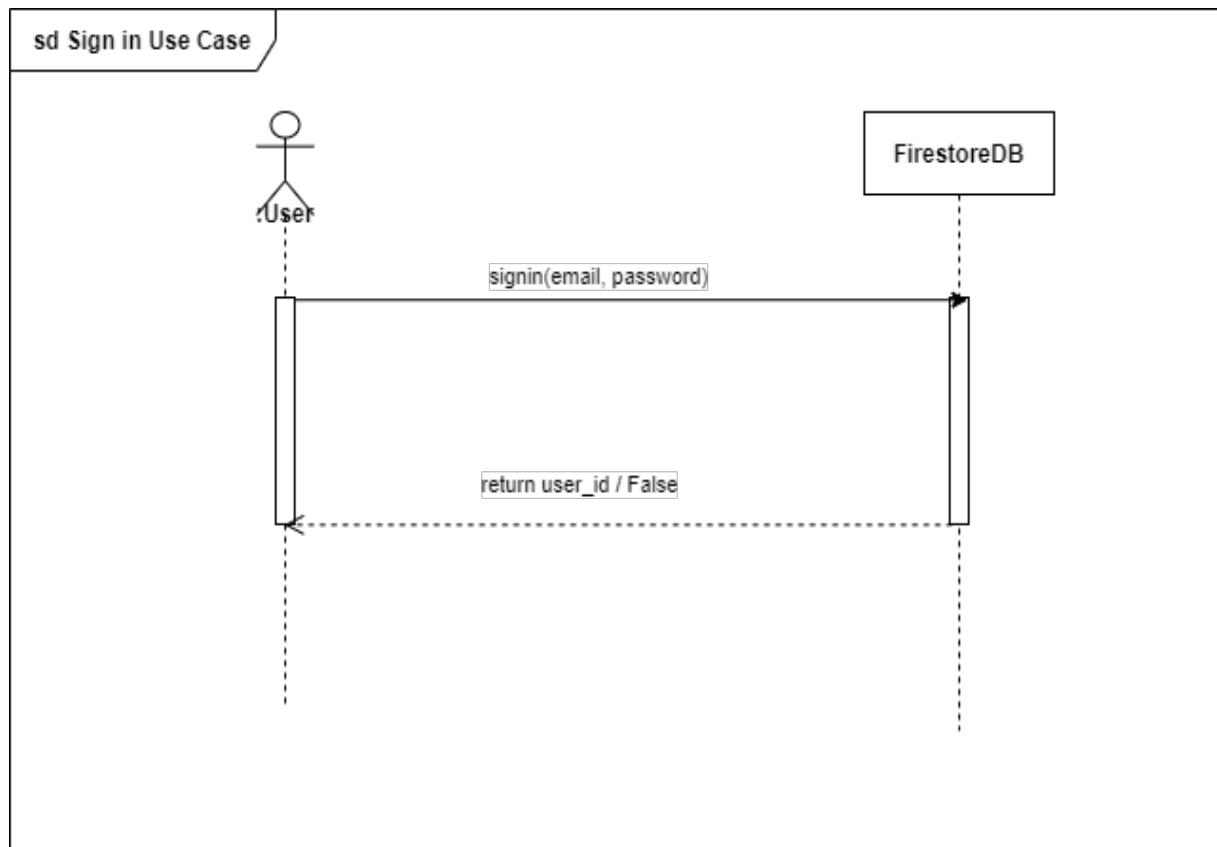
6. 소프트웨어 주요 기능 (동적 모델, 시퀀스 다이어그램 등으로 동작 표현)

