

모임 채팅 앱 (meeting chatting application)

본 보고서를 Project 프로젝트의 최종보고서로 제출합니다.

학과(전공): 컴퓨터 공학과
담당 교수: 장성진
팀 명: 하리
제 출 일: 0000년 00월 00일

팀 장 : 김 권 (20153211)
팀원1 : 강 재 민 (20143207)
팀원2 : 김 재 욱 (20153215)
팀원3 : 정 통 윤 (20153146)

동 의 대 학 교
컴퓨터 공학과(전공)장 귀하

요 약 문 >

<p>프로젝트 목표 (300자내외)</p>	<p>저희는 본 프로젝트를 진행하며 모임 채팅앱을 만들고자 합니다. 이 앱을 만들면서 하고자 하는 목표는 아래와 같습니다.</p> <ol style="list-style-type: none"> 1. 상호 간에 채팅이 가능한 앱을 만든다. 2. 1:1만 가능한 채팅이 아닌 여러 명에서 채팅이 가능하도록 만든다. 3. 모임 전용 통장을 개설하는 경우, 이 통장을 등록할 경우 금액을 모든 사람이 확인 가능하게 한다. 4. 모임을 가지며 만날 곳을 정할 수 있도록 GPS를 통한 주변 검색을 가능하게 한다. <p>위의 목표를 우선적으로 달성한 후 아래의 목표를 진행할 것입니다.</p> <ol style="list-style-type: none"> 1. 빅데이터를 통한 사용자의 GPS 검색 기록을 통해 자주 검색하는 장소와 비슷한 곳을 알려준다. 		
<p>내용 (500자내외)</p>	<p>저희는 본 프로젝트를 진행하며 모임 채팅 앱을 만들고자 합니다. 위의 목표를 가지고 프로젝트를 진행할 것이며, 아래와 같은 기능을 가진 앱을 만들고자 합니다.</p> <ol style="list-style-type: none"> 1. 상호 간에 채팅을 통해 대화를 할 수 있다. 이는 1:1의 대화 즉, 두 명에서만 가능한 채팅이 아닌 여러 명에서 함께 대화를 할 수 있는 채팅방을 만든다. 2. 모임 전용 통장을 개설하는 경우, 이 통장을 등록하게 되면 현재 금액을 보여주고자 하며, 이는 같은 채팅방의 멤버들이 모두 확인이 가능하다. 3. GPS기능을 이용해 주변 검색을 가능하게 하고, 이를 통해 모임을 가질 장소를 정하도록 하고자 한다. <p>위 기능들을 완벽히 구현한 뒤, GPS에 빅데이터를 추가한다.</p> <ol style="list-style-type: none"> 1. 빅데이터를 통한 사용자의 이전 GPS 검색 기록을 통해 자주 검색하는 장소와 비슷한 곳을 우선적으로 보여준다. <p>이를 통해 본 앱의 사용자들이 모임 장소를 더욱 편하게 찾을 수 있게 된다고 생각합니다.</p>		
<p>기대효과 (200자내외)</p>	<p>본 프로젝트의 기대효과는 아래와 같습니다.</p> <ol style="list-style-type: none"> 1. 현재 모임만을 위한 앱을 찾기 힘든데, 이 앱을 통해 좀 더 쉽게 접근할 수 있다. 2. 모임을 가지며 만날 장소를 찾기 위한 다른 앱의 사용을 줄일 수 있다. 3. 현재 모임을 진행하기 위해 많은 앱을 사용해야 하는데, 이를 줄일 수 있다. 4. 빅데이터의 활용이 가능해지면, 모임 장소의 검색 시간을 단축할 수 있다. 5. 모임 전용 계좌의 잔고가 얼마인지 확인하기 쉽다. 6. 후에 서로 아는 사람과의 모임만이 아닌 친목을 쌓기위한 다양한 만남의 용도로도 사용이 가능하다. 		
<p>Keywords</p>	모임	채팅	모바일 앱
	GPS	가계부	빅데이터

※ 이하 결과보고서 목차 및 내용은 예시이며, 분반별로 변경하여 활동 가능합니다.

목 차

I . 서 론	1
II . 프로젝트 수행 내용	2
1.section제목	3
1.1 sub-section 제목	4
1.2 sub-section 제목	5
1.3 sub-section 제목	6
2. section제목	7
2.1 sub-section 제목	8
2.2 sub-section 제목	9
2.3 sub-section 제목	10
III . 결 론	11
IV . 참고문헌	12
[부록]	13

1. 서론

1.1 프로젝트의 배경 및 필요성

본 프로젝트를 진행하며 저희가 만들려고 하는 것은 모임을 좀 더 편하게 가질 수 있도록 하기 위한 앱입니다.

프로젝트를 진행하기에 앞서, 저희가 만들고자 하는 앱과 비슷한 앱이 있는지 확인을 해 보았습니다. 동호회 혹은 주변 사람과의 즉석 만남을 위한 앱은 많았지만, 저희가 생각하는 방향과는 다른 앱들만 나왔습니다. 일반적으로 모임을 만들어 만남을 진행할 때, 몇 가지의 불편한 점을 아래에서 보여주고자 합니다.

1. 친구들과 만날 때, 늘 어디를 갈지 정하는 것이 힘듭니다. “만나서 정하자”라고 하지만 만나서 정하다 보면 의견을 모으기 힘들어 결국 근처에 보이는 음식점 혹은 자주 가던 음식점을 가는 경우가 많이 있습니다. 혹은 검색이나 지도 앱을 통해 음식점을 찾아보곤 하는데 이를 조금 더 편하게 찾아볼 수 있다면 편할 것이라고 생각하였습니다.

2. 저희 주변의 친구들의 경우, 모임을 하면서 하나의 공용 통장을 만들어 매달 각자 돈을 넣고, 만날 때, 이 돈을 이용하여 계산하는 것을 보았습니다. 이처럼 공용 통장을 사용할 경우, 잔액이 얼마인지 확인하는 것이 어렵습니다. 확인을 하기 위해선 가계부 앱을 이용하여 관리를 하는 것이 아니면 별도로 확인할 방법을 찾을 수가 없습니다. 물론, 사용 때마다 통장을 관리하는 사람이 사용 금액을 가르쳐 줘 확인을 하는 방법도 있지만, 이 방법도 매우 불편합니다.

3. 모임 원들 간에 연락을 취하는 경우, 카카오톡의 단체방, 네이버 밴드, 전화, 문자를 이용하는 경우가 많이 있습니다. 단체방이나 밴드의 경우, 다른 채팅 방들과 함께 이용하기 때문에 알림을 꺼두는 경우가 많이 있고 이로 인해 연락이 잘 안될 때도 있습니다. 이를 타 채팅 앱이 아닌 모임만을 위한 채팅 앱이 있으면, 서로 간에 연락을 하는 것이 더욱 편할 것이라고 생각하였습니다.

4. 위에 나온 것처럼 모임을 위해서는 현재 지도, 채팅, 가계부 등 너무 많은 앱들을 필요로 합니다. 이를 하나의 앱에서 이용 가능하게 한다면 더욱 편리해 질 것이라고 생각하였습니다.

