



세종대학교

---

# Report

Database – Planner

이름 박상우

이름 정지운

# 목차

- 개요
- 시장분석
- 필요성
- 시스템 구성도
- 요구사항분석명세서
- 개발일정
- 테이블명세서
- 도메인기술서
- 다이어그램
- 정규화
- 실행모습
- 오류처리
- 역할분담
- 마무리

## ○ 개요

각각 회원가입을 통해 로그인한 사용자의 일정을 캘린더를 이용해 관리하고 일정에 관한 정보를 각 사용자가 자신에 맞게끔 일정을 추가, 수정, 삭제하도록 제공하며 친구의 일정도 확인 가능한 서비스이다. 기존 서비스들을 분석 후 기획서 작성을 시작으로 매주마다 구체적인 개발방안을 바탕으로 데이터베이스 설계 및 구현을 완료하려 한다.

각각 회원가입을 통해 로그인한 사용자의 일정을 캘린더를 이용해 관리하고 일정에 관한 정보를 각 사용자가 자신에 맞게끔 일정을 추가, 수정, 삭제하도록 제공하며 친구의 일정까지 확인 가능한 서비스를 제공한다. 친구의 일정에 추가적인 코멘트를 통해 협업이 가능하며 친구 관계인 사용자들이 편하게 소통할 수 있다.

## ○ 시장 분석

### <GOOGLE 캘린더>

1. 손쉬운 일정 예약 - 여러 캘린더를 겹쳐 놓고 사람들이 언제 시간이 나는지 확인, 회의 세부정보를 입력하면 Google 캘린더가 초대장을 발송하고 답장을 업데이트.
2. 프로젝트 캘린더 공유 - 캘린더는 회사 전체로 공유하거나 선택한 동료와 공유할 수 있으며 사용자는 캘린더를 다른 사람과 공유할 때 일정 세부정보를 보거나 캘린더를 변경할 수 있는 사람을 사용권한 설정에서 선택할 수 있다.
3. 웹페이지에 캘린더 삽입 - 일일, 주간 또는 월간 캘린더를 Google 사이트 도구 또는 웹 페이지에 삽입할 수 있으며 별도로 프로그래밍할 필요가 없다.
4. 캘린더 게시 - 캘린더를 게시하고 Google 캘린더 갤러리에서 전 세계 사용자가 검색할 수 있게 하여 외부 회사 일정을 홍보할 수 있다.
5. 모바일 액세스 - BlackBerry나 iPhone과 같은 휴대기기를 사용하여 일정에 대한 세부정보를 보고, 새 일정을 추가하며, 참석자를 초대할 수 있으며 SMS를 통해 캘린더 알림을 받을 수 있다.

## <NAVER캘린더>

1. 일정, 기념일, 할 일까지 모두 한 번에 관리 - 일정 뿐만 아니라 할 일을 간단하게 입력하고, 반복되는 기념일도 쉽게 등록가능
2. 제 때 알려주는 알림 - 잊기 쉬운 기념일, 한 번 등록 해 놓으면 계산하기 귀찮은 음력 일정도 제 날짜에 알려주어 편리
3. 한 번 터치로 빠르게 일정 쓰기 - 월뷰/듀얼뷰/주뷰에서 날짜를 길게 눌러 일정, 할 일, 기념일을 등록할 수 있다.
4. 할 일은 체크리스트로 관리 - 오른쪽으로 밀어 완료처리, 왼쪽으로 밀어 삭제. 할 일 그룹도 쉽게 만들기 가능
5. 함께 관리하는 공유 캘린더 - 캘린더를 친구, 연인, 가족, 회사 등의 구성원과 공유하여 함께 관리할 수 있다.

