

2017년도 1학기 네트워크 프로그래밍 최종보고서

2017. 06. 19

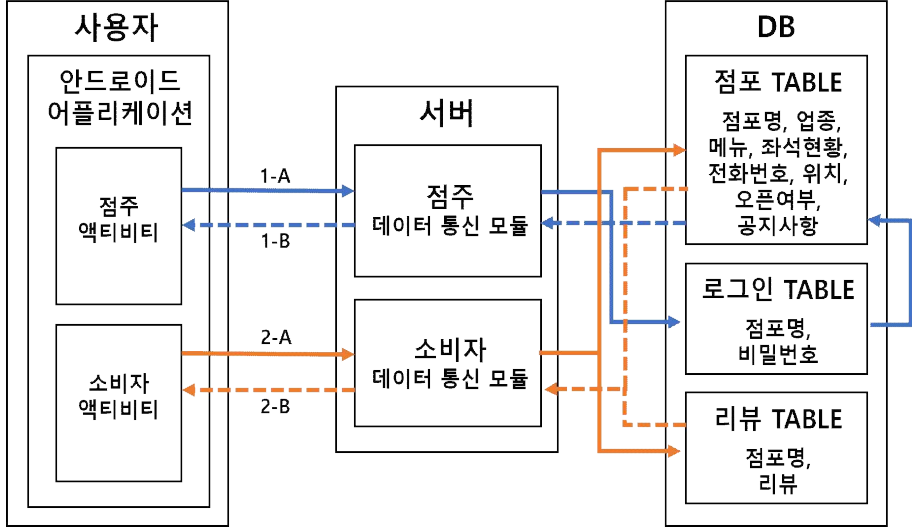
소 속	송실대학교 IT대학 컴퓨터학부
과 목	네트워크 프로그래밍
담 당 교 수	최 종 선
프로젝트 명	어서옵SSU (TCP통신 및 멀티스레드 기반 송실대 상가 정보 교류 플랫폼)

어서옵SSU				
TCP통신 및 멀티스레드 기반 송실대 상가 정보 교류 플랫폼				
분반 (00조)	가반 8조			
팀 명	월화수목금금금			
팀 원 (총: 4명)	이름	학번	개인별 역할	팀내 기여도
	박정훈	20132357	Team Leader	33 %
	최형석	20132476	Member1	33 %
	유형열	20152306	Member2	34 %
팀 구성	<pre> graph TD A["최형석 (Team Leader)"] --> B["박정훈 (Member 1)"] A --> C["유형열 (Member 2)"] B --> D["서버 개발, DB 관리"] C --> E["클라이언트 개발, 발표"] D --> F["· 멀티 스레드 기반 서버 개발 · DB 테이블 생성 및 관리 · 메시지 프로토콜 정의"] A --> G["프로젝트 관리, 클라이언트 개발, 발표"] G --> H["· 클라이언트(사장) 개발 · UI 디자인 · 메시지 프로토콜 정의"] E --> I["· 클라이언트(소비자) 개발 · UI 디자인 · 메시지 프로토콜 정의"] </pre>			

〈 개발계획서 요약 〉

팀 명		(8조) 율화수목금금금		
프로젝트 명		TCP 통신 및 멀티쓰레드 기반의 송신타 상가 정보교류 플랫폼		
배경 및 당위성		학교 인근 상가의 정보를 종합적으로 제공 받을 수 있는 시스템의 부재로 인하여 필요한 점포의 정보를 찾기 힘든 현 실태를 고려할 때, 이 플랫폼의 개발은 소비자에게 정보를 제공받을 권리를 보장할 수 있을 것이다. 더 나아가 상가의 주요 소비자인 학생들의 스터디 및 조별 과제 장소 탐색을 위해 실시간 좌석 현황을 확인할 수 있게 해줌으로써 소비자들의 헛걸음을 방지하고 학교 인근 상가의 전반적인 활성화를 도모해보고자 한다.		
제안 내용	최종 목표	본 프로젝트는 TCP 통신, 멀티쓰레드를 기반으로 하여 mysql, 안드로이드 어플리케이션을 활용해 학교 인근 상가의 점주와 소비자가 손쉽게 실시간으로 상호 교류할 수 있는 장을 제공함으로써 소비자들의 정보 제공 편의를 보장하고 상가 내 점포들의 활성화를 도울 수 있는 플랫폼을 개발한다.		
	시스템 개요	<div><p style="text-align: center;">시스템 개요</p></div> <p>안드로이드 어플리케이션에서 사용자 모드를 점주와 소비자로 구분함으로써 서버와 통신 시에 주고받는 정보의 종류를 나눠 각 요청에 따라 데이터베이스 내 데이터를 수정, 확인할 수 있게 구성한다. 주요 기능은 다음과 같다.</p> <ul style="list-style-type: none">- 점포 공지사항, 메뉴, 전화번호 등 개괄적 정보 제공- 점주의 점포 관리 기능 제공- 점포 내 좌석 정보 실시간 제공- 소비자 한줄 평 작성 기능 제공		
	개발 방법	<ul style="list-style-type: none">- mysql을 이용하여 서버 측의 데이터베이스를 구축, 클라이언트의 안드로이드 어플리케이션에 요청받은 정보를 제공한다.- 로그인 기능을 구현함으로써 확인된 점주에게 데이터베이스 수정 권한을 준다.- 어플리케이션의 점포관리 화면에서 좌석 정보를 실시간으로 수정할 수 있게 한다.- 점포 정보 확인 화면에서 소비자들이 리뷰를 작성할 수 있게끔 한다.		
기대효과		<p>학습적 측면</p> <ul style="list-style-type: none">- 강의시간에 학습한 내용을 숙지하고 TCP 기반의 소켓 통신을 활용함으로써 전반적인 네트워크 프로그래밍의 이해도 향상을 도모한다.- 서버단을 리눅스에서 구축하고, 클라이언트단을 안드로이드 어플리케이션으로 개발함으로써 서로 다른 플랫폼 상에서의 데이터 송수신을 실습해 볼 기회를 가질 수 있다. <p>실용적 측면</p> <ul style="list-style-type: none">- 송신타학교 인근 상가의 점주가 입력한 정보를 소비자에게 전달 제공함으로써 점주의 입장에서 광고효과를, 소비자의 입장에서 편리함을 얻을 수 있다.- 상가의 정보 교류 플랫폼을 제공함으로써 송신타학교 인근 상권의 전반적 활성화를 도모한다.		
중심어		점포 정보 제공	송신타 상권 정보교류 플랫폼	안드로이드 어플리케이션
		mysql	TCP 프로토콜	멀티쓰레드