6.1 Sequence Diagrams

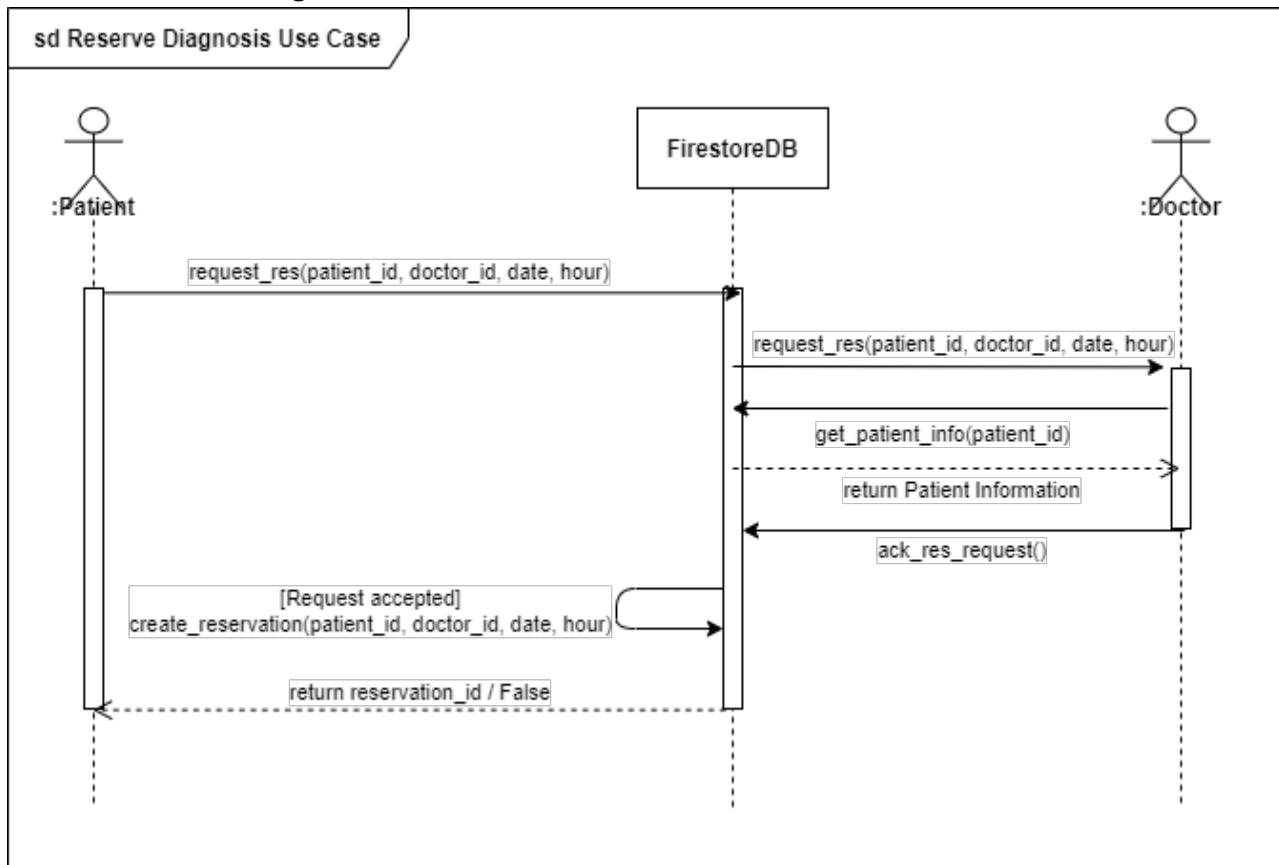
6.1.1 Sign up



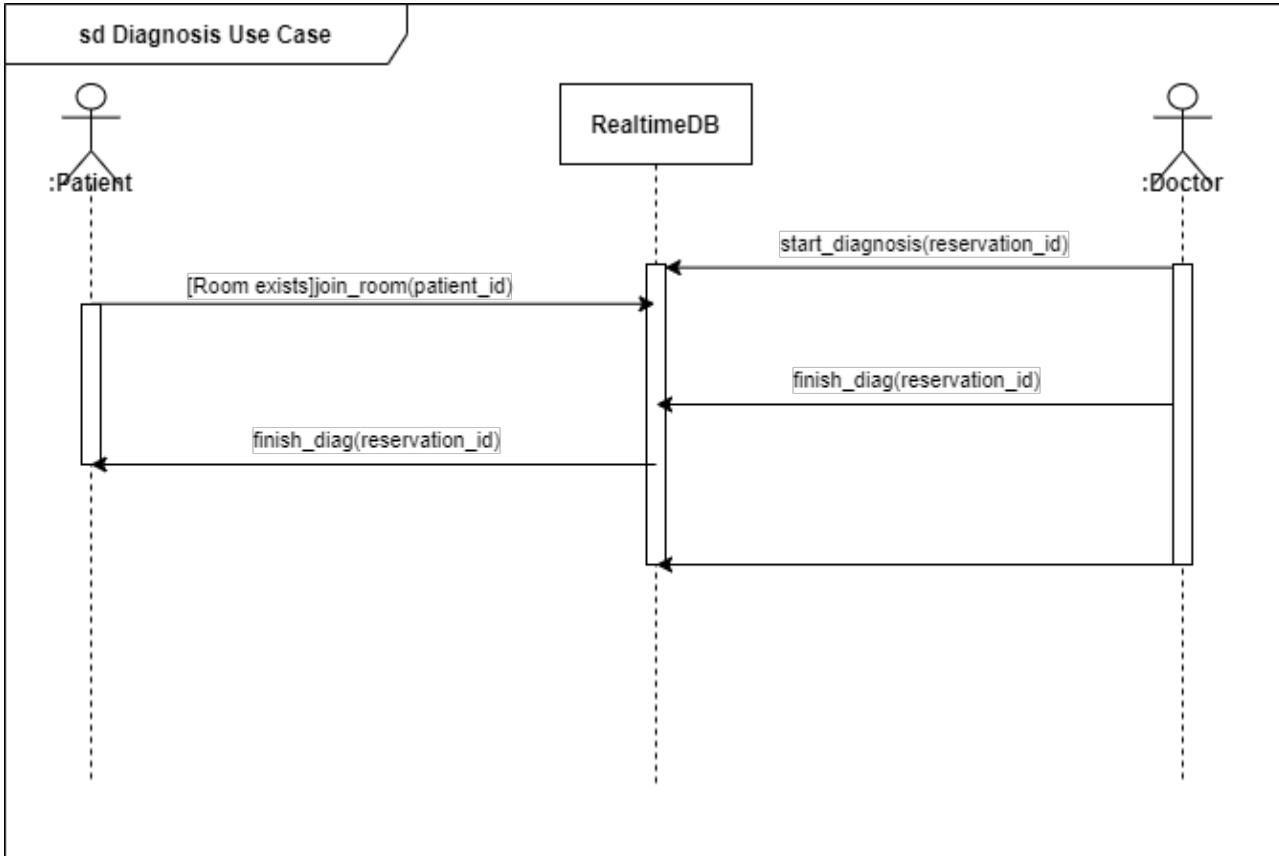
6.1.2 Sign in



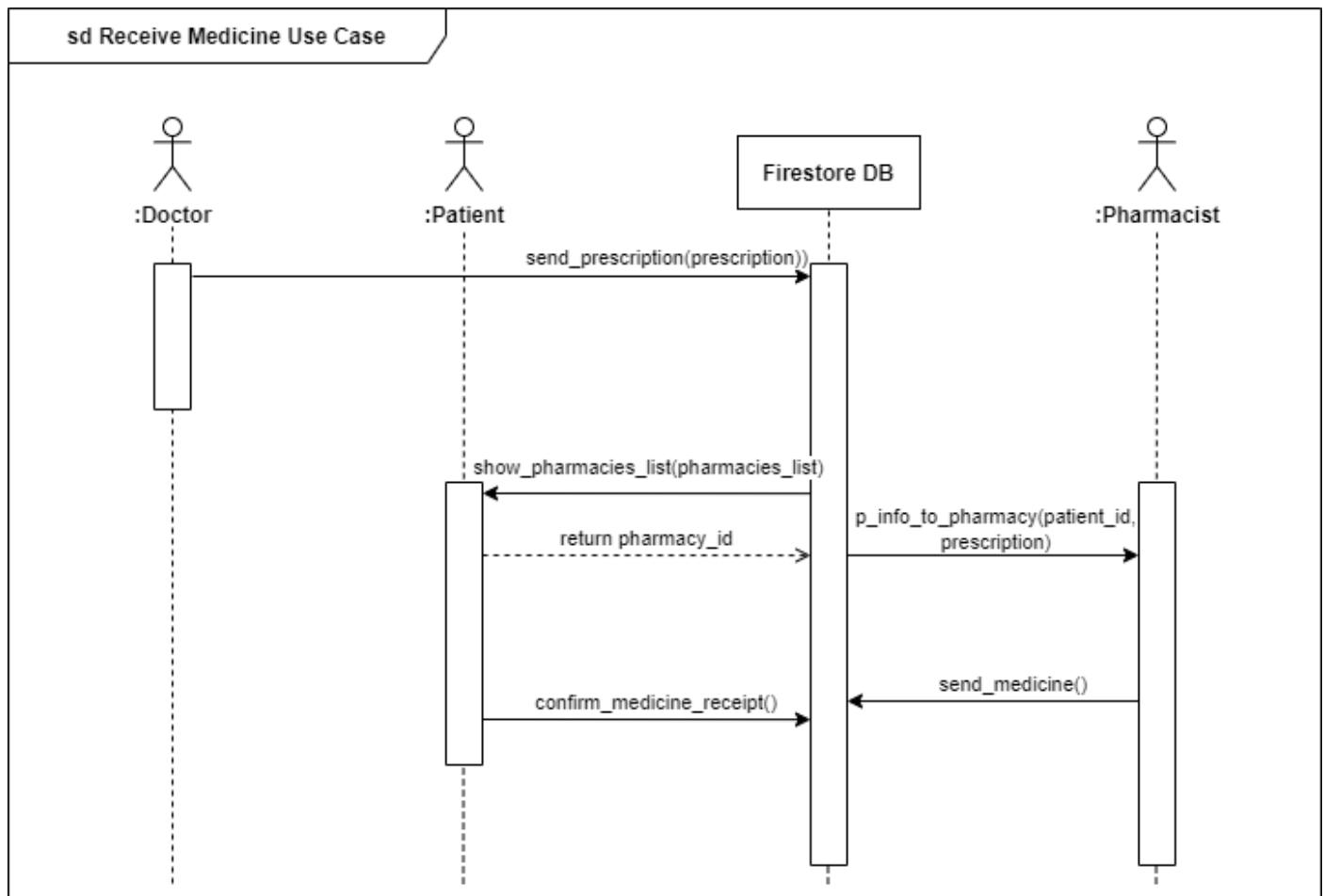