## ○ 필요성

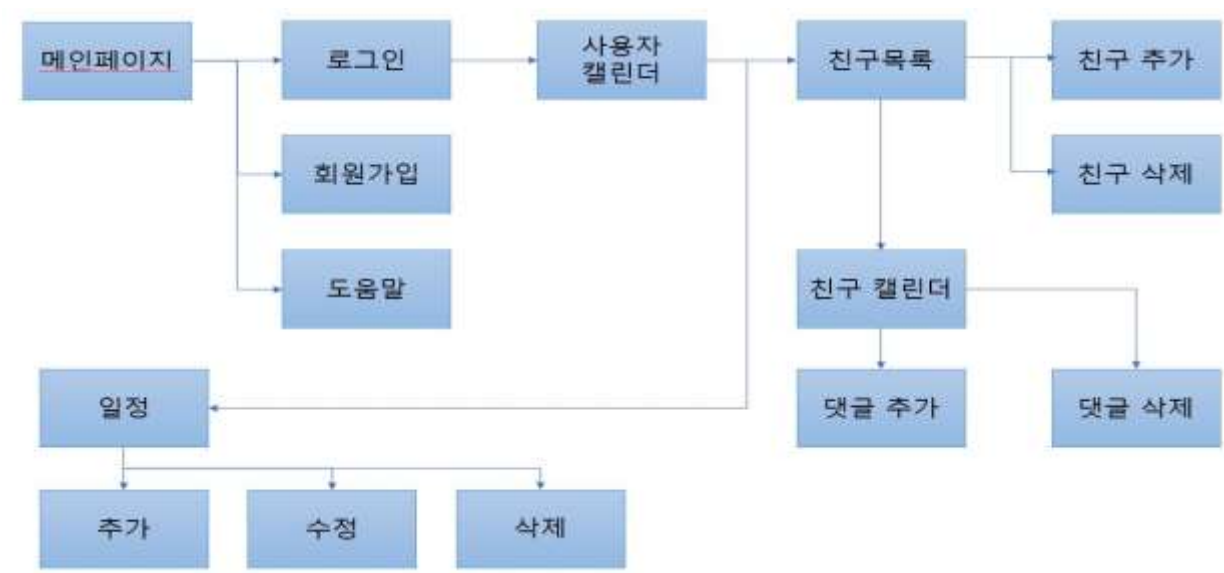
일상생활속 많은 일정들을 각각 기억하기 힘들고 간단한 메모를 통해서도 정보 추가, 일정수정 등에 대한 불편함 및 제한이 있다. 또한 친구의 일정에 대해 정보가 필요한 경우도 생각해 볼 수 있다. 날짜와 시간이 정해지지 않은 언젠가 할 일은 일반적인 TODO서비스를 이용하지만 한눈에 일정을 볼 수 없다는 단점이 있다. 확실히 시간이 정해진 일의 경우 캘린더 앱을 이용하지만 완료한 일에 대한 체크가 안된다. 이 일정들을 한번에 확인하고 관리할 서비스가 필요하다고 판단했다.

또한 기존 캘린더는 기능이 다양하지만 다양한 기능 때문에 무게가 무겁거나 사용하기 복잡하다. 기존의 비슷한 서비스를 제공하는 캘린더보다 더욱 간단하고 친구 일정확인에 대한 특화된 기능을 제공한다.

일상 생활 속 많은 일정들을 각각 기억하기 힘들고 간단한 메모를 통해서도 정보 추가, 일정수정 등에 대한 불편함 및 제한이 있어 이 프로젝트를 기획했다.

친구의 일정에 대해 정보가 필요한 경우, 프로젝트 일정 관리 시 팀원 간의 원활한 협의가 필요한 경우, 대량의 정보를 많은 친구들에게 공유 시 일일이 연락해야 하는 부담감을 줄여준다.

○ 시스템 구성도



- 일정관리의 기본기능인 추가, 수정, 삭제 기능 제공
- 친구등록을 통해 친구의 일정관리 확인 가능
- 댓글 기능을 통해 친구와 정보 교환 가능
- 라벨링 기능을 통한 일정관리 그룹화 제공

○ 요구사항 분석 명세서

분류	세부 기능	기능 설명
일정	일정 생성	새로운 일정 생성
	일정 삭제	선택된 일정 삭제
	일정 변경	해당 일정 이름 변경
	일정 수정	해당 일정의 세부 내용 수정
	세부 일정 열기	일정 클릭 시, 새 화면으로 넘어가 내용 및 댓글을 보여줌
친구	친구 추가	새로운 탭 생성
	친구 삭제	해당 탭 삭제
회원	회원 가입	회원가입을 통한 아이디 생성
	로그인	회원가입 여부를 판단 후 로그인

○ 개발 일정

구분	내용	일정					
		5/6~5/12	5/13~5/19	5/20~5/26	5/27~6/2	6/3~6/9	6/10~6/14
설계	UI 설계						
	디자인 설계						
	기능 구현						
	데이터 저장						
	프로그램 수정						
발표	발표 준비						
향후 계획에 맞춰 프로젝트 최종 마무리							

○ 테이블 기술서

테이블 이름		User							
테이블 설명		회원							
PRIMARY KEY		member_id							
FOREIGN KEY									
INDEX									
NO	PK	AI	FK	NULL	컬럼 이름	TYPE	DEFAULT	설명	참조 테이블
1	Y	Y			member_id	INT		사용자 아이디	
2					username	VARCHAR(45)		닉네임	
3					password	VARCHAR(45)		비밀번호	

회원을 관리하는 테이블입니다. 각 회원의 고유 아이디, 유저이름, 비밀번호로 구성되어 있습니다.