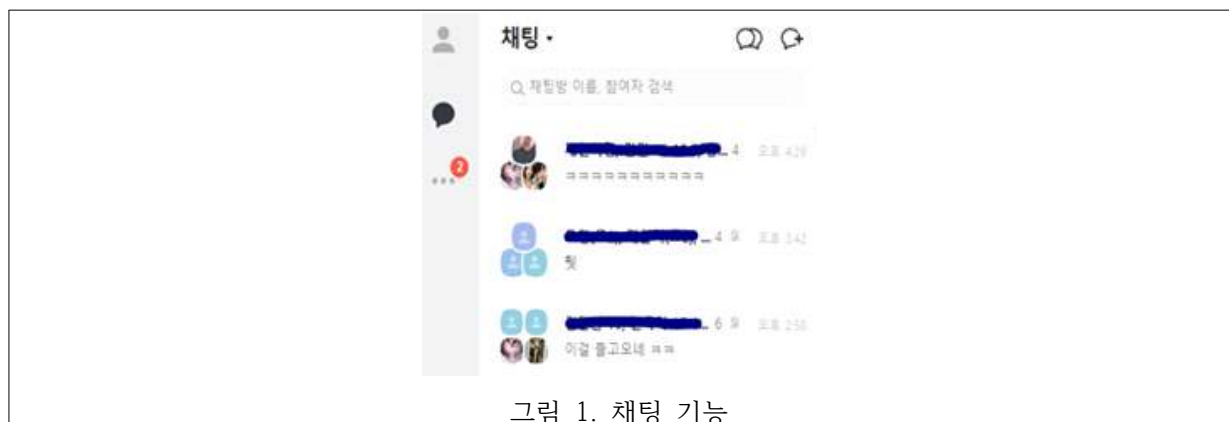
위에서 언급한 문제점들을 하나의 앱에서 다 처리할 수 있도록 하기 위해 모든 기능들을 포함한 앱을 만들고자 합니다.

2. 설계 수행 내용

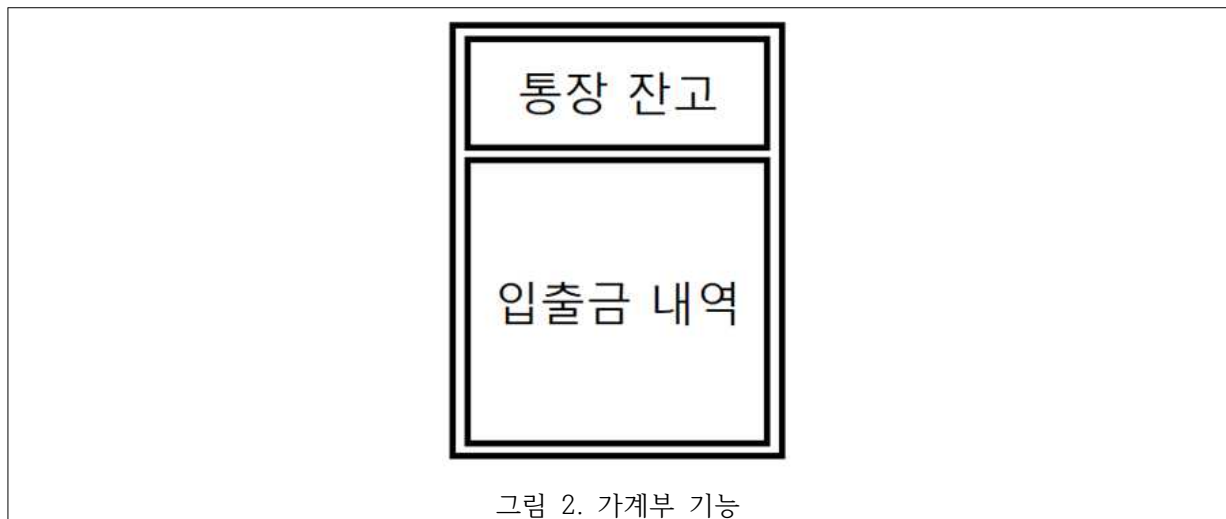
2.1 설계 과제 목표

친구들과 연락을 하거나 즉석에서 만나는 경우 사용을 하는 채팅 앱은 시중에 많이 있다. 하지만 모임만을 위한 채팅 앱이 부족하다고 우리는 생각하였다. 예를 들어 가게부를 사용하기 위해서는 채팅 앱 뿐만 아니라 가게부 앱도 설치하여 사용을 하여야 한다. 또한, 장소 검색을 하는데 있어 채팅 앱만으로 검색을 하는 데는 부족함이 있고 마찬가지로 다른 지도 앱을 이용하거나 검색 앱을 통해 검색을 하여야 한다. 한 가지 앱만으로는 필요한 기능을 처리 하는데 불편함이 있고 이러한 문제점을 해결하기 위해 모임 채팅 앱을 만들어 조금 더 사용 하는데 편하게 하고자 한다.

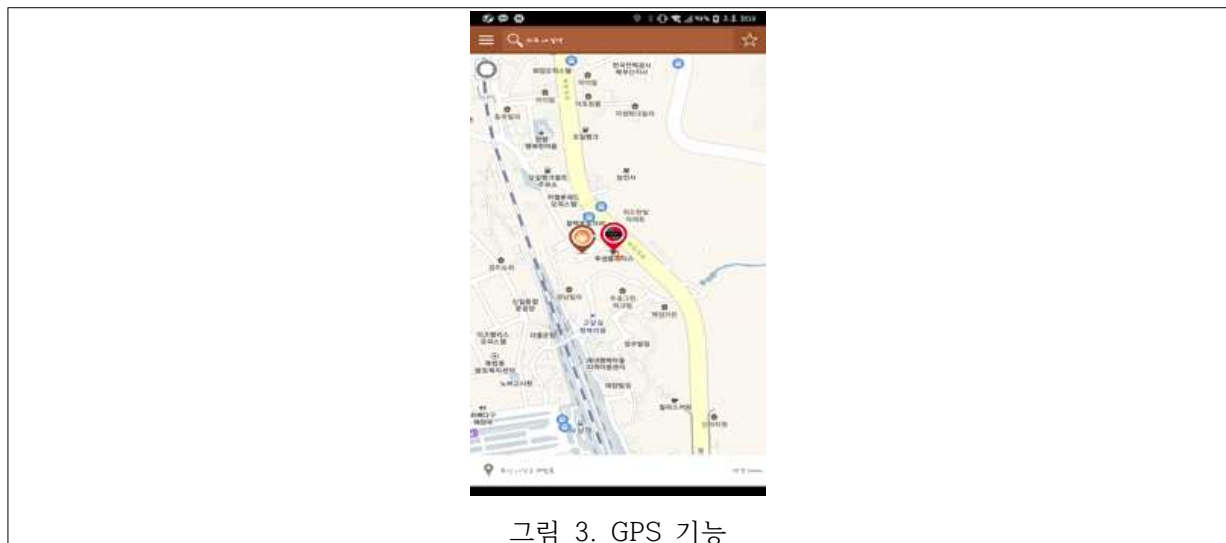
모임 채팅 앱에는 총 3가지의 큰 기능이 있다. 첫 번째는 채팅 기능이다. 자신과 친구 등록이 되어있는 경우, 채팅방을 만들어서 사람들을 초대할 수 있다. 채팅방을 초대하면 서로 채팅을 하는 것이 가능하며, 2명뿐만 아니라 여러 명을 초대하여 많은 사람들과 채팅을 하는 것도 가능하다.



두 번째는 가게부 기능이다. 이 기능은 모임 전용 통장을 만드는 사람들을 위한 기능이다. 친한 친구들과 모임을 만드는 사람들의 경우, 모임 전용 통장을 만드는 것을 가끔 볼 수 있다. 이런 경우를 위한 기능으로 카드회사와 카드번호 뒤 4자리를 등록하여 사용가능하다. 문자 동기화 기능을 추가하여 문자가 들어오는 경우 필요한 정보를 뽑아 가게부에 자동으로 등록되도록 한다. 카드회사마다 문자의 내용이 다르다는 문제점이 있어 모든 문자의 완벽한 동기화가 불가능하였다. 이 점을 위해 문자의 내용이 등록되지 않는 경우, 직접 내역을 추가할 수 있도록 하였다. 가게부 기능을 이용하면 등록된 모임의 채팅방에 한해서 현재 남은 금액을 보여준다. 한 명만 등록을 하여도 모든 사람이 같이 금액을 볼 수 있다.