〈 개발계획서 세부 〉

팀 명	(8조) - 월화수목금금금
제목	어서옵SSU
시스템 구성도	 <p>The diagram illustrates the system architecture between an Android application and a server. On the left, the '사용자' (User) section contains an '안드로이드 어플리케이션' (Android Application) with two main components: '점주 액티비티' (Owner Activity) and '소비자 액티비티' (Consumer Activity). In the center is the '서버' (Server) section with '점주 데이터 통신 모듈' (Owner Data Communication Module) and '소비자 데이터 통신 모듈' (Consumer Data Communication Module). On the right is the 'DB' (Database) section containing three tables: '점포 TABLE' (Store Table) with fields like store name, address, menu, location, phone number, and opening status; '로그인 TABLE' (Login Table) with store name and password; and '리뷰 TABLE' (Review Table) with store name and reviews. Arrows indicate data flow: 1-A (solid blue) from Owner Activity to Owner Module; 1-B (dashed blue) from Owner Module to Owner Activity; 2-A (solid orange) from Consumer Activity to Consumer Module; 2-B (dashed orange) from Consumer Module to Consumer Activity. Additionally, solid blue arrows show data being sent from both server modules to the DB tables, and dashed orange arrows show data being retrieved from the DB tables back to the server modules.</p> <p style="text-align: center;">안드로이드 - 서버 시스템 구성도</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p>1-A : 점주 Client에서 로그인하여 점포의 정보를 수정한다. 서버는 받은 로그인 정보와 DB의 정보를 비교하여 유효성을 확인한다.</p> <p>1-B : 유효한 접근일 경우 수정 요청 받은 정보를 DB에서 수정하여 그 결과를 Client UI에 적합한 형태로 전송한다.</p> <p>2-A : 소비자 Client는 서버에 특정 점포 관련 정보를 요청한다. 서버는 DB에 존재하는 점포 TABLE, 리뷰 TABLE 정보를 읽어 들인다.</p> <p>2-B : 읽어 들인 점포 정보를 서버에 전달하면 서버는 그 정보를 다시 소비자 Client UI에 적합한 형태로 바꾼 뒤 전송한다.</p> </div>
주요 SW 모듈	<ul style="list-style-type: none"> - 안드로이드 어플리케이션 : 점주와 소비자의 모드를 구분하여 서버에 필요한 정보를 요청한다. - 점주 데이터 통신 모듈 : DB내 점포의 PW를 확인하여 점주 관리모드에 필요한 통신을 주고받는다. - 소비자 데이터 통신 모듈 : 각 점포의 정보를 요청하여 수신받고, 리뷰를 작성할 수 있도록 한다. - 데이터 베이스 : 점포관련 정보, 로그인 정보, 리뷰 정보를 나누어 저장한다.
위험요소	<ul style="list-style-type: none"> - 여러 사용자가 동시에 서버에 접속하여 정보를 요청할 때, 일시적으로 오버헤드가 발생할 가능성이 있다. - 실시간 좌석 정보 최신화에 있어 점주 측의 조작이 번거로울 수 있다. - 다양한 종류의 데이터가 요구됨으로써 따른 데이터베이스 구현 및 정보 관리의 어려움이 예상된다.
위험요소의 해결 방법	<ul style="list-style-type: none"> - 이후 사용자 부하 테스트를 통해 멀티쓰레드 기능 구현 필요성을 판단한다. - 직관적인 점포 좌석도를 제공하며 한 번의 터치로 최신화가 반영되도록 구현함으로써 편리성을 보장하도록 한다. - 관계형 데이터베이스의 특성을 이용하여 개발에 요구되는 정보를 분산하여 유연하게 데이터 베이스를 설계하고 구현한다.

목 차

1. 프로젝트 개요	1
1-1. 개발 기술의 개요	1
1-2. 개발 기술의 필요성	2
1-3. 개발 기술의 이상효과 및 활용방안	2
2. 관련기술 현황	3
2-1. 관련 기술	3
2-2. 요구 분석	4
3. 수행내용 및 결과	5
3-1. 개발 목표	5
3-2. 개발 내용	5
3-2-1. 개발 내용 및 범위	6
3-2-2. 구현 내용	7
3-2-3. 개발 결과	11
4. 장애요소 및 해결방법	13
5. 개발 후기	14
5-1. 에로 사항	14
5-2. 개발 후기(소감)	15

<그림 목차>

[그림 1] 승실대 상가 정보 교류 플랫폼	1
[그림 2] 초기화면	1
[그림 3] 시스템 구성도	6
[그림 4] 메시지 프로토콜 정의	6
[그림 5] 카테고리별 분류	7
[그림 6] 메뉴 화면	7
[그림 7] 점포 정보 화면	8
[그림 8] 로그인 화면	8
[그림 9] 점포 정보 수정	8

<표 목차>

[표 1] 개발 내용	5
[표 2] 개발 방법에 따른 실험 결과	11
[표 3] 장애요소 및 해결 방법	13
[표 4] 에로 사항	14
[표 5] 개발 후기(소감)	15

1. 프로젝트의 개요

1-1 개발 기술의 개요

우리는 송실대 인근 상권이 전반적으로 활성화되지 못한 채 몇 년간 정체되어 있는 모습에 안타까움을 느꼈다. 그 결과 이러한 문제의식을 지니고 우리 학교 인근 상가에 특화된 그 자체에 대한 정보를 나눌 수 있는 공간이 필요하다는 합의점에 이를 수 있었다. 그래서 우리는 이 프로젝트를 통해 송실대 인근 상권의 전반적인 정보를 공유하는 장이 되는 플랫폼을 구현하고자 하였다.



[그림 1] 송실대 상가 정보 교류 플랫폼

이러한 기획 의도를 달성하기 위해서는 여러 가지 구성요소가 요구된다. 정보 공유 플랫폼을 목표로 하였기에 서버 측에서는 접속된 여러 클라이언트로부터 요구되는 정보에 대하여 실시간으로 즉각적인 정보를 제공해야한다. 이러한 특성을 기반으로 접속을 요청한 클라이언트에 대해 TCP 기반의 소켓 통신이 이루어지게끔, 멀티쓰레드를 이용해 서버를 구현하였다. 이 때 많은 양의 정보를 효율적으로 관리하고 전송하기 위해 관계형 데이터베이스를 이용해 필요한 정보를 적절히 분산시킬 것이 요구된다.