테이블 이름		Friend							
테이블 설명		친구							
PRIMARY KEY		id							
FOREIGN KEY		mem_id, friend_id							
INDEX									
NO	PK	AI	FK	NULL	컬럼 이름	TYPE	DEFAULT	설명	참조 테이블
1	Y	Y			id	INT		아이디	
2			Y		mem_id	INT		사용자 아이디	User
3			Y		friend_id	INT		친구 아이디	User

친구 목록을 관리하는 테이블입니다. 친구간의 관계를 정의하는 테이블 입니다.

테이블 이름		Board							
테이블 설명		일정							
PRIMARY KEY		board_id							
FOREIGN KEY		author, type							
INDEX									
NO	PK	AI	FK	NULL	컬럼 이름	TYPE	DEFAULT	설명	참조 테이블
1	Y	Y			board_id	INT		게시물 아이디	
2			Y		author	INT		작성자	User
3					title	TEXT		제목	
4			Y		type	INT		라벨	Work_type
5					content	TEXT		내용	
6					regtime	VARCHAR(40)		시간	

일정을 관리하는 테이블입니다. 작성자, 제목, 내용, 라벨, 작성 시간 등으로 구성되어 있습니다.

테이블 이름		Comment							
테이블 설명		댓글							
PRIMARY KEY		id							
FOREIGN KEY		work_id, author_id							
INDEX									
NO	PK	AI	FK	NULL	컬럼 이름	TYPE	DEFAULT	설명	참조 테이블
1	Y	Y			id	INT		아이디	
2			Y		work_id	VARCHAR(45)		게시물 아이디	Board
3					content	TEXT		내용	
4			Y		author_id	VARCHAR(45)		작성자 아이디	User

일정 댓글을 저장하는 테이블입니다. 어떤 일정에 작성된 댓글인지, 작성자, 댓글의 내용이 있습니다.

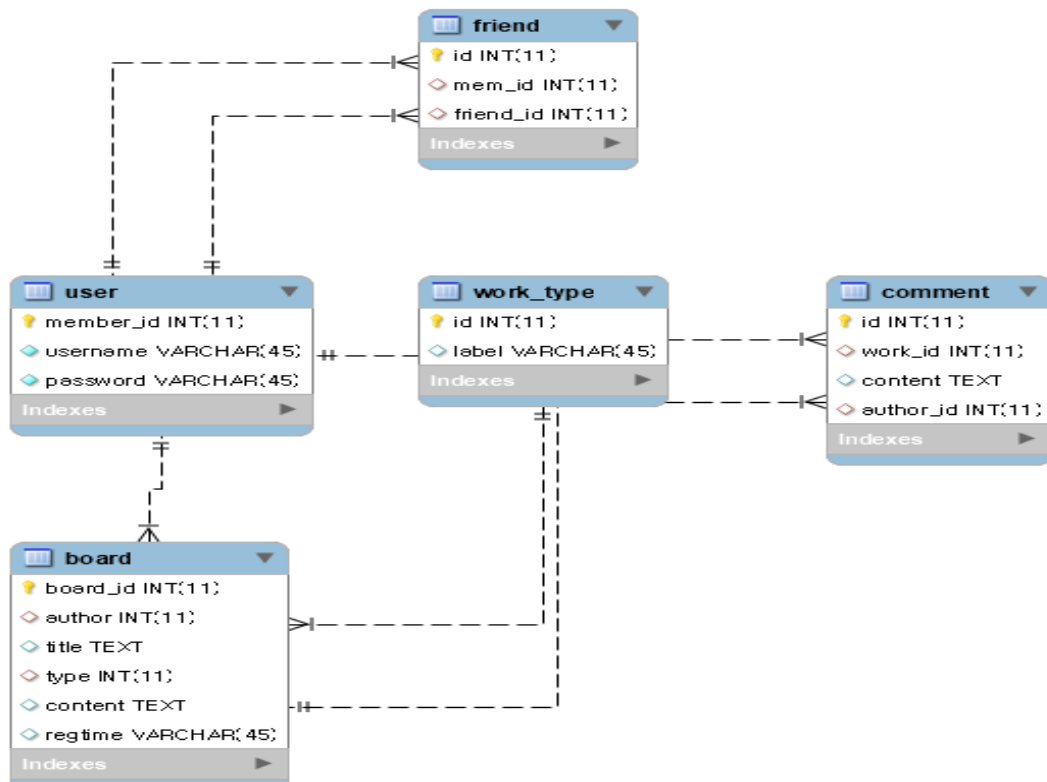
테이블 이름		Work_type							
테이블 설명		라벨종류							
PRIMARY KEY		id							
FOREIGN KEY									
INDEX									
NO	PK	AI	FK	NULL	컬럼 이름	TYPE	DEFAULT	설명	참조 테이블
1	Y	Y			id	INT		아이디	
2				Y	label	VARCHAR(45)		내용	