6.1.3 Reservation Diagnosis



6.1.4 Diagnosis

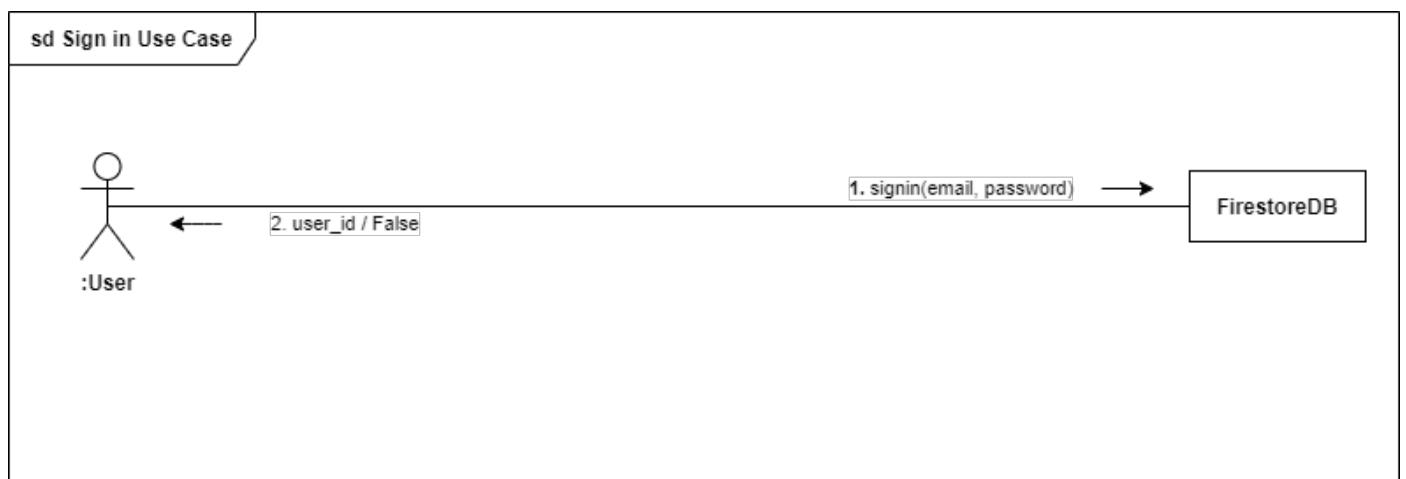


6.1.5 Receipt of Medicine (Version.02 추가 내용)