추가적으로 우리는 안드로이드 기반의 어플리케이션을 제작함으로써 플랫폼 이용자인 고객들과 점주 상호간의 커뮤니케이션이 활발히 이루어지도록 유도하였다. 이는 또한 플랫폼 접근성을 높여 상가 내 각종 정보를 직관적으로 조회해 볼 수 있게끔 한다. 마지막으로 정보공유의 장으로써 역할을 수행하기 위해 채팅 기능을 구현하였다. 이는 상가를 이용하면서 실질적으로 소비자들이 필요한 정보를 소비자-점주 간, 소비자-소비자 간에 주고받을 수 있도록 만들 것이다.



[그림 2] 초기화면

1-2 개발 기술의 필요성

이른바 ‘고민사거리’라는 명칭으로 대표되는 숭실대 인근 상권의 현주소이다. 이 이름은 오랜 기간 숭실대 대표 상가거리를 아울러 부르는 명칭으로 굳어져 인근 상권이 상당히 오랜 기간 침체기를 겪어왔음을 나타낸다. 우리는 학기 중 생활을 해 나가면서, 점심을 먹을 때나 팀 프로젝트를 진행할 때, 회식을 할 때 모두 적절한 장소를 찾는데 어려움을 겪어왔다. 이러한 문제의식을 가지고 머리를 맞대고 회의한 끝에 상권 내 점포들이 고르게 알려지지 못했다는 점이 가장 큰 문제라고 지적이 되었다. 산을 타고 위치한 학교 특성상 이처럼 접근성이 좋은 특정 카페, 특정 음식점의 경우만 소비자들이 찾는 탓에 작고 조용한 음식점과 카페들은 알려지지 못하는 경우가 많다고 생각이 들었다. 플랫폼 구상 과정 중에 주위의 학우 분들과 이야기를 나눠보니, 실제로 많은 사람들이 항상 찾는 점포만 가게 되는 경우가 많아, 결코 크지 않은 상가임에도 많은 음식점과 카페의 위치를 모르는 경우가 있었다. 우리는 이 과정에서 함께 대학가를 공유하는 많은 학우분들, 교직원분들, 교수님들께서도 숭실대 인근 상권에 전반적으로 만족하지 못하는 모습을 볼 수 있었다. 그 외에도 상권이 정체되어 있는 이유로 여러 가지를 추측해 낼 수 있었지만, 우리는 최종적으로 숭실대 인근 상권의 활성화를 목표로 하여 이러한 장애요인들을 딛고 문제를 해결해보고자 하였다.

1-3 개발 기술의 예상효과 및 활용방안

이 프로젝트를 통하여 우리는 숭실대 인근 상가의 총체적인 정보가 집약되는 종합 안내 시스템이자, 플랫폼 이용자 간에 활발한 커뮤니케이션이 이루어지는 정보교류의 장을 만들어내고자 하였다. 플랫폼 사용자는 스마트폰을 통해 한 눈에 필요로 하는 상가 정보에 접근할 수 있다. 이 정보들은 점포의 업종별로 제공되며 점포의 메뉴, 가격, 위치, 전화번호, 리뷰 등의 정보를 종합적으로 조회해 볼 수 있게끔 구성되어 있다. 또한 각 점포마다 별도로 운영되는 채팅기능을 통해 소비자-점주 간, 소비자-소비자 간 실시간으로 교류를 나누며 현재 좌석 정보, 영업시간 등의 정보를 주고받을 수 있도록 구성하였다. 이러한 화면 구성은 플랫폼 이용자로 하여금 숭실대 인근 상가의 정보 접근성을 매우 높여줄 것이며, 이용자가 많으면 많아질수록 더욱 신뢰할 수 있는 정보가 교류되는 장이 될 것으로 기대된다.

이 플랫폼은 단순히 상가 내 정보를 전달해 주는 것을 넘어서 상권의 활성화를 도모하기 위해 보다 넓은 관점의 서비스 제공을 목표로 하여 구현하고자 하였다. 이에 따라 우리는 플랫폼 이용자 항목에 별도로 점주를 위한 기능들을 개발하였다. 이를 통해 상가 내 점포들의 점주 분들이 플랫폼 속에서 점포 내 여러 정보를 수정·제공하고, 공지사항을 비롯하여 각종 이벤트 정보를 홍보 할 수 있도록 하였다. 이러한 종합적 관점의 서비스를 제공함으로써 이 프로젝트는 “고민사거리”라는 명칭으로부터의 탈피, 숭실대 상권의 활성화에 한 축을 담당하는 역할을 수행해 낼 수 있을 것이라 기대된다.

2. 관련 기술 현황

2-1 관련 기술

프로젝트를 진행하는 데 활용한 주요 기술들의 특징 및 목적은 다음과 같다.

○ **리눅스 환경에서의 C서버** : 본 프로젝트에서는 서버 측 구현을 위해 리눅스 환경에서 C언어로 개발을 진행한다. 구체적인 개발 환경은 ‘우분투(Ubuntu)’라는 운영체제를 사용하는데, 이는 리눅스 OS의 배포판 중 하나로 특히 데스크톱PC에서 사용할 수 있게 특화된 운영체제이다. 우분투 OS는 오픈소스로서 무상으로 이용가능하며, 이 또한 리눅스 계열 운영체제 중 하나이기 때문에 호환성이 높은 GCC(GNU Compiler Collection)를 활용한 컴파일이 가능하다. GCC는 자동 레지스터 할당, 반복되는 코드 제거, 상수 곱셈 최적화 등 매우 많은 최적화를 수행하는 컴파일러이므로 서버의 역할을 위한 많은 연산 처리에 효과적으로 사용될 수 있다.

○ **안드로이드 기반 소켓 통신** : 클라이언트 측 구현은 안드로이드 플랫폼을 활용한다. 이 때문에 안드로이드 개발용 언어인 JAVA 언어를 사용한다. 이로 인해 JAVA 언어에서 동작하는 소켓 통신 방식, 모듈을 거의 그대로 활용할 수 있다. JAVA 언어는 파일 입출력뿐만 아니라 소켓 통신 또한 STREAM을 이용하기 때문에 서버와의 통신을 위해 별도로 해줄 작업은 존재하지 않는다. 여기서 STREAM은 데이터 세트 간에서 데이터의 전송이 실행되고 있는 것으로, 문자 형식의 데이터 항목이 연속한 열로 되어 있는 것을 말한다. 즉, 단지 소켓을 생성한 후 목적에 맞는 STREAM에 소켓을 연결해주는 것만으로 서버와의 통신이 가능하다. 다만 안드로이드에서 소켓 통신을 하기 위해서는 추가적으로 메인 스레드가 아닌 보조 스레드에서 통신을 해야 한다.