일정의 종류를 나타내는 테이블로 매일, 매주, 매월, 월수금, 화목토 등의 종류를 나타냅니다.

## ○ 용어 사전 및 도메인 기술서

테이블	속성 이름	TYPE	논리 이름	설명	외래키
user	member_id	INT	아이디	사용자의 고유 아이디	
	username	VARCHAR(45)	닉네임	사용자의 이름	
	password	VARCHAR(45)	비밀번호	비밀번호	
friend	id	INT	아이디	친구 관계의 고유 아이디	
	mem_id	INT	사용자 아이디	친구 관계 정의를 위한 속성	user.member_id
	friend_id	INT	친구 아이디	친구 관계 정의를 위한 속성	user.member_id
board	board_id	INT	게시물 아이디	일정의 고유 번호	
	author	INT	작성자	일정의 작성자	user.member_id
	title	TEXT	제목	일정의 제목	
	type	INT	라벨	일정의 종류	work_type.id
	content	TEXT	내용	일정의 내용	
	regtime	VARCHAR(45)	시간	일정 등록 시간	
comment	id	INT		아이디	
	work_id	VARCHAR(45)	게시물 아이디	게시물의 아이디	Board.id
	content	TEXT	내용	댓글의 내용	
	author_id	VARCHAR(45)	작성자 아이디	댓글 작성자의 아이디	user.member_id
work_type	id	INT		아이디	
	label	VARCHAR(45)	내용	일정의 종류	

## ○ ER 다이어그램





## ○ 정규화 과정

### 제 1정규형

<u>아이디</u>	닉네임	비밀번호	게시글 아이디	내용
1	testUser1	****	1, 2, 4	~~, ~~ , ~~
2	testUser2	****	3, 5	~~, ~~
3	testUser3	****	6, 7	~~, ~~

최초 유저 테이블 작성 시 사용자가 작성한 게시글의 ID를 속성으로 설정했습니다.

이 후 한 사용자가 작성할 수 있는 게시글이 여러 개이기 때문에 board\_work 속성의 원자가 여러 개로 설정해야 하는 것을 확인했습니다. 이 후 정규화 과정을 통해 속성 값을 하나씩만 포함하도록 분해하였습니다.

<u>아이디</u>	닉네임	비밀번호	<u>게시글 아이디</u>	내용
1	testUser1	****	1	~~
1	testUser1	****	2	~~
1	testUser1	****	4	~~
2	testUser2	****	3	~~
2	testUser2	****	5	~~
3	testUser3	****	6	~~
3	testUser3	****	7	~~

제 1 정규화를 마친 유저 테이블의 함수 종속 관계도입니다.



제 1정규형은 만족하지만 기본 키였던 ID가 중복되어 아이디와, 작성 게시글 아이디를 기본키로 변경했고, 여전히 불필요한 데이터가 중복되는 것을 확인했습니다.

이 때 **삽입 이상**으로는

4	testUser4	****	NULL	NULL
---	-----------	------	------	------

새로운 사용자를 추가할 경우 게시글이 없기 때문에 NULL이 되어 하는데 개체 무결성 제약조건을 위반하는 삽입이상이 발생합니다.