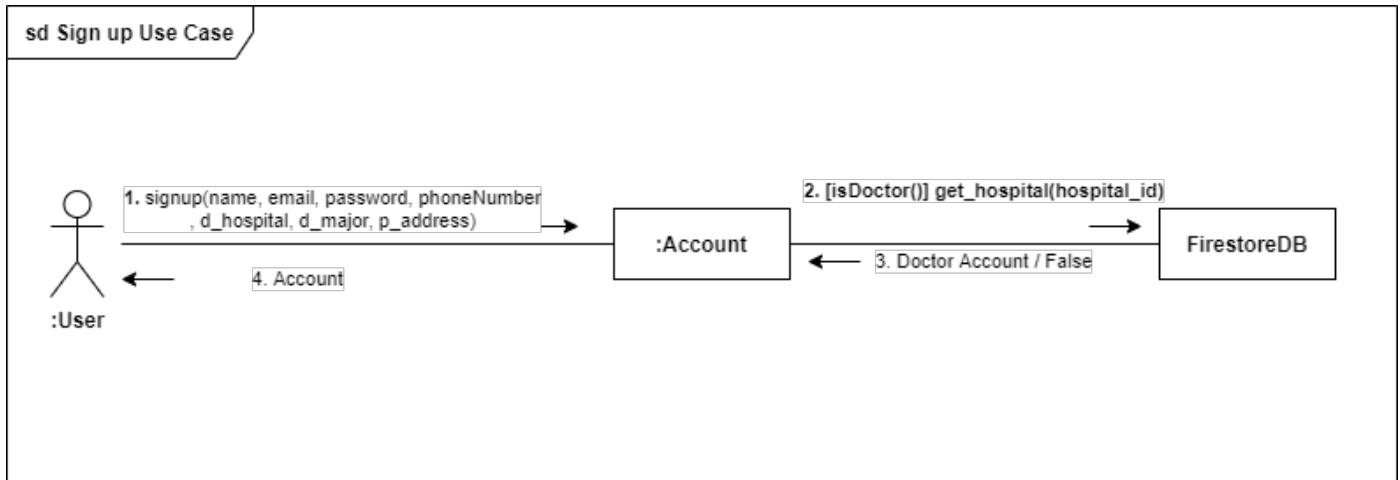


6.2 Communication Diagram

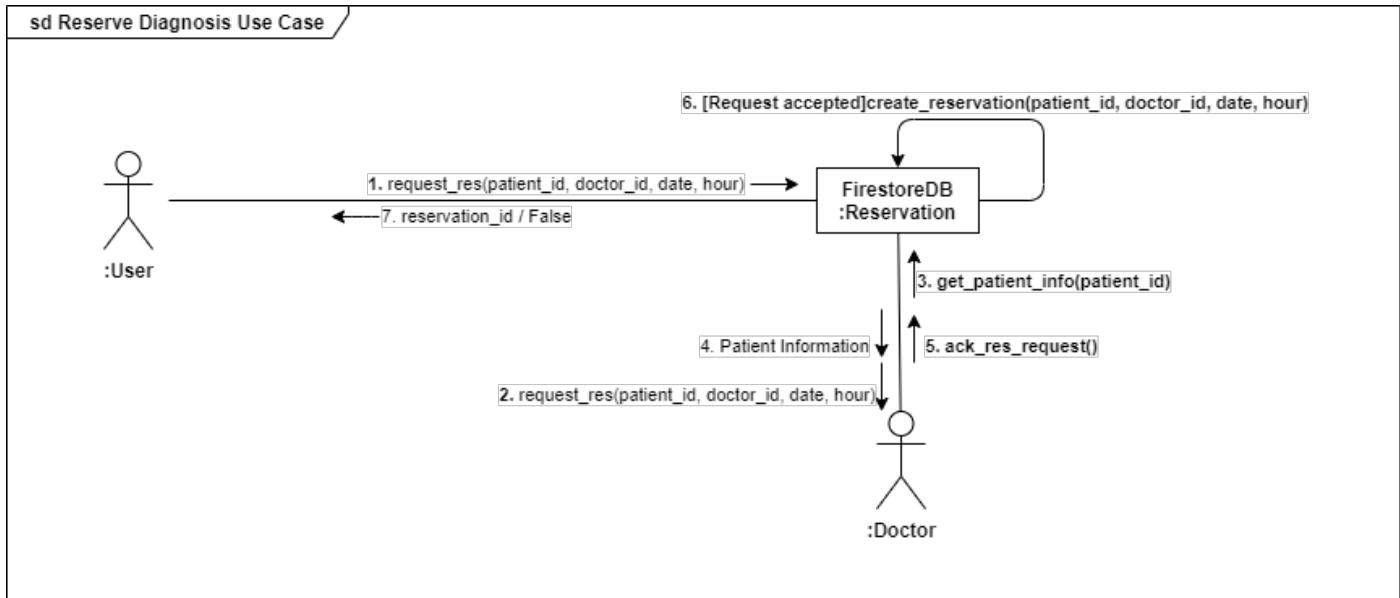
6.2.1 Sign in



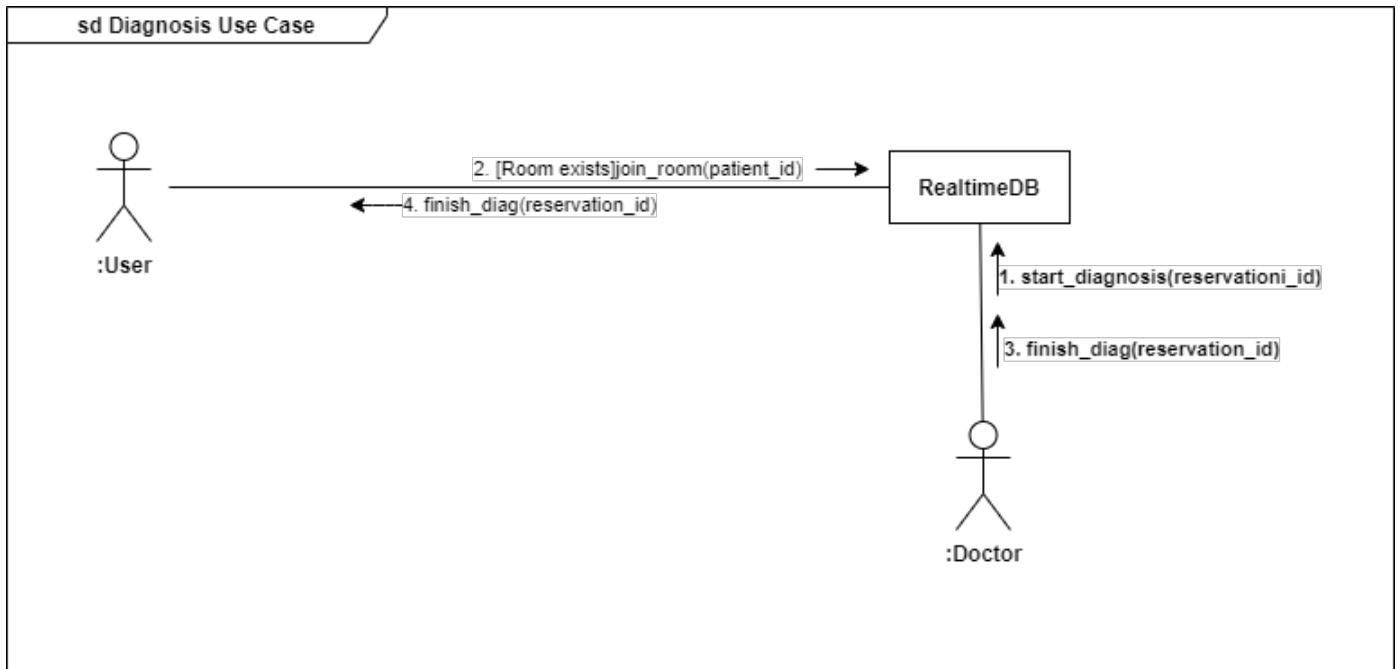