○ **MySQL** : 서버에서 데이터를 관리하기 위한 목적으로 MySQL 데이터베이스 관리 시스템을 사용한다. MySQL은 리눅스 환경에서 쉽게 사용할 수 있을 뿐만 아니라 C서버와의 연동도 용이하다. 또한 다중 사용자, 다중 쓰레드(thread)를 지원함으로써 본 프로젝트의 몇몇 기능을 구현하는 데 효과적이고, 공개 소스(오픈 소스)로 개발되는 무료 프로그램이기 때문에 라이선스에 대한 비용 부담이 들지 않는 장점을 지닌다.

○ **클라이언트 UI 구현을 위한 안드로이드 디자인** : 클라이언트 측 인터페이스는 XML 코드와 JAVA 언어의 이벤트 처리를 통해 구현된다. 디자인을 함에 있어서 XML 코드를 함께 활용하기 때문에 JAVA 언어만을 통해 구현을 하는 것보다 상대적으로 직관적이고 편리하게 개발을 할 수 있다. 또한 이벤트 처리 과정은 JAVA 언어의 이벤트 처리 방식과 거의 동일하게 이루어지기 때문에 이에 대한 추가적인 학습 없이 곧바로 코드를 구현할 수 있다는 이점이 있다.

○ **TCP 통신** : 서버와 클라이언트 간 통신을 위해 TCP 방식으로 데이터를 전송한다. 프로그램의 기능을 고려하였을 때, 핵심이 되는 기능을 구현하기 위해서는 신뢰성있는 데이터 전송이 요구되는 부분들이 다수 존재한다. 또한 실제로 프로그램을 사용하며 데이터를 주고받는 과정을 고려하였을 때, 오가는 데이터의 크기가 서버에 부하를 줄 만큼 크지 않기 때문에 UDP방식 대비 데이터 송수신 시간 차이는 크지 않다고 판단하여

TCP 방식을 사용한다.

2-2 요구 분석

본 프로젝트에서 요구되는 주요 기능들은 다음과 같다.

사장 클라이언트 : 관리자 회원가입, 점포 정보 수정, 공지사항 게시, 손님들과의 채팅

고객 클라이언트 : 점포 정보 파악, 사장 및 다른 고객과의 채팅, 점포의 위치, 메뉴 등 점포 정보 확인

기본적으로 위의 클라이언트 측 사용자 인터페이스를 구현하기 위해서는 안드로이드 플랫폼이 활용된다. 디자인, 입력에 따른 처리, 이벤트 발생 등 프로그램 사용을 위한 일련의 과정들이 모두 이를 통해 구현된다.

관리자 회원가입, 점포 정보 수정/파악 등은 TCP 기반 통신을 통해 클라이언트가 서버에 요구사항을 요청한다. 이때 서버는 회원정보, 점포 정보 등의 정보를 MySQL을 통해 데이터베이스에 저장해두었다가 클라이언트의 요청에 따라 적절한 결과 값을 반환하여 준다. 뿐만 아니라 TCP방식은 사장-고객, 고객-고객 간의 채팅 기능을 구현하는 데 있어서도 중요한 역할을 한다. 채팅 기능의 특성상 원 터치 방식으로 채팅이 이루어지기 때문에 채팅 정보를 담고 있는 메시지는 1바이트에 불과하다. 따라서 데이터 송수신 과정에서 정보 손실은 큰 문제로 다가올 수 있다. 이때 신뢰성 있는 데이터 전송을 보장하는 TCP방식의 데이터 전송이 상당한 이점으로 작용한다.

3. 수행내용 및 결과

3-1 개발 목표

본 프로젝트에서는 TCP 소켓 통신과 멀티쓰레드 서버를 기반으로 해서 송실대 인근 상권의 종합적인 정보를 전달하고 이용자 간의 활발한 커뮤니케이션을 돕는 서비스를 제공하여, 침체된 상권의 활성화를 이끌 수 있는 플랫폼을 개발한다.

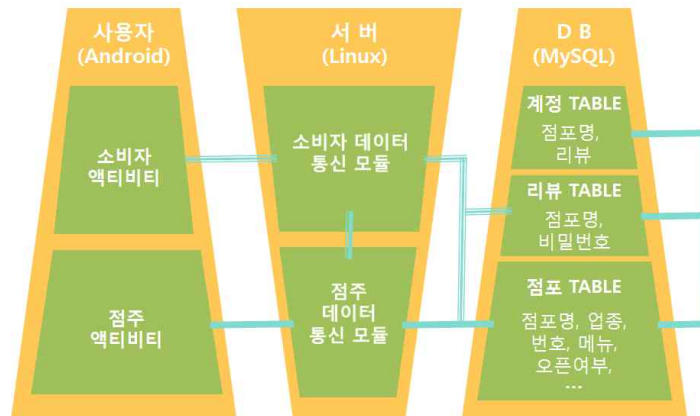
3-2 개발내용

[표 1] 개발 내용

개발 목적	개발 내용	구현 내용
서버 구현	<ul style="list-style-type: none"> 다수의 클라이언트 요청을 효과적으로 처리하기 위한 기능 클라이언트로부터 수신한 데이터 저장, 관리를 위한 기능 	<ul style="list-style-type: none"> 멀티 스레드 기반의 서버를 운용 MySQL 과의 연동을 통한 데이터 관리
클라이언트 구현	<ul style="list-style-type: none"> 점포의 정보를 확인하는 주체로서의 소비자 클라이언트 기능 연구 상권 내 정보 교류에 참가 구성원으로서 참여하는 사장 클라이언트 기능 연구 	<ul style="list-style-type: none"> 업종 별 점포 목록 제공 및 각 점포의 전화번호, 메뉴, 위치 등 세부 정보 제공 액티비티 구현 로그인을 통해 인증받은 사장에게 실시간 점포 오픈 여부 설정 가능 및 관리 점포의 세부 정보 수정기능 제공 액티비티 구현
플랫폼으로서의 기능 구현	<ul style="list-style-type: none"> 실시간 좌석 정보 기능을 제공하기 위한 수단으로서 좌석도 확인 기능 사장-고객, 고객-고객 간 다양한 커뮤니케이션을 위한 기능 	<ul style="list-style-type: none"> 실제 구현 가능성 검토 및 플랫폼 이용자 간의 높은 활용도를 위해 점포별 채팅창 별도 구현 상호 교류 활성화를 위해 점포 측의 공지사항 게시, 고객의 리뷰 남기기 기능 구현

3-2-1 개발 내용 및 범위

이 프로젝트를 통해 구현하는 플랫폼이 목표로 하는 바는 크게 두 가지로 나눌 수 있다. 첫째는 상가의 전반적인 정보 제공 플랫폼이고, 둘째는 이용자 간에 상호 교류를 촉진할 수 있는 정보교류 플랫폼이다. 이와 같은 기능을 활용하여 소비자들의 편의를 도모하고 숭실대 상권 활성화에 기여하는 것을 최종 목적으로 한다. 이러한 목표 아래 요구되는 시스템 구성도는 다음과 같다.