**갱신 이상**으로는 닉네임이나 비밀번호를 변경할 때 모든 튜플의 속성 값을 변경해야 하는데 이 때 일부만 변경한다면 데이터 일관성을 유지할 수 없습니다.

**삭제 이상**으로는 유저가 게시글을 삭제할 때 사용자의 게시글이 하나만 있었다면 사용자 데이터도 모두 삭제되어 사용자의 데이터가 손실되는 문제가 있습니다.

제 1정규형을 마치고도 발생하는 이상을 해결하기 위해 기본키가 아닌 모든 속성이 기본키에 완전 함수 종속을 제거했습니다.



유저 테이블과, 게시글 테이블을 나눠 정규화를 진행했습니다.

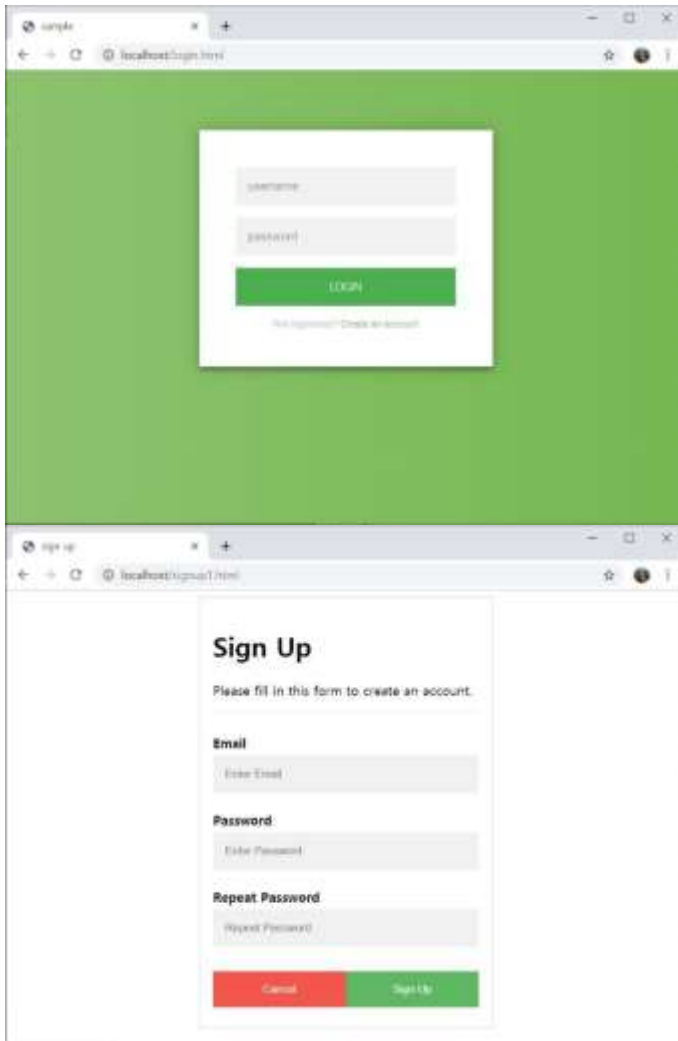
아이디	닉네임	비밀번호
1	testUser1	****
2	testUser2	****
3	testUser3	****

게시글 아이디	작성자 아이디	내용
1	1	~~
2	1	~~
3	2	~~
4	1	~~
5	2	~~
6	3	~~
7	3	~~

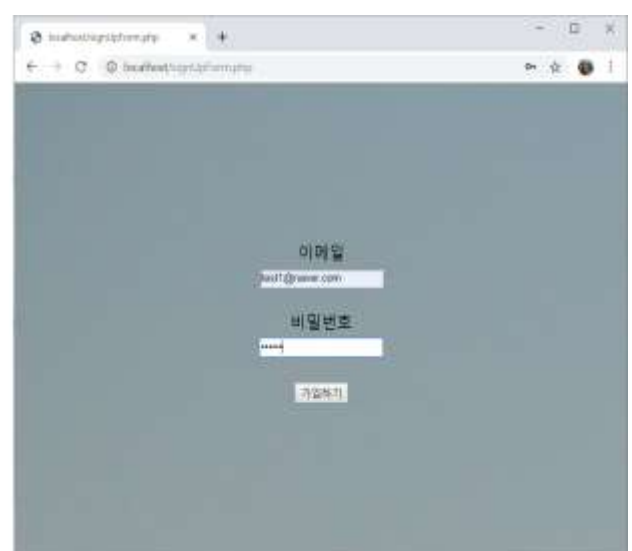