세 번째는 GPS 기능이다. 채팅 앱 내에서 지도를 켜서 검색을 할 수 있으며 이로 인해 따로 지도 앱을 켜서 검색할 필요가 없다. 장소를 검색하게 되면 해당하는 장소들이 나오게 되고 선택을 하면 그 곳의 위치가 지도에 마커로 표시가 된다.



이 모든 기능들을 포함한 앱을 만드는 것을 이번 프로젝트의 목표로 한다. 이 기능들의 모든 구현이 완벽해지면 추가로 빅데이터 기능을 넣는다. 빅데이터를 활용하여 GPS 검색을 사용할 경우 검색된 내용들의 유사도를 따지고 이를 이용하여 유사한 정도로 먼저 보여주도록 하고자 한다. 다음으로 스마트폰용으로만 만들어 졌기에 스마트폰 뿐만 아닌 다른 기기, 즉 컴퓨터 용을 만들고자 한다. 스마트폰으로만 채팅이 아닌 컴퓨터 등과도 채팅이 가능하도록 하여 다양한 기기에서 이용이 가능하도록 하고자 한다.

2.2 설계 구상 및 흐름도

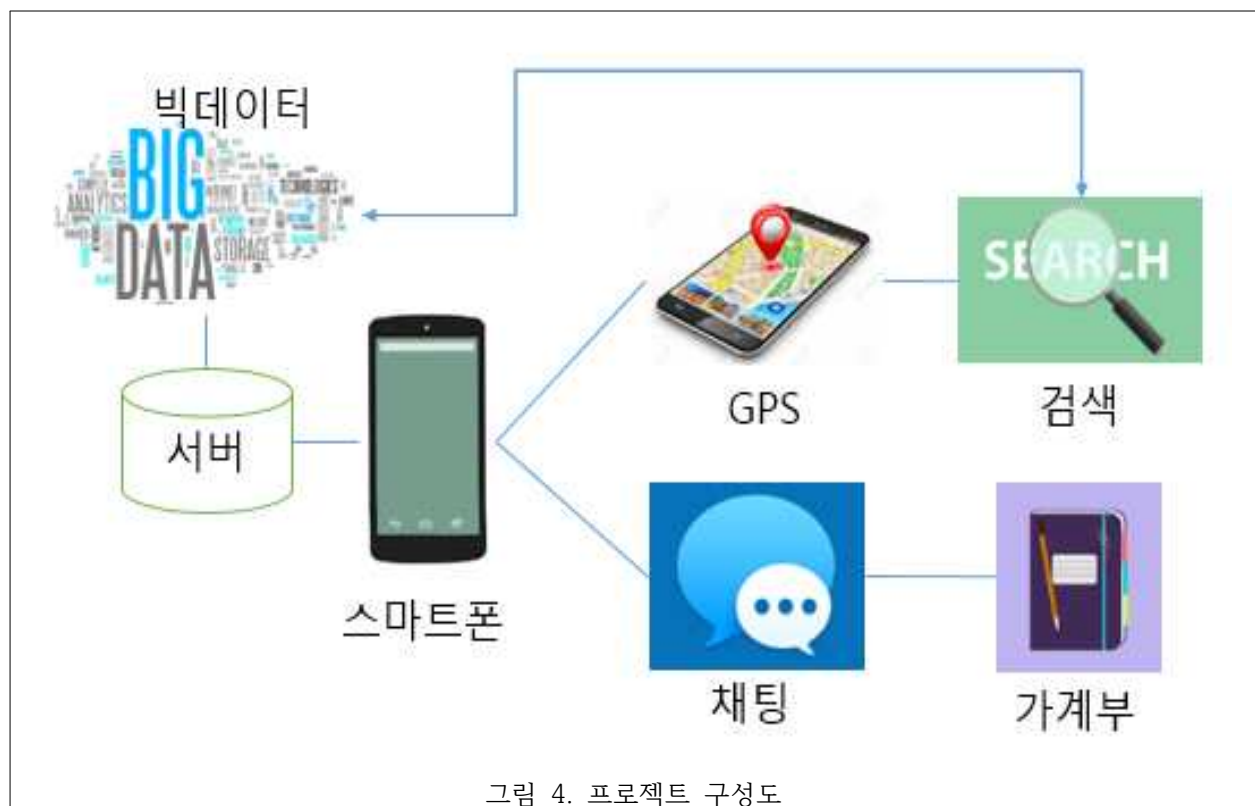
- 상호간에 채팅이 가능한 앱을 만든다.
- 둘만의 채팅이 아닌 여러 명과도 채팅이 가능한 채팅방을 만든다.
- 모임 전용 통장을 개설하는 경우, 이 통장을 등록하면 이 통장의 금액 및 내역을 채팅방

사람들이 확인가능하도록 한다.

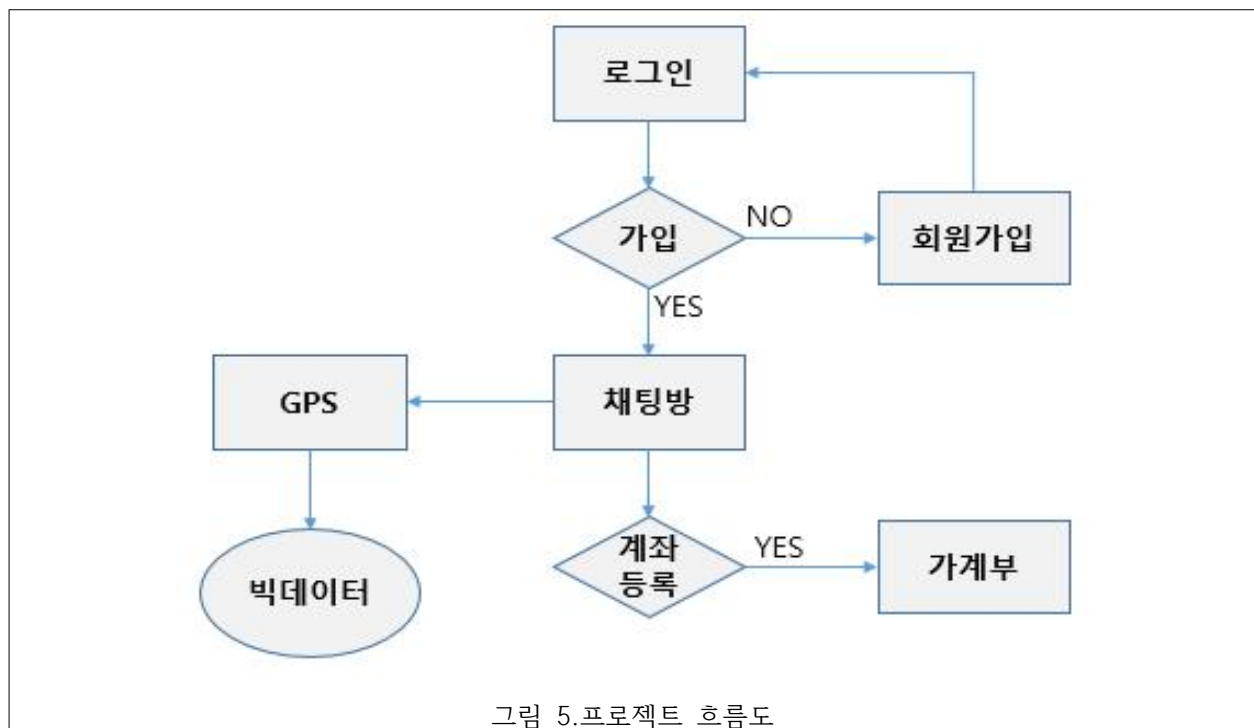
- 모임 전용 통장을 개설하는 경우, 이 통장의 사용내역이 문자로 올 경우, 문자에서 필요한 내용만 뽑아 데이터를 DB에 저장하고 이를 내역에 띄어준다.
- 모임을 가질 장소를 검색할 수 있도록 GPS를 이용한 주변 검색을 가능하게 한다.