[그림 3] 시스템 구성도

클라이언트 부분에서 소비자와 사장 측 액티비티는 서버와의 통신과정을 거쳐 데이터베이스를 이용해 점포의 각종 정보와 리뷰, 공지사항 등을 수정, 조회해 공유할 수 있다. 또한 데이터베이스와는 별도로 각 점포마다 존재하는 실시간 채팅창을 이용하여 접속해 있는 사용자들로 하여금 상호간 직접적인 교류의 장을 제공한다. 이 때 점포의 전반적인 정보가 저장되어 있는 데이터베이스는 리눅스 서버를 거쳐 가공된 형태로 안드로이드 어플리케이션에 전달되어 사용자가 직관적으로 조회할 수 있게끔 한다. 위의 구성도와 같이 이루어진 모듈들은 정보 접근성과 정보 교류의 편리성을 최대한 높이는 방향으로 구현하기 위해 설정되었다.

구분		클라이언트 → 서버	서버 → 클라이언트
회원가입	0	길이, 점포명, PW	
로그인	1	길이, 점포명, PW	로그인 성공여부
정보 수정	2	길이, 점포명, 업종, 메뉴, ...	
정보 조회	3	길이, 점포명	업종 / 번호 / 메뉴 / ...
리뷰 수정	4	길이, 점포명, 리뷰	
리뷰 조회	5	길이, 점포명	리뷰1 / 리뷰2 / 리뷰3 / ...
채팅	6	길이, 점포명, 채팅 메시지, ...	발화자 / 인원 수 / 채팅 메시지
점포 목록 조회	7	카테고리(업종)	가게1 / 가게2 / 가게3 / ...
점포 정보 저장	8	길이, 점포명, 업종, 메뉴, ...	

[그림 4] 메시지 프로토콜 정의
각 데이터는 다음과 같이 정의된 메시지 프로토콜의 형태로 가공되어 전달된다. 본 프로젝트

의 메시지 프로토콜은 위에서 언급한 기능이 수행되는 것을 주 목적으로 하기 때문에 그 밖의 기능은 배제하였다. 예를 들어 각 점포의 채팅창에서 불 필요한 상호간 비방이 이루어지는 경우를 막기 위해 미리 정의된 특정 메시지만을 주고받을 수 있도록 하였다. 또한 직관적으로 한 눈에 파악하기 어려운 게시판 형식이 아닌 짧은 평가를 남길 수 있는 한줄평으로써의 리뷰 기능을 구현하였다.

3-2-2 구현 내용

우선 시스템 구성도를 통해 확인 할 수 있는 모듈별로 나누어 구현 내용을 설명하자면, 클라이언트 측 기능은 크게 고객 액티비티와 사장 액티비티 구현부로 나뉜다.

고객 클라이언트의 기능

○ 카테고리별 점포 정보 파악



[그림 5] 카테고리별 분류



[그림 6] 메뉴 화면

이는 고객이 각 카테고리에 어떤 점포가 있는지 확인할 수 있는 기능이다. 기능이 동작하는 과정은 크게 3단계로 구성된다. 먼저, 고객이 카테고리 화면에서 원하는 종류를 선택하면 내부에서는 서버로 데이터 전송 요청을 시도한다. 보내지는 데이터는 메시지 프로토콜 7번 형식에 맞추어 송신된다. 서버에서는 수신된 데이터 정보를 바탕으로 해당 카테고리에 존재하는 모든 가게 데이터를 다시 클라이언트 측으로 전송한다. 마지막으로 클라이언트 측은 서버에서 받은 데이터를 점포 이름 단위로 나누어 리스트에 보여주게 된다. 이렇게 해서 고객은 각 카테고리별 점포의 정보를 확인할 수 있다.

○ 점포의 위치, 메뉴 등 점포 정보 확인

이는 고객이 자신이 알고자하는 점포에 들어가 전화번호, 공지사항, 메뉴 등을 확인할 수 있는 기능이다. 이는 앞서 설명한 카테고리별 점포 정보 파악을 통해 특정 점포를 찾은 뒤 그 점포 이름을 눌러 들어오게 되는 화면이다. 기능의 동작 방식은 카테고리별 점포 정보

파악의 방식과 유사하다. 한 가지 차이점은 클라이언트가 서버로 특정 점포에 대한 정

자리 있나요?	자리 있어요	자리 없어요
오픈 했나요?	오픈 했어요	문 닫았어요
맛 있나요?	맛 있어요	그저 그데요

[그림 7] 점포 정보 화면

보 요청을 시도할 때 보내는 데이터가 다르다는 것이다. 즉, 서버로 데이터 송신시 점포에 대한 이름을 같이 보내어 서버로 하여금 특정 점포에 대한 정보만을 구분하도록 처리한다. 서버에서는 이렇게 얻어진 정보를 다시 클라이언트로 보낸다. 클라이언트가 서버로부터 받은 데이터는 2번 프로토콜에 맞게 정의되었기 때문에 점포명, 업종, 메뉴 등 각 정보를 구분하기 위한 구분자로서 '/'를 사용하였다. 이렇게 나뉘어진 각 정보를 저장해둔 뒤 각 요소에 맞게 정보를 변환해줌으로써 기능의 구현이 이루어지게 된다.