6.2.2 Sign up



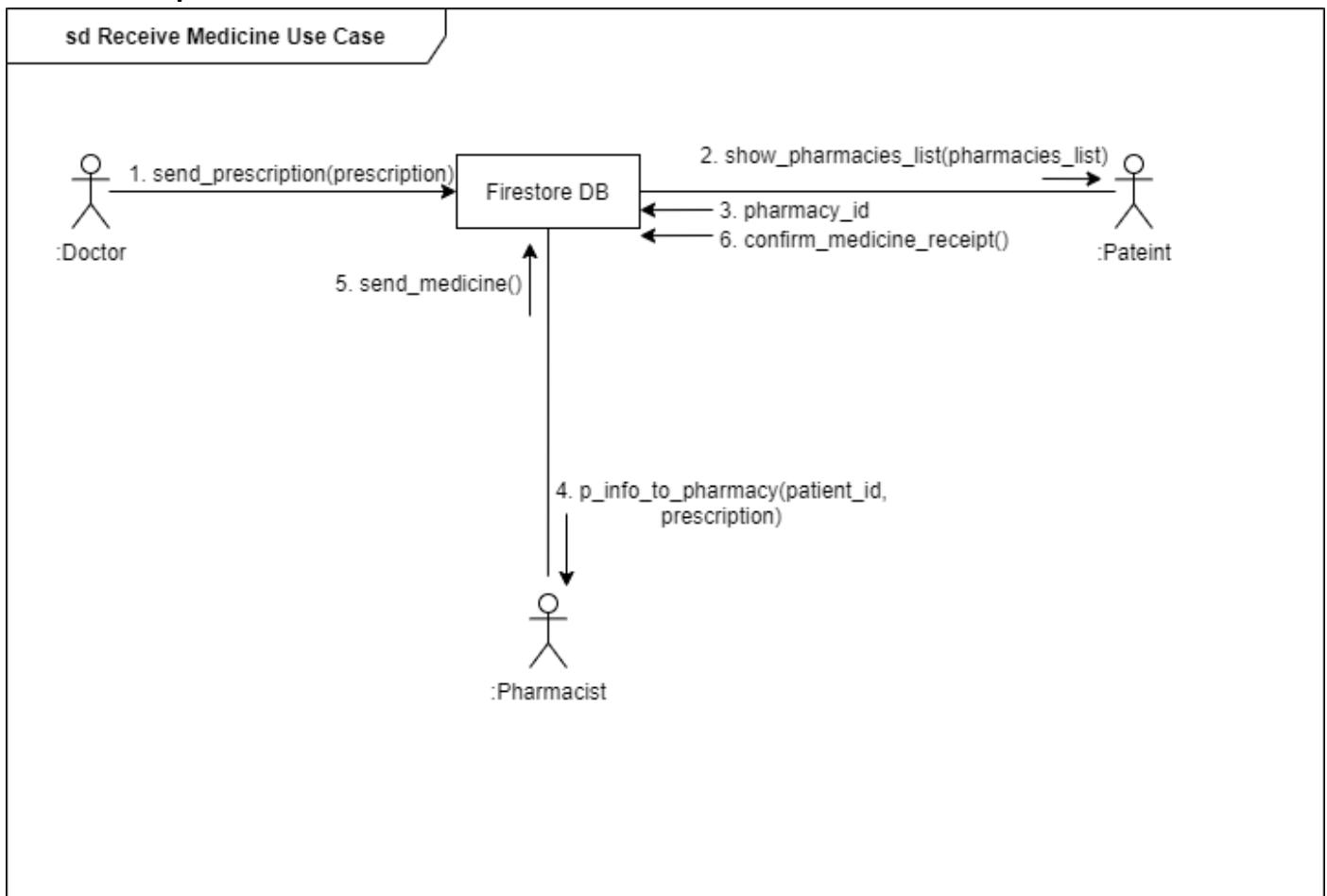
6.2.3 Reservation Diagnosis



6.2.4 Diagnosis

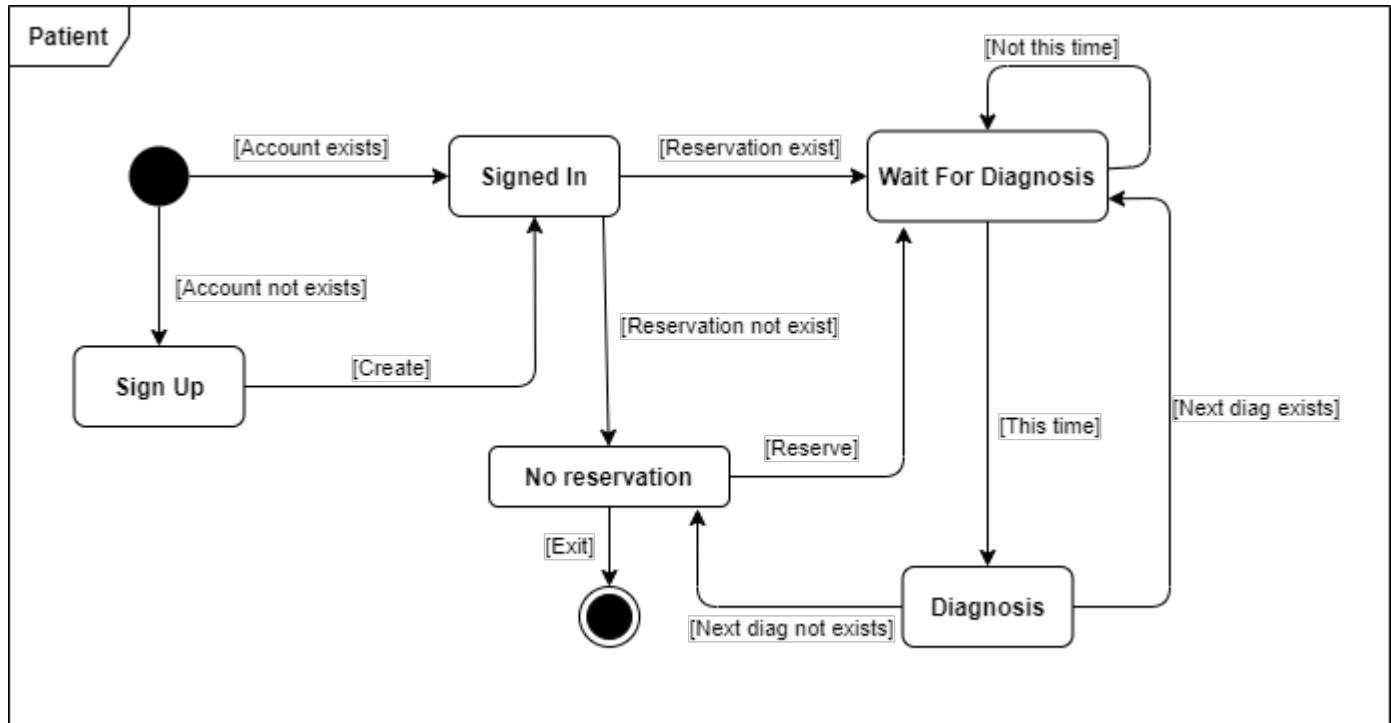