위 목표를 우선적으로 달성한 후 아래의 목표를 진행할 것이다.

- 빅데이터를 통한 사용자의 이전 검색 기록을 통해 자주 검색하는 장소와 비슷한 장소를 우선적으로 보여준다.
- 모든 문자 양식이 등록되어 있는 것이 아니기에 직접 추가를 해주어야 하는 경우가 있는데, 이를 해결하기 위해 문자 양식을 추가로 등록한다.
- 스마트폰에서만 이용이 가능한 것이 아닌 컴퓨터용 채팅 프로그램도 만들어 서로간에 채팅이 가능하도록 한다.

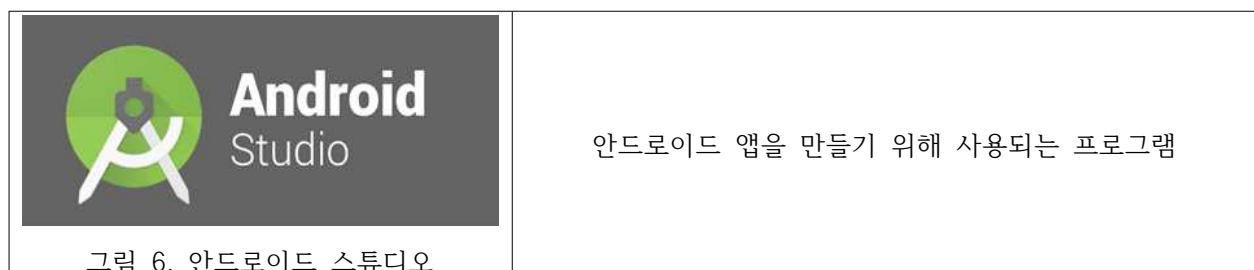


위 그림의 서버는 JSP로 구성을 하였다. 스마트폰은 서버와 연결을 하여 앱을 사용하는데 사용된다. GPS 및 채팅은 기본적인 기능이다. 채팅방에서 추가로 가계부를 등록하여 가계부 기능을 사용할 수 있다. GPS에서 검색을 하면 DB에 저장되고 빅데이터를 이용해 이를 바탕으로 검색한 장소들과 비슷한 곳을 우선적으로 보여주게 된다.



모임 채팅앱의 흐름은 위 이미지와 같다. 가장 먼저 로그인을 해야 하는데, 이때 본인이 이미 가입 되어있다면 로그인을 하여 앱의 기능들을 이용할 수 있고, 가입되어 있지 않다면 회원가입을 먼저 진행한 후 로그인을 하여 앱의 사용이 가능하다. 로그인을 완료한 후 채팅방을 개설할 수 있는데, 친구를 추가하여 채팅방에 초대할 수 있다. 채팅방을 이용하며 GPS를 선택하게 되면 지도가 나오며 자신의 주변 검색이 가능해진다. 카테고리를 검색하게 되면 해당하는 장소들의 목록이 나오게 되고 장소의 이름을 검색하면 해당 장소가 나오게 된다. 검색으로 나온 곳을 선택하게 되면 선택된 장소의 위치가 지도에 표시된다. 검색을 한 내용들이 DB에 저장되고 이를 바탕으로 많이 검색된 장소와 비슷한 곳 위주의 결과를 우선적으로 보여주게 된다. 채팅방을 이용하면서 채팅방의 사람들과 같이 모임 전용 통장을 개설한 경우 이를 등록을 할 수 있다. 이를 등록하게 되면 가계부 기능을 이용할 수 있는데, 몇 가지의 문자패턴을 미리 저장해 두었고 이에 해당하는 결제 문자를 이용한다면 추가적으로 등록을 해주지 않아도 자동적으로 내역이 DB에 저장되고 가계부 내역에도 출력이 된다. 모든 문자의 패턴이 등록된 것이 아니기 때문에, 문자내용만으로 등록이 안되는 경우가 있는데 이는 본인이 추가해주는 것이 가능하다.

2.3 사용 프로그램



ORACLE®

DATABASE

그림 7. 오라클 데이터 베이스

데이터들을 저장하는데 사용되는 프로그램



그림 8. JSP

프로그램의 서버를 구성하는데 사용되는 프로그램



그림 9. SQLite

모바일에서 사용되는 데이터베이스

3. 프로젝트 추진 계획

2.1 과제 추진 일정

설계과제 추진전략 및 방법																		
No	과제 내용	추진일정															기간 (주)	추진 방법
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
1	주제 선정	■	■														7	주제 조사 및 선정
2	관련 정보 수집			■	■	■											8	문헌 조사, 인터넷 자료조사, 관련 기관 문의
3	사용 프로그램 선택			■	■	■											4	프로젝트에 이용 될 프로그램 선정
4	로그인 기능 구현						■	■	■	■							5	로그인 기능 및 회원가입기능 구현
5	채팅 기능 구현							■	■	■	■	■	■	■	■		5	채팅 기능 및 친구 추가 삭제 기능 구현
6	지도 기능 구현							■	■	■	■	■	■	■	■		2	GPS를 이용한 검색 기능 및 주변 장소 정보 검색
7	가계부 기능 구현							■	■	■	■	■	■	■	■		3	가계부 기능 및 문자 동기화 기능 구현
8	UI 구성										■	■	■	■	■			UI 구성 및 개선
9	오류 수정															■	3	오류 수정 및 기능 보완

3.2 과제 추진체계

성명 / 학번		수행분야	역할
	김권	<ul style="list-style-type: none"> ▪ UI 구현 ▪ 가계부 기능 구현 및 설계 ▪ 전체 통괄 	팀장
	20153211		
	강재민	<ul style="list-style-type: none"> ▪ 채팅 기능 구현 및 설계 ▪ GPS 기능 구현 및 설계 ▪ 프로그램 통합 	팀원
	20143207		
	김재욱	<ul style="list-style-type: none"> ▪ 프로그램 설계 및 구현 ▪ 문자 동기화 기능 구현 및 설계 ▪ UI 구현 	팀원
	20153215		
	정통운	<ul style="list-style-type: none"> ▪ UI 구현 ▪ 가계부 기능 구현 및 설계 	팀원
	20153146		

4. 프로젝트 진행 과정

4.1 프로젝트 계획 수립

주제 선정

프로젝트를 시작하기 앞서 주제를 정하는 것이 어려웠다. 어떤 주제를 가지고 프로젝트를 진행할 것인지 결정하기 위해 각자 1가지씩의 주제를 가지고 의견을 나누었다. 의견을 나누며 결정된 주제는 지금의 주제와는 비슷하면서도 다른 카페 앱이었다. 주변 장소의 카페를 카테고리 별로 찾아볼 수 있는 앱이었는데, 독창성이 떨어진다는 의견으로 다시 한 번 주제를 선정하게 되었다. 이 후, 의견을 나누어 결정된 것이 현재의 주제 “모임 채팅 앱”이다.

관련 정보 수집

주제 선정을 통한 프로젝트 진행에 앞서 비슷한 앱이 없는지 알아보기 위해 시장조사를 진행하였다. 시장 조사 결과 우리가 생각하는 비슷한 앱이 시중에 없는 것을 확인할 수 있었다. 다음으로는 우리의 프로젝트의 필요성에 대해서 생각해 보았다. 생각해 본 결과 모임을 진행하는데 있어 너무 많은 앱을 사용해야한다는 단점을 해결할 수 있기에 우리 프로젝트가 필요하다는 결론을 내릴 수 있었다.