사장 클라이언트의 기능

[그림 8] 로그인 화면

[그림 9] 점포 정보 수정

서버에 사장 클라이언트들을 위해 제공되는 사장 액티비티의 경우에는 플랫폼 상에서의 정보 교류가 보다 활발하고 다양한 측면에서 이루어지도록 하는 것이 기능의 핵심 목표이다. 우선 사장 액티비티에서만 제공되는 기능들은 두 가지가 있다. 먼저 점포 관리자(사장) 회원가입 기능을 통해 0번 메시지 프로토콜과 8번 프로토콜을 차례로 전송하여 사장 이용자를 등록한다. 이와 같은 기능을 통해 사장 이용자들을 유치하고, 비밀번호를 통해 로그인하여 인증된 이용자만이 사장 액티비티에 접근 할 수 있도록 제한한다. 로그인은 1번 프로토콜로 입력된 점포명과 관리자 비밀번호를 전송하고, 유효한 값인지 여부를 서버에서 한 바이트를

받아와 확인한다. 이렇게 인증된 이용자의 경우에는 점포 관리 화면으로 접근 할 수 있는데, 이 화면에서는 점포 내 각종 정보를 조회하고 수정할 수 있도록 한다. 점포의 메뉴, 오픈 여부, 공지사항과 같은 이 정보들은 이 화면에서만 수정이 가능하다. 이는 2번 메시지 프로토콜을 통해 이루어진다.

또한 플랫폼을 구현하는데 있어 핵심적인 역할을 수행하는 부분으로 정보교류의 활성화를 위한 기능들이 있다. 우선 채팅창을 각 점포마다 구현함으로써 실시간으로 접속해 있는 이용자 간에 필요한 정보를 나눌 수 있도록 하였다. 이는 즉각적인 교류의 수단이 됨으로써 상호 호혜적으로 정보를 공유할 수 있는 기본 바탕이 될 것이다. 채팅을 위한 쓰레드의 구현 및 이용과 함께 6번 프로토콜을 통하여 채팅 데이터를 실시간으로 송수신 할 수 있도록 구현하였다. 또한 점포 별로 존재하는 리뷰 기능이 있다. 이 기능은 각 점포의 데이터베이스에 기록되어 다른 이용자들이 해당 점포에 대해 남긴 짧은 평가를 확인 해 볼 수 있게끔 한다. 4번과 5번 메시지 프로토콜을 이용한 이 기능은 앞서 말한 채팅 기능에 더하여 시간에 구애받지 않고 보다 많은 이용자들이 상호간 정보를 공유할 수 있게끔 해 줄 것이다. 마지막으로 앞서 점포 관리 액티비티에서 이용된다고 설명하였던 2번 프로토콜을 이용해 사장 이용자가 게시하는 공지사항이 있다. 이 기능은 사장이 직접적으로 고객 이용자에게 점포 내 각종 공지와 이벤트 정보를 전달, 홍보 할 수 있게끔 해줌으로써 보다 넓은 다양한 방식의 상호 교류가 이루어 질 수 있도록 할 것이다.

DB와 연계된 리눅스 서버 측 기능은 다음과 같다.

○ 회원가입

클라이언트의 회원가입 요청을 처리하는 기능이다. 먼저 서버는 클라이언트의 회원가입 요청에 대해서 이를 처리할 수 있는 함수를 호출한다. 그 후 이 함수의 내부에서 클라이언트가 작성한 id와 password를 입력 받고, DB에 접속해 각각을 사용자의 정보를 기록하는 user TABLE에 저장한다.

○ 로그인

클라이언트의 로그인 요청을 처리하는 기능이다. 먼저 서버는 클라이언트의 로그인 요청에 대해서 이를 처리할 수 있는 함수를 호출한다. 그 후 이 함수의 내부에서 클라이언트가 작성한 id와 password를 입력 받고, DB에 접속해 각각을 user TABLE에서 검색한다. 최종적으로 각 정보가 user TABLE에 존재할 경우 로그인을 승인하고, 존재하지 않을 경우 로그인을 승인하지 않는다.

○ 가게 목록 조회

서버 측에서 고객 클라이언트에게 고객이 선택한 유형의 입장 가능한 가게 목록을 전송하는 기능이다. 먼저 서버는 클라이언트의 가게 목록 조회 요청에 대해서 이를 처리할 수 있는 함수를 호출한다. 이후 이 함수의 내부에서 클라이언트로부터 가게의 유형을 입력 받고, DB에 접속해 해당 유형의 가게 정보를 store TABLE에서 검색한다. 해당 유형의 가게가 있을 경우 차례대로 가게 목록 정보를 버퍼에 저장하고 검색이 끝나면 버퍼에 저장되어 있는 정보를 클라이언트에게 전송한다.

○ 가게 정보 저장

사장의 가게 정보 저장 요청을 처리하는 기능이다. 서버는 클라이언트의 가게 정보 저장 요청에 대해서 이를 처리할 수 있는 함수를 호출한다. 이어서 이 함수의 내부에서 클라이언트의 로그인 아이디와 해당 가게 정보를 입력받고, DB에 접속해 로그인 아이디와 일치하는 가게의 이름을 **store TABLE**에서 검색한다. 그 후 일치하는 정보가 있는 해당 행에 가게 정보를 저장한다.

- 가게 정보 조회

클라이언트의 가게 정보 조회 요청을 처리하는 기능이다. 기능을 수행하기 위한 과정을 나타내면, 먼저 서버는 클라이언트의 가게 정보 조회 요청에 대해서 이를 처리할 수 있는 함수를 호출한다. 이어서 이 함수의 내부에서 클라이언트로부터 가게 이름을 입력 받고, DB에 접속해 가게 이름을 **store TABLE**에서 검색한다. 그 후 해당 가게의 이름이 **store TABLE**에 존재하는 경우 해당하는 행의 가게 정보를 읽어 클라이언트에 전송한다.

- 가게 정보 수정

클라이언트의 가게 정보 수정 요청을 처리하는 기능이다. 서버가 클라이언트의 가게 정보 수정 요청을 처리하는 과정은 다음과 같다. 먼저 서버는 클라이언트의 가게 정보 수정 요청에 대해서 이를 처리할 수 있는 함수를 호출한다. 그 뒤 이 함수의 내부에서 클라이언트로부터 가게 이름과 수정을 원하는 정보를 포함한 가게의 정보를 입력받고 DB에 접속해 해당 가게의 이름을 **store TABLE**에서 검색한다. 해당 가게의 이름이 **store TABLE**에 존재하는 경우 해당하는 행에 각각의 가게 정보를 알맞게 저장한다.

- 리뷰 저장

클라이언트가 작성한 리뷰를 가게 이름에 알맞게 DB에 저장하는 기능이다. 먼저 서버는 클라이언트의 리뷰 저장 요청에 대해서 이를 처리할 수 있는 함수를 호출한다. 그러면 이 함수의 내부에서 클라이언트로부터 가게 이름과 저장을 원하는 리뷰 정보를 입력받고 DB에 접속해 해당 가게의 이름을 **review TABLE**에서 검색한다. 그래서 해당 가게의 이름이 **review TABLE**에 존재하는 경우 해당하는 행에 리뷰 정보를 저장한다.