6.2.5 Receipt of Medicine (Version.02 추가 내용)

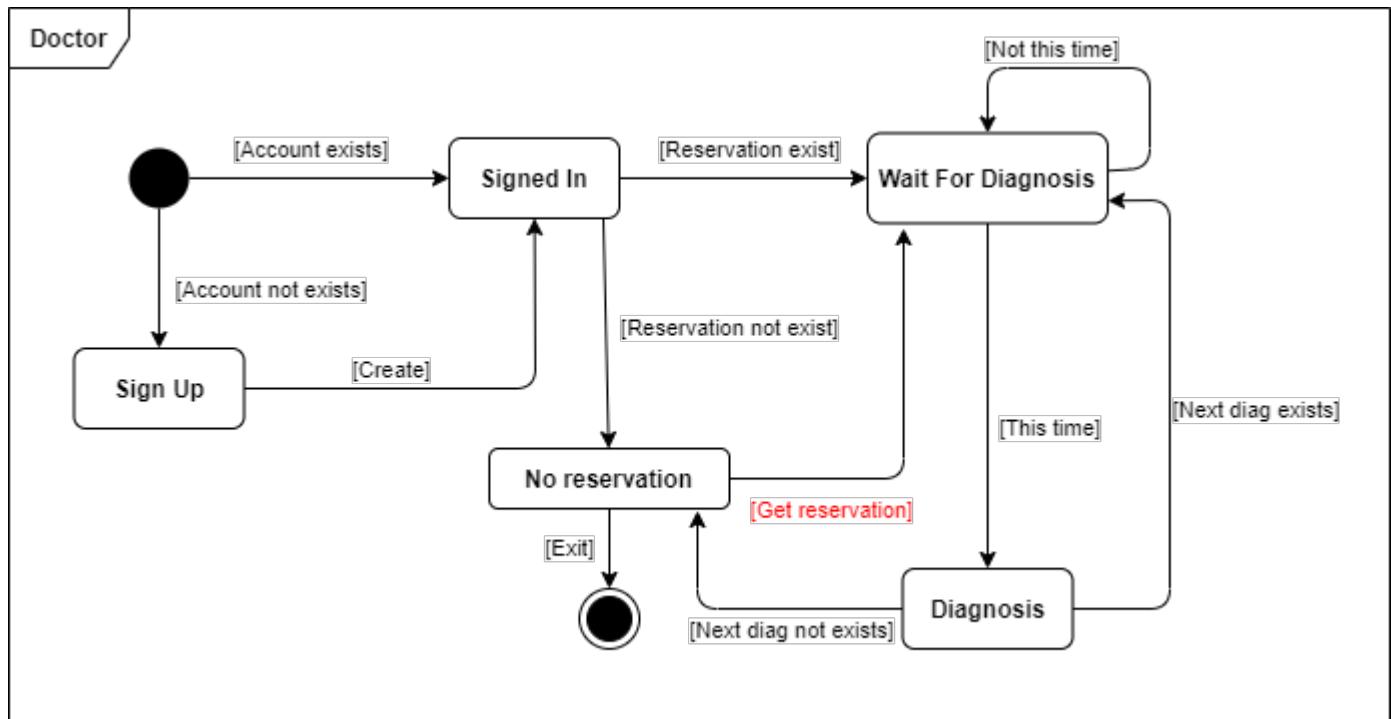


6.3 State Machine Diagram

6.3.1 Patient



6.3.2 Doctor



7. 테스트 계획 및 결과 (Version.02 추가 내용)