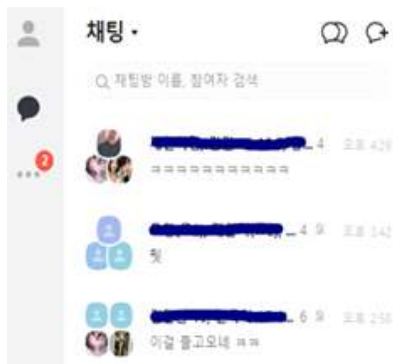


그림 14. 채팅 기능



그림 15. GPS 기능

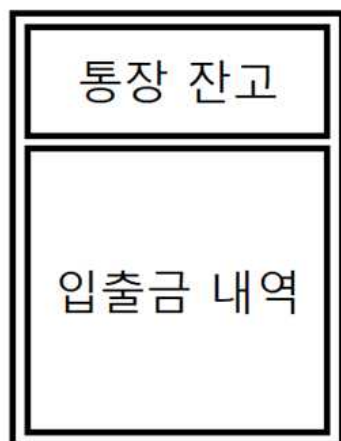


그림 16. 가계부 기능

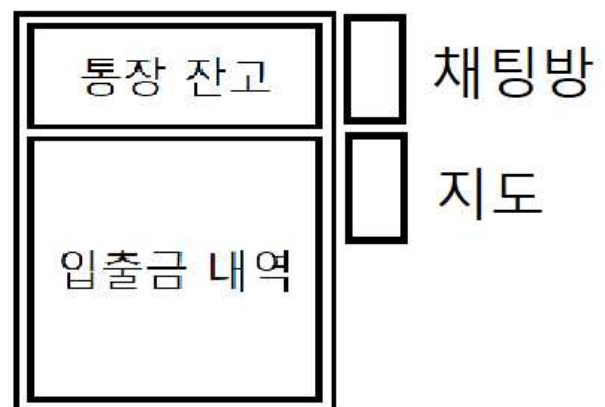


그림 17. 프로젝트 시작 구상도

사용 프로그램 선택

주제를 토대로 사용해야 할 프로그램에 대해서 생각을 해보았다. 우선 채팅 서버를 구성하기 위해 서버가 필요했다. 무슨 서버를 이용할지 고민하다 톰캣 서버를 이용하여 구성하기로 결정하였다. 그 다음으로 무슨 데이터베이스를 이용할지 고민을 하였다. 처음에는 오라클 데이터베이스를 사용하기로 결정을 하고 프로젝트를 진행하려고 하였다. 하지만 파이어 베이스를 사용하면 로그인 기능에서 조금 더 높은 완성도를 줄 수 있을 거 같아 로그인 부분에서만 파이어 베이스를 이용하기로 결정했다. 다음으로 안드로이드 스튜디오를 이용해 앱을 만들기 로 결정하여 프로젝트 구현을 진행했다.

```
정보: 프로토콜 핸들러 ["http-nio-80"]을(를) 초기화합니다.  
10월 16, 2019 3:15:26 오전 org.apache.tomcat.util.net.NioSelectorPool getSharedSelector  
정보: Using a shared selector for servlet write/read  
10월 16, 2019 3:15:26 오전 org.apache.coyote.AbstractProtocol init  
정보: 프로토콜 핸들러 ["ajp-nio-8010"]을(를) 초기화합니다.  
10월 16, 2019 3:15:26 오전 org.apache.tomcat.util.net.NioSelectorPool getSharedSelector  
정보: Using a shared selector for servlet write/read  
10월 16, 2019 3:15:26 오전 org.apache.catalina.startup.Catalina load  
정보: Initialization processed in 999 ms  
10월 16, 2019 3:15:27 오전 org.apache.catalina.core.StandardService startInternal  
정보: 서비스 [Catalina]를(를) 시작합니다.  
10월 16, 2019 3:15:27 오전 org.apache.catalina.core.StandardEngine startInternal  
정보: Starting Servlet Engine: Apache Tomcat/8.5.46  
10월 16, 2019 3:15:27 오전 org.apache.jasper.servlet.TldScanner scanJars  
정보: 적어도 하나의 JAR가 TLD를 찾기 위해 스캔되었으나 아무 것도 찾지 못했습니다. 스캔했으나 TLD가 없는 JAR들의 전체 목록을  
10월 16, 2019 3:15:28 오전 org.apache.jasper.servlet.TldScanner scanJars  
정보: 적어도 하나의 JAR가 TLD를 찾기 위해 스캔되었으나 아무 것도 찾지 못했습니다. 스캔했으나 TLD가 없는 JAR들의 전체 목록을  
10월 16, 2019 3:15:28 오전 org.apache.coyote.AbstractProtocol start  
정보: 프로토콜 핸들러 ["http-nio-80"]을(를) 시작합니다.  
10월 16, 2019 3:15:28 오전 org.apache.coyote.AbstractProtocol start  
정보: 프로토콜 핸들러 ["ajp-nio-8010"]을(를) 시작합니다.  
10월 16, 2019 3:15:28 오전 org.apache.catalina.startup.Catalina start  
정보: Server startup in 1307 ms
```

그림 18. 정상적인 서버의 동작

로그인 기능 구현

파이어 베이스를 이용하여 로그인 기능에서 조금 더 높은 완성도를 주려고 하였지만, 파이어 베이스를 사용하면 계정 관리를 하는 것이 좀 복잡해 오라클 데이터베이스를 사용하여 기능을 구현하도록 계획을 수정하였다. 맨 처음 만든 로그인화면은 아래와 같다.

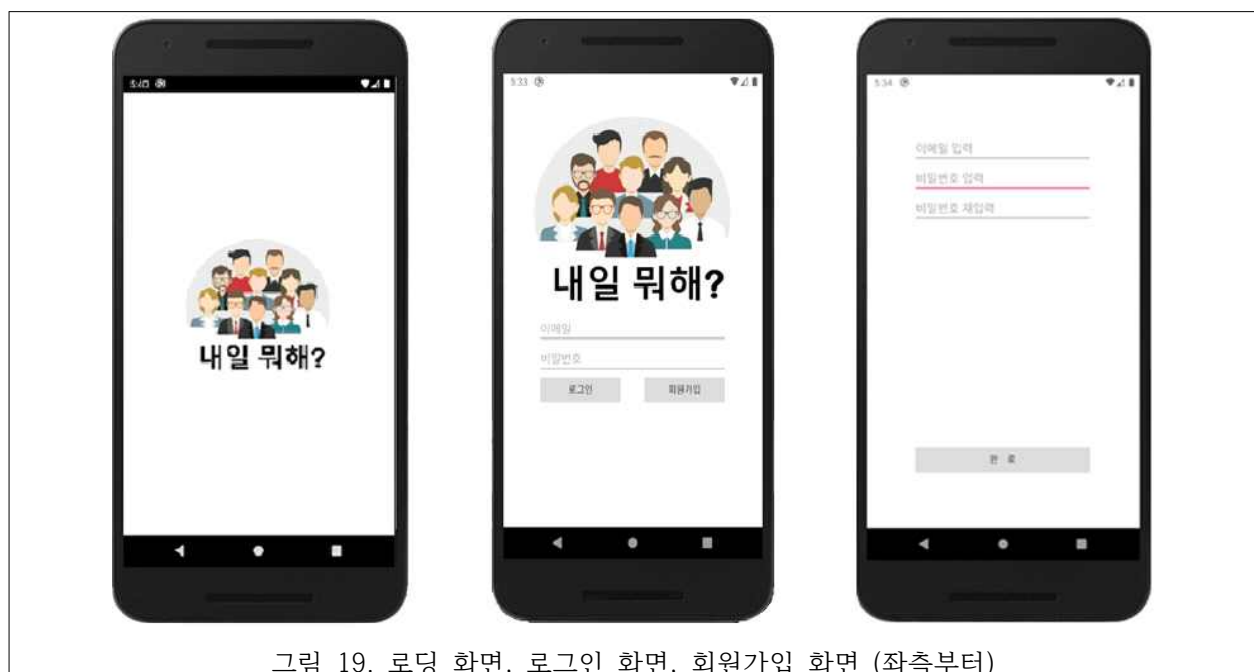


그림 19. 로딩 화면, 로그인 화면, 회원가입 화면 (좌측부터)

위 시작품을 바탕으로 프로젝트를 진행해 나가며 기능을 추가했다. 기능을 추가하며 가장 먼저 한 것은 서버에 로그인 성공 및 실패 로그를 출력해 로그인이 정상적으로 작동이 되는지를 확인하였다.