- 리뷰 조회

클라이언트의 리뷰 조회 요청을 처리하는 기능이다. 먼저 서버는 클라이언트의 리뷰 저장 요청에 대해서 이를 처리할 수 있는 함수를 호출한다. 그러면 이 함수의 내부에서 클라이언트로부터 가게 이름을 입력받고 DB에 접속해 해당 가게의 이름을 **review TABLE**에서 검색한다. 그래서 해당 가게의 이름이 **review TABLE**에 존재하는 경우 해당하는 행의 리뷰 정보를 버퍼에 저장하고 그 버퍼의 내용을 클라이언트에 전송한다.

- 채팅

특정 가게에 입장한 클라이언트가 채팅을 입력했을 때 해당 가게에 입장 상태인 모든 클라이언트에게 입력한 채팅을 전송하는 기능이다. 서버가 클라이언트의 채팅 요청을 처리하는 과정은 다음과 같다. 먼저 서버는 클라이언트의 채팅 요청에 대해서 이를 처리할 수 있는 함수를 호출한다. 그러면 이 함수의 내부에서 클라이언트로부터 가게의 이름과 채팅관련 정보를 입력받고 해당 가게에 입장 상태인 클라이언트에게만 선별적으로 채팅 정보를 전송한다.

3-2-3 개발결과

[표 2] 개발방법에 따른 실험 결과

구 분		개발방법	개발 결과
채팅 기능		접속되어 있는 클라이언트들이 DB를 거치지 않고 해당 점포의 채팅 화면에 메시지를 주고받을 수 있도록 구현하였다.	각 점포화면마다 개별적으로 채팅창을 구현하였으며, 각 채팅 메시지는 6번 메시지 프로토콜을 이용하여 전달된다. 서버에서는 해당 메시지 프로토콜을 수신할 시 communication() 함수를 호출하여 해당 작업을 수행한다.
클라이언트	고객님	고객이 상가 내 점포의 다양한 정보를 직관적으로 조회하도록 하기 위해서 여러 액티비티를 이용하여 구성하였다. 이를 위해 점포의 리스트를 제공하기 위한 액티비티를 통해 개별 점포의 정보가 담긴 액티비티에 진입할 수 있게끔 하였다. 또한 많은 내용이 담겨있는 리뷰와 메뉴에 대하여 별도로 액티비티를 구성해 전체 화면으로 해당 정보를 조회할 수 있도록 하였다.	업종별 점포 분류를 위한 CategoryActivity 및 각 하위 업종에 해당하는 Activity, 개별 점포의 정보를 조회하는 StoreDataActivity, 리뷰와 메뉴를 조회하기 위한 ReviewActivity 및 BuyerMenuActivity를 유기적으로 연결하여 구현
	사장님	기본적인 구조는 고객의 경우와 동일하나 점포 관리자로서 점포의 정보를 수정, 관리할 수 있도록 액티비티를 구성했다. 또한 임의로 리뷰를 등록할 수 없도록 별도의 리뷰 액티비티를, 메뉴와 가격을 수정할 수 있도록 메뉴 액티비티를 구현하였다. 인증된 이용자만 점포의 정보를 수정, 관리할 수 있도록 회원가입 및 로그인 절차 구현	점포 관리 화면으로 진입하는 UpdateStoreActivity, 리뷰와 메뉴를 조회하기 위한 SellerReviewActivity 및 SellerMenuActivity를 유기적으로 연결하여 구현 사장 이용자의 가입을 위한 SignUpActivity와 초기 화면 진입시 비밀번호를 입력받아 이용자 인증을 확인하기 위한 LoginActivity 구현
서버	회원가입	클라이언트로부터 전송받은 id와 password를 user TABLE에 저장한다.	join()
	로그인	클라이언트로부터 전송받은 id와 password를 user TABLE의 정보와 비교한다.	login()
	가게 목록 조회	store TABLE에 저장되어 있는 가게 목록을 클라이언트에 전송한다.	loadList()

구 분		개발방법	개발 결과
DB	사용자	<p>다음의 문장을 입력하여 user 테이블을 생성한다.</p> <pre>CREATE TABLE user(id VARCHAR(50) password VARCHAR(50));</pre>	user TABLE 생성
	가게 정보	<p>다음의 문장을 입력하여 store 테이블을 생성한다.</p> <pre>CREATE TABLE store(name VARCHAR(50), type VARCHAR(1), number VARCHAR(50), menu VARCHAR(8000), open VARCHAR(1), notice VARCHAR(1000), location VARCHAR(8000));</pre>	store TABLE 생성
	리뷰	<p>다음의 문장을 입력하여 review 테이블을 생성한다.</p> <pre>CREATE TABLE review(name VARCHAR(50), review VARCHAR(8000));</pre>	review TABLE 생성

4. 장애요소 및 해결방법

[표 3] 장애요소 및 해결방법

장애요소 항목	장애요소 내용	해결방법
Byte Order 변경 문제	정수형 데이터의 경우, 클라이언트 측에서 전송한 값을 서버에서 제대로 읽지 못한다.	자바와 리눅스는 각각 빅엔디안과 리틀엔디안 방식을 따른다. 그렇기 때문에 자바 기반의 클라이언트와 리눅스 기반의 서버 사이에서 정수형 데이터를 송수신할 때에 서버 측에서 <code>htonl()</code> 함수를 통해 송수신할 데이터의 바이트 순서를 변경했다.
채팅 기능을 구현하는 과정에서 발생한 문제	채팅 기능을 구현하는 과정에서 간헐적으로 데이터 손실이 발생했다.	문제 원인을 조사해본 결과, 서버 측에서는 계속해서 각 클라이언트별로 개별적인 스레드를 구성해주는데, 클라이언트 측에서는 하나의 데이터 STREAM만 가지고 채팅을 시도하였다. 이 때문에 서버에서 스레드를 생성하는 부분에 문제가 발생했다. 이를 해결하고자 클라이언트 측에서 채팅을 시도할 때 채팅 정보를 보낼 때마다 매번 데이터 STREAM을 구성하고 종료하는 방식으로 바꾸었다.
멀티스레드에서의 임계영역 처리 문제	멀티 스레드 기반의 서버에서는 동일한 메모리 영역으로의 동시접근이 발생할 수 있다.	Mutual Exclusion을 이용하여 해결했다.
자바와의 통신에서의 캐리지 리턴 처리 문제	클라이언트 측에서 <code>BufferedReader</code> 의 <code>readLine()</code> 을 사용하여 서버로부터 데이터를 전송받도록 했지만 전송이 제대로 되지 않았다.	<code>readLine()</code> 은 수신 데이터에 개행문자가 포함되어야 한다. 자바에서의 개행문자는 " <code>\n</code> "이지만, 스트림에서의 개행문자는 " <code>\r\n</code> "이므로 서버에서 송신할 데이터의 마지막 부분에 " <code>\r\n</code> "을 추가하여 이를 해결했다.
안드로이드 스레드 소켓 통신 권한 문제	클라이언트 측 메인 스레드 내에서 소켓 통신이 되지 않았다.	네트워크 사용은 네트워크의 상태와 서버의 사정에 따라 소모되는 시간을 유추하기 힘들고, 시간이 많이 소모될 확률이 크다. 그렇기 때문에 API 10부터는 기본적으로 메인 스레드의 네트워크 사용을 강제적으로 막고 있다. 그렇기 때문에 메인 스레드 외의 스레드를 생성하여 소켓 통신을 했다.
안드로이드 스레드 UI 수정권한 문제	클라이언트 측 스레드에서 UI를 수정할 수 없다.	이를 해결하기 위한 방법으로 찾은 것이 3가지가 있었는데 그 중에서 <code>runOnUiThread()</code> 를 사용하여 해결했다.