7.1. 테스트 계획

대분류	소분류	ID	테스트 스텝	기대 결과
Account Management	Sign up	1	유저가 회원가입 액티비티에서 유효한 입력 후 회원가입 버튼을 누른다.	DBMS에 유저정보가 등록된다.
	Sign in	2	유저가 유효한 ID/PW를 입력후 로그인 버튼을 누른다.	로그인이 완료된 홈화면으로 이동한다.
		3	유저가 유효하지 않은 ID/PW를 입력후 로그인 버튼을 누른다.	오류메시지를 출력한다.
		4	유저가 'ID/비밀번호 찾기' 버튼을 누른다.	계정 찾기 액티비티로 이동한다.
	Modify User Info	5	계정 정보 수정 화면에서 유효한 입력 후 수정 버튼을 누른다.	DBMS에 등록된 유저정보가 수정된다.
Diagnosis	Reservation	6	초진 환자 또는 진료 이력이 없는 경우 예약 버튼을 누른다.	초진 예약 정보 입력창으로 이동한다.
		7	재진 환자가 확인된 경우 예약 버튼을 누른다.	담당 의사 확인 화면으로 이동한다.
	Video Streaming Service	8	의사가 통화 버튼을 누른다.	Realtime DB에 방 생성이 되고, 환자는 접속 가능.
		9	의사가 생성한 방이 존재할 때 환자가 통화 버튼을 누른다.	방 접속 후 의사와의 화상통화 화면이 표시된다.
		10	유저가 종료버튼을 누른다.	의사는 예약 정보 입력창이 표시된다. 환자는 홈화면으로 이동한다.
		11	의사가 진료 후 다음 예약 정보를 입력 완료한다.	환자의 예약 정보 표시창에 예약 정보가 출력된다.
Receipt of Medicine	Select Pharmacy	12	진료가 끝난 환자에게 약국리스트를 선택하게 한다.	선택된 약국에 처방전이 전송된다.
		13	환자가 선택한 약국에 전송된 처방전에 따른 약물 조제가 불가능한 경우	환자에게 다시 약국을 선택하게 한다.
	Send Prescription	14	진료가 끝난 의사가 처방전을 업로드한다.	환자가 선택한 약국에 처방전이 전송된다.
	Confirmation of medicine receipt	15	약사가 약물 발송을 누른다.	환자가 약물을 수령하고 약사에게 완료 확인결과가 간다.

7.2. Test Log(2020.12.18 최신)

Test ID : 1) 정상 작동 확인

Test ID : 2) 정상 작동 확인

Test ID : 3) Firebase Auth Service 고유 오류메시지 출력. 변경 요망.

Test ID : 4) Version2 구현 제외

Test ID : 5) Version2 구현 제외

Test ID : 6) 7번 테스트와 통합 내용. 정상 작동 확인

Test ID : 7) 6번 테스트와 통합 내용. 정상 작동 확인

Test ID : 8) 변경 내용 정상 작동 확인

Test ID : 9) 정상 작동 확인

Test ID : 10) 정상 작동 확인

Test ID : 11) 의사의 예약 승인후 정상 작동 확인

Test ID : 12) Version2 구현 제외

Test ID : 13) Version2 구현 제외

Test ID : 14) Version2 구현 제외

Test ID : 15) Version2 구현 제외

8. 프로젝트 이슈사항

8.1 수행 중의 문제점

8.1.1. AppRTC

프로젝트를 진행하는 기간동안, AppRTC의 https 프로토콜에서 인증서가 만료되었다. 이 문제는 Version1이후에 발생한 문제이며, 테스트를 진행하다가 발견되었다. 이를 해결하기 위해서는 여러가지 방법이 있는데, 만약 이를 출시하게 된다면, 직접 서버를 운용하여 SSL 인증서를 발급받아서 진행하는 것이 안드로이드 마켓 정책에 위배되지 않는 방법이다. 하지만, 개발기간내에 이 방법을 사용할 수는 없었다. 그래서 앱 자체에서 SSL 인증서를 무시하는 방법을 선택하였고, 현재 버전에서는 정상 작동한다. 물론, 지금 이 상태로는 배포할 수 없다. 사업화가 진행된다면, 추가적으로 서버를 구입하여 직접 운용하는 방식을 선택해야한다.

WebRTC에 사용되는 인증서가 만료됨.

2 Answers

Active Oldest Votes



You are using a demo signaling server operated by an entity (Google) that does not consider this demo signaling server a production system. The certificate expired a week ago.

1



Run your own server, instructions can be found here: <https://github.com/webrtc/apprtc>



share improve this answer follow

answered Nov 11 at 15:14



Philipp Hancke

Modify target :

WebSocketRTCCClient.java
WebSocketChannelClient.java
util/AsyncHttpURLConnection.java

< Issue 1-1 >

8.1.2 RSA 암호화 쳐방전

의료법 상 어느정도 이상으로 상세한 의료데이터는 개인이 소지할 수 없다. 그래서, 쳐방전을 DB에 저장하기 위해서 의사와 환자만 가지는 고유의 Public Key와 Private Key를 이용해서 RSA 암호화를 진행하고 이를 앱 내에서 처리하려고 하였다. Version1에서는 프로토타입 개발방법을 섞어서 사용하였기에 새로운 기술에 대해서 유연하게 대처할 수 있었지만, 이번 버전에서는 프로토타입을 잠시나마 만들어 보아도, 이미 프레임이 짜여져서 거대해진 프로그램 상에 이러한 기능을 추가하는 것은 개발 기간동안 불가능하였다. 실제 우리 팀은 암호학과 관련된 책을 구입하여 선수지식을 습득하고 원리를 이해하는 것을 마쳤지만, 적용하지 못해서 굉장히 아쉽다. 이로 인해서, 암호화되지 않은 쳐방전은 소지할 수 없기때문에, 쳐방전을 환자에게 발송하고 DB에 저장되지 않는 방식을 선택하였다.