ID	PWD	NAME
1 강재민	2014	강재민

그림 20. 로그를 확인하기 위해 사용한 아이디 및 패스워드

<pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p> <pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p> <pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p> <pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p> <pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p> <pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p> <pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p> <pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p> <pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p>	<pre>select * from USERTBL where id = '강재민'and pwd = '2014'</pre> <p>성공</p> <pre>select * from USERTBL where id = ''and pwd = ''</pre> <p>실패</p> <p>그림 22. 로그인 실패</p>
---	---

그림 21. 로그인 성공

그 다음으로 회원가입 기능을 구현하였다. 회원가입을 하기 위해서 메일 확인을 받아야 하는데 이를 통해 중복가입을 방지하도록 하였다.

ID	PWD	EMAIL	AUTH
1 강재민	2014	tt@tt.t	TRUE
2 kjae0401	20142014	tt@tt.tt	TRUE
3 kang0401	alalalal	kjae0401@naver.com	TRUE
4 a3	2014	ttttt@ttt.ttt	TRUE
5 a2	2014	tttt@ttt.ttt	TRUE

그림 23. 회원가입이 완료된 아이디

다음으로 회원가입을 완료 후 로그인을 진행하는데 문제가 없음을 확인하고 채팅기능을 구현하도록 넘어갔다.

채팅 기능 구현

프로젝트의 큰 3가지 기능 중 하나인 채팅 기능을 구현하며 친구 관련 기능들도 다 구현을 하였기에 많은 시간이 들어갔다. 처음으로는 서로 친구임을 확인할 수 있는 DB를 만들었다. 서로 친구인지 확인이 되어야 하는 이유는 이를 모른다면 한 쪽에서만 친구 추가를 하였음에도 양쪽 다 친구 추가가 될 수 있기 때문이다.

MYID	FRIENDID
1 강재민	kjae0401
2 강재민	a3
3 kjae0401	강재민
4 kjae0401	a2
5 kjae0401	a3
6 a2	kjae0401

그림 24. 서로 친구임을 확인할 수 있는 DB

위처럼 DB를 구성하여 서로 간에 친구임을 확인할 수 있는 DB를 만들었다. 이렇게 만든 후 친구 추가 기능을 구현하였다. 친구 추가를 하며, 이미 친구 등록이 되어있거나 회원가입이 되어있지 않은 아이디의 경우 아무런 변화가 없다. 하지만 정상적으로 가입이 되어있고 친구가 아닌 아이디를 친구추가하면 문제없이 DB에 저장되고 친구 목록에도 생성된다.

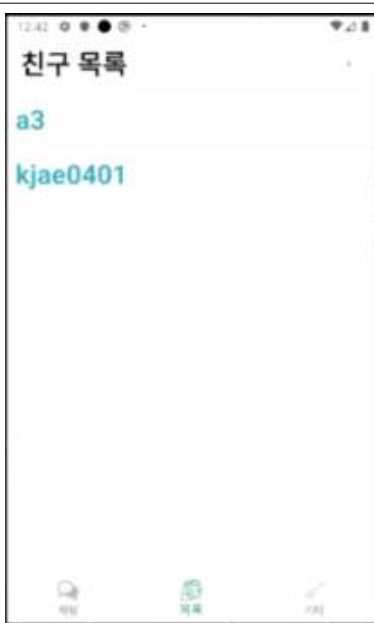


그림 25. 기존의 친구 목록

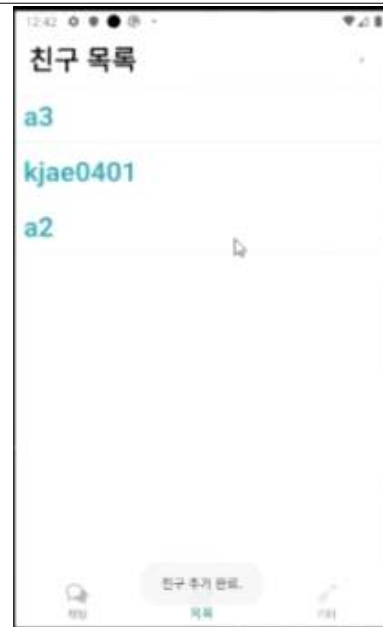


그림 26. 친구가 추가된 모습

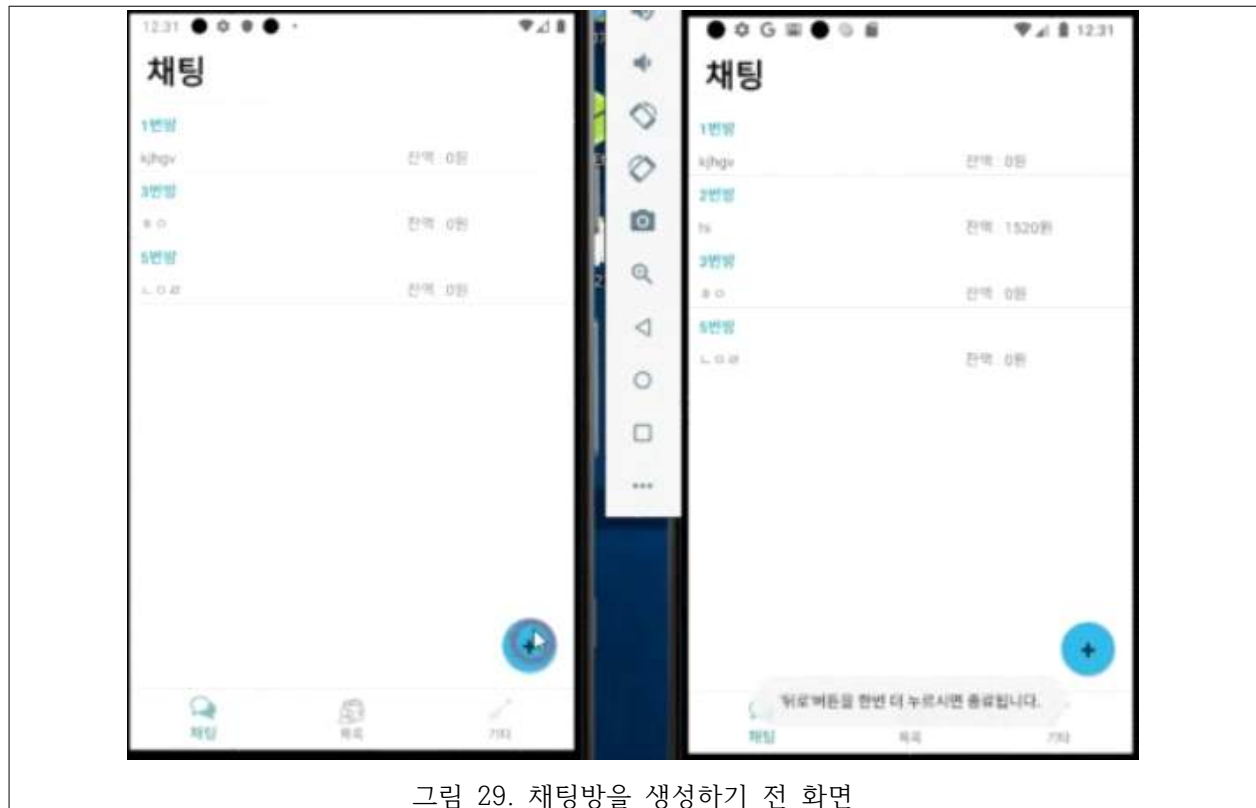
MYID	FRIENDID
1 강재민	kjae0401
2 강재민	a3
3 kjae0401	강재민
4 kjae0401	a2
5 kjae0401	a3
6 a2	kjae0401