5. 개발 후기

[표 4] 에로 사항

에로사항	개발하면서 가장 어려웠던 점에 대한 극복방법
개발 일정을 세우는 데 대한 막연함	소프트웨어 공학적 접근법을 이용해 일정을 세분화하였다. 대형 과제를 수행하는 데 주로 사용되는 Agile Process를 본 프로젝트에 적용해 봄으로써 향후 현업에서의 프로젝트 방식에 대해서도 미리 체험해보고 이 같은 방식이 갖는 이점을 심분 활용하였다. 기획, 설계, 분석, 구현, 테스트, 완료에 이르기까지 전체 프로세스를 약 2주 조금 못 미치는 기간을 두고 반복하여 시간을 효과적으로 사용하며 프로그램의 퀄리티를 높이는데 기여하였다.
가상 머신에서 생성한 서버로 클라이언트가 접속하지 못함	우분투 서버 개발을 위한 환경은 가상 머신인 VirtualBox를 이용했다. VirtualBox에서 기본적으로 네트워크는 NAT로 설정되어 있다. 여기서 NAT란 사설 IP주소를 공인 IP주소로 바꿔주는데 사용하는 통신망의 주소 변환기이다. 문제가 되는 점은 사용자들의 고유한 사설망을 침입자들로부터 보호한다는 특성이다. 이로 인해 서버를 실행하여도 클라이언트 측에서는 서버로의 접속이 불가능하였다. 이를 해결하기 위해 네트워크 설정을 NAT에서 브리지 어댑터로 변경하였다. 브리지 어댑터는 게스트OS가 독립적인 네트워크 IP를 갖게 해주는 것으로, 이를 활용하여 외부로부터의 접속이 가능하게 하였고 결과적으로 문제를 해결할 수 있었다.
MySQL에 한글 입력 시, 입력과 출력이 제대로 되지 않음	MySQL은 기본적으로 언어설정이 latin1으로 되어 있다. latin1은 1바이트를 지원하기 때문에 2바이트인 한글을 저장하면 그 값이 제대로 저장되지 않는다. 그러므로 한글의 출력 또한 제대로 되지 않는 것이다. 이는 언어설정을 utf-8으로 변경하여 해결했다.

[표 5] 개발 후기(소감)

팀원	개발 후기 (소감)
최형석	<p>프로젝트 진행시 마다 항상 느끼지만 밑그림부터 시작하여 무언가를 우리의 힘만으로 구현한다는 것은 정말 막막한 기분이 드는 일이다. 이번 프로젝트 역시 아이템 기획 과정에 있어 두 번이나 큰 방향을 바꾸는 등 험난한 과정을 거쳐왔으나, 팀원간에 원활히 의견을 주고받는 과정이 있었기에 목적의식에 충실한 기획, 기획에 충실한 개발을 할 수 있었다고 생각한다. 물론 프로젝트를 기획하며 예상하지 못했던 많은 부분에서 어려움을 겪어 오랜 시간이 소요 되었지만 그럼에도 오히려 이러한 어려움을 해결해 나가며 기획에 맞게 결과물을 충실하게 구현해 낼 수 있었던 것 같아 우리 스스로에 대해 뿌듯하다. 부족한 역량을 지니고 시작되었기에 발생하는 문제들을 직접 몸으로 부딪히고 배우느라 힘든 점도 많았으나, 혼자가 아닌 함께였기에 결국 이렇게 프로젝트를 마무리 지을 수 있었다. 함께해준 팀원들에게 다시 한번 감사를 전하고 싶다.</p>
박정훈	<p>네트워크 과목에서 진행한 팀 과제는 지금까지 진행한 팀 과제에 비해 진행 과정에서 예상치 못한 문제들이 많이 발생하였다. 이러한 문제들을 팀원들과 함께 해결하는 과정에서 지금까지 알고 있었던 내용을 실습하여 숙지할 수 있었고 몰랐던 내용에 대해서는 새롭게 접해볼 수 있는 기회가 되었다.</p>
유형열	<p>그동안 소규모이지만 나름 여러 번 프로젝트를 경험해보았다고 생각했는데 이번 프로젝트와 같이 시간을 많이 쏟으며 개발을 진행한 적은 처음이었다. 그리고 결과물은 만족스럽게 나왔다고 자부한다. 본 강의 타 강의에서 배운 내용을 활용하기도 하고 많은 검색을 통해서 알게 된 내용도 프로젝트에 적극 반영하였다. 첫 발표 때 구상한 기획 내용과 거의 흡사하게 결과물이 나오게 된 점은 스스로도 칭찬할만한 점이 아닐까 생각한다. 또한 팀원들과의 호흡도 이번이 가장 좋았다고 단언할 수 있다. 각자 맡은 부분을 성실히 수행해주어 기간 내에 프로젝트를 마칠 수 있었다. 다시 한 번 팀원들의 수고에 깊은 감사를 보내고 싶다. 해를 거듭함에 따라 좀 더 여러 지식을 프로젝트에 적용할 수 있다는 부분은 본인으로 하여금 다가올 다른 프로젝트에 대해서도 큰 기대를 하게 한다. 본 프로젝트의 성공적인 마감에 향후 과제에 좋은 초석이 되길 바란다.</p>