8.1.3 협업 관리

처음에는 개발을 할 때, 대부분의 테스크를 병렬적으로 각자 진행할 수 있다고 생각하였다. 하지만, 절때 그렇지 않았다. 테스트를 하기 위해서 또는 액티비티간 프로토콜을 관리하기 위해서는 대부분의 개발 과정에서 순차적인 과정을 거쳐야했다. 추가적으로 Github에서 Stash와 같은 문제점으로 인해서 많은 어려움을 겪었다. 이로 인해서 인원 한 명이 코드나 자료를 종합하여 관리하는 방식을 선택하여 해결하였다.

이러한 문제점들을 겪고 느낀점 중 가장 큰 것은 설계의 중요성이다. 솔직히 우리 팀은 아마추어다. 객체지향을 객체지향답게 사용했는지에 관해서 첫 번째 의문이 들고, 처음보는 기술에 대해서 유연하게 대처를 했는지도 의문이다. 물론 애자일 방식을 메인으로 채택하고 프로토타입 방식을 일부분 채택하기 때문에 지금의 결과를 본 것을 부정하지는 않는다. 하지만, 설계와 어긋난 부분도 몇 개 발견되고 있고, 설계 자체의 내용도 수정해야 하는 부분을 계속 봐왔고, 계속 수정해왔다. 프로그램의 사이즈가 커질 수록 설계는 굉장히 중요함을 빼저리게 느꼈고, 이번 방학동안 반드시 복습을 해야함을 깨달았다.

9. 실행 결과

실행 결과 화면은 위에서 언급하였기 때문에 생략하겠습니다.

9.1 시연 동영상 URL :

https://www.youtube.com/watch?v=f_qe5PsuE4Y

9.2 Github URL

<https://github.com/ndo04343/KNUSD-20202-09>

9.3 팀원별 활동 현황

초반에는 Github 관리를 분업을 해서 잘 진행하였는데, 후반으로 갈 수록, 바빠지고, Stash오류가 날까봐, 최대한 병렬적인 업데이트가 아닌 순차적인 업데이트를 진행하였습니다. 덕분에 코드 관리는 로컬에서 대부분 이루어지고, Kakao Talk 단체 채팅방에서 문서도 관리가 진행되었습니다. 설계, 구현에 대해서 항상 모두 같이 있을 때 진행하였음을 참고바랍니다. 그리고 한 명은 Contributor가 분명히 등록되어 있는데, 학기 초반에 업데이트를 진행할 때 분명히 Commit하였는데도 Contributor에 뜨지 않았습니다. 아래 사진을 참고바랍니다.

Update UML Diagrams

동희 authored and 김동희 committed on 27 Sep

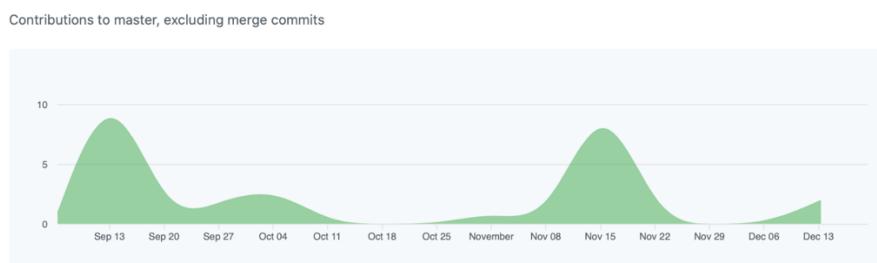
Contributors 3

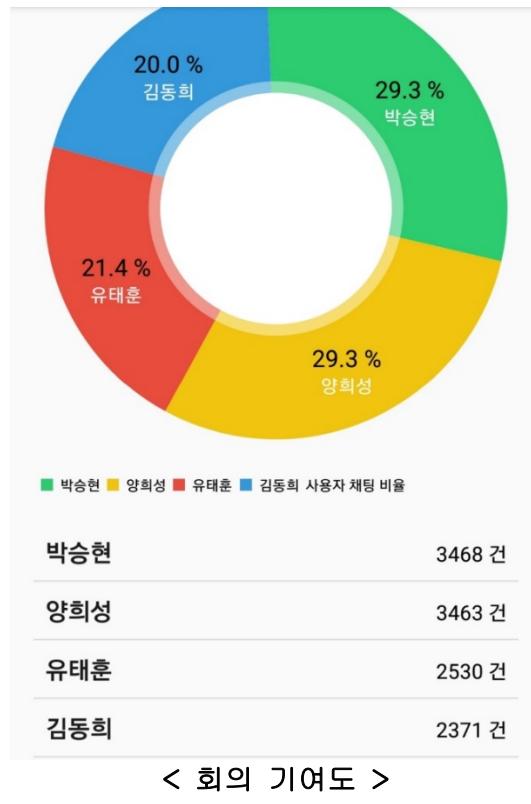
ndo04343 Heesung Yang

dbxogns77

ipod12345

이러한 이유들로 코드와 자료를 종합하는 인원인 팀장이 대부분의 커밋을 진행하였음을 알립니다. (업데이트 프로세스 통합)





9.4. 참고자료

- 1) Do it! 안드로이드 앱 프로그래밍
- 2) Firebase 공식문서 : <https://firebase.google.com/docs/android/setup>
- 3) AppRTC 공식문서 : <https://webrtc.github.io/webrtc-org/native-code/android/>
- 4) 수학으로 이해하는 암호의 원리