그림 27. 친구가 추가되기 전 DB

MYID	FRIENDID
1 강재민	kjae0401
2 강재민	a3
3 kjae0401	강재민
4 kjae0401	a2
5 kjae0401	a3
6 강재민	a2
7 a2	kjae0401

그림 28. 7번에 친구가 추가된 DB

위 그림처럼 친구 추가가 완벽히 구현되고 채팅 기능의 작업을 시작하였다. 먼저 채팅방을 생성하는 작업을 하였다 채팅방 생성버튼을 누르면 채팅방에 초대할 친구를 선택할 수 있고, 친구를 선택하여 확인버튼을 누르면 채팅방이 생성된다. 채팅방을 생성하고 난 후 채팅기록이 서버로 넘어가야 하기에 채팅기록이 서버에 넘어가는 것을 확인하였다. 다음으로 진행한 작업은 채팅방에 추가의 인원을 초대하는 것이었다. 생성된 채팅방에서 추가버튼을 눌러 초대할 사람을 선택하여 채팅방에 추가의 인원을 초대하도록 하였다.



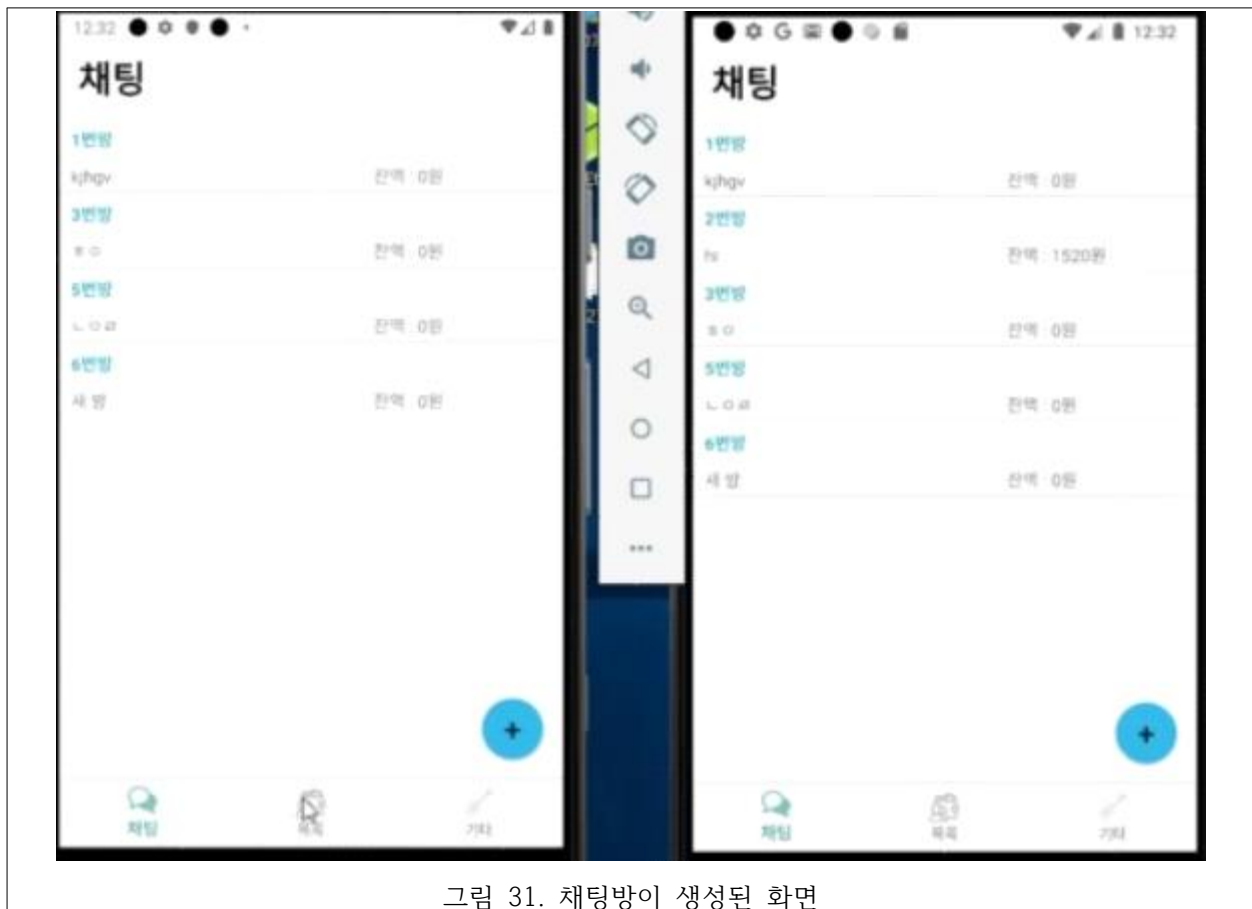


그림 31. 채팅방이 생성된 화면

채팅방을 생성하는 데 문제가 없고 서로 채팅을 주고받는데 문제가 없는지를 확인하였다.

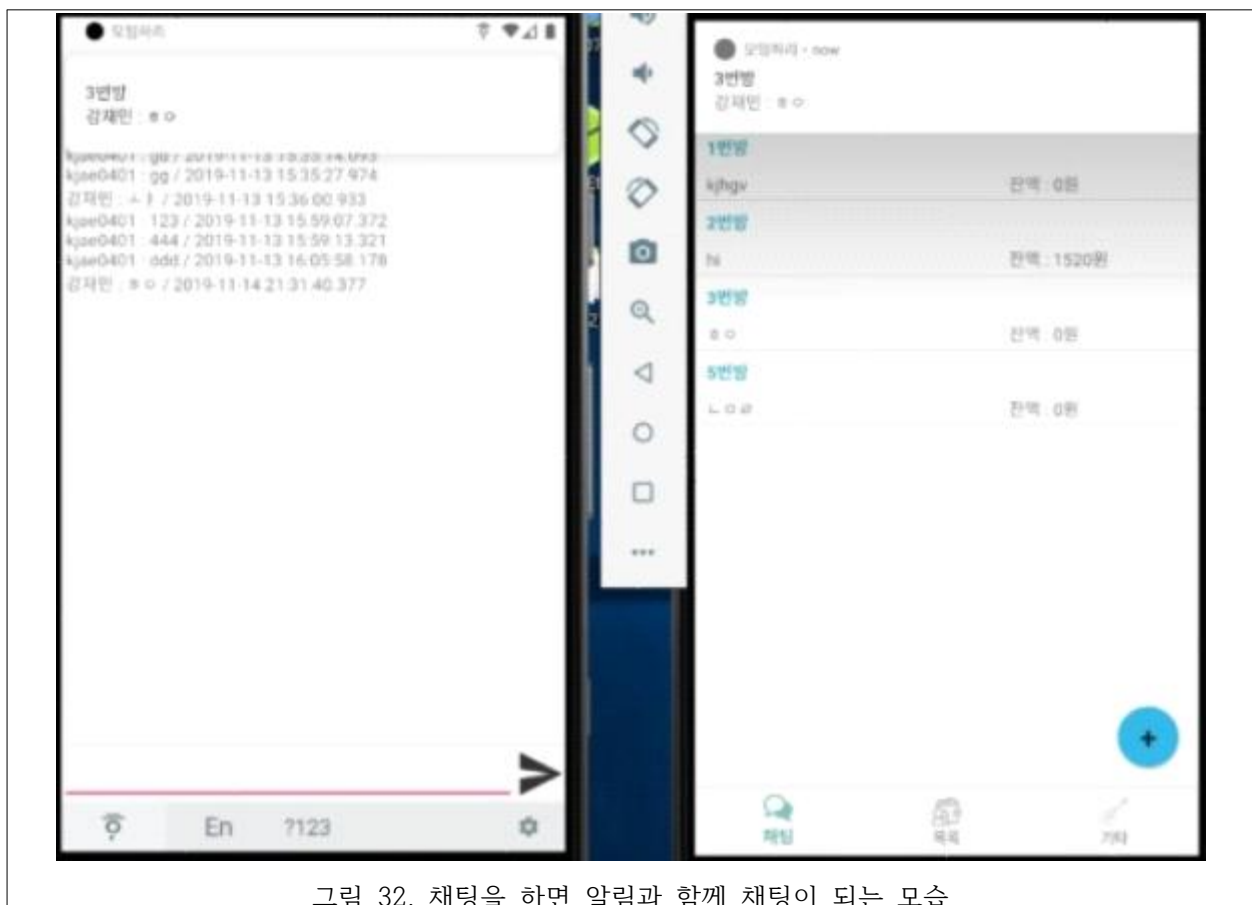
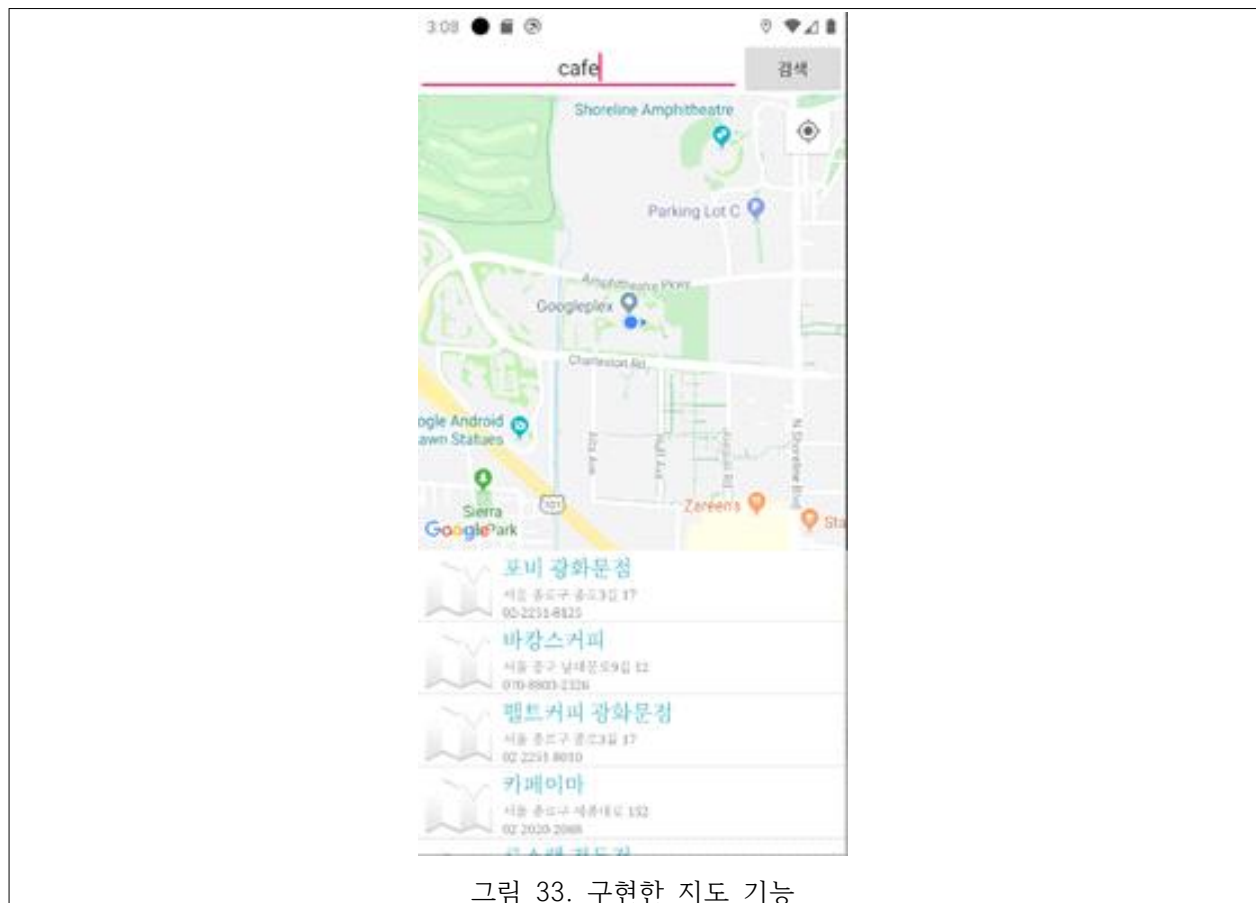


그림 32. 채팅을 하면 알림과 함께 채팅이 되는 모습

이렇게 채팅까지 가능한 것을 확인하고 GPS 기능 구현을 진행하였다.

지도 기능 구현

프로젝트의 큰 3가지 기능 중 하나인 지도 기능을 구현하기 위해 많은 방법을 시도하였다. 가장 먼저 시도한 방법은 구글 API를 이용하여 지도를 구현하는 것이었다. 하지만 이를 이용하려다보니 현재 위치에 고정되어 있고, 다른 위치의 지도를 확인하는 것에서 오류가 발생하였다. 또한 장소를 선택한 곳을 지우는데 문제가 발생되어 다른 방법을 시도하기로 하였다. 그 다음으로 선택한 방법은 네이버 API를 이용하는 것이었다. 하지만 주어진 방법을 따라하는데 오류가 발생하여 다음 방법으로 네이버 지도를 웹 크롤링하여 사용하는 것을 택하였다. 이렇게 하려고 하였으나 웹 크롤링의 특성상 내용을 불러와서 사용을 하는데 검색을 이용하는데 속도가 많이 떨어졌다. 그 후, 어떻게 할지 고민을 하다 구글의 지도를 구현하고, 네이버 검색 API를 합쳐 사용하기로 하였다. 구글의 지도를 이용하여 지도를 구현하고 네이버 검색 API를 이용하여 검색한 것을 지도에 표시해 주도록 하여 지도 기능을 구현하였다.



가계부 기능 구현

프로젝트의 큰 3가지 기능 중 하나인 가계부 기능을 구현하는데 많은 문제가 발생하였다. 가장 먼저 닥친 문제는 문자 내용을 받아와야 하는데, 휴대폰 기종 및 카드회사에 따라 문자 내용이 다 달라진다는 것이었다. 이를 해결하기 위해 자료 조사를 통해 최대한 많은 입출금 문자 내용을 찾아보았다. 이렇게 찾은 문자 내용들을 모두 패턴식으로 바꾸어 내용 중 원하는 정보만 받아올 수 있도록 하였다.

5. 결론

5.1 개선사항 및 향후 연구방향

처음 목표로 했던 모임을 위한 채팅앱의 개발은 완료했지만 더 추가하거나 개선해야 할 점들이 몇 가지 존재한다.

1. GPS로 검색한 장소의 정보를 채팅방에 공유하는 기능
2. UI의 개선
3. 가계부 문자 내용의 수집 및 추가
4. 채팅의 알림 표시
5. 빅데이터를 이용한 데이터 분석 및 결과 도출

위 5가지의 기능을 추가하거나 개선 한다면 더욱 완성된 모임 채팅앱이 될 것이라고 생각한다.

설계 목표의 중요도 및 달성도

목표	중요도(%)	달성도(%)	수행내용
정상 작동	20	20	프로그램이 작동을 하는데 문제가 없는가
채팅 기능	20	20	채팅 기능을 이용하는데 문제가 없는가
GPS 기능	20	20	GPS 기능을 이용하는데 문제가 없는가
가계부 기능	20	20	가계부 기능을 이용하는데 문제가 없는가
UI	10		앱을 사용하는데 UI가 불편함이 없는가
프로그램 동작 속도	10	10	프로그램의 동작 속도가 느리지 않는가
합계	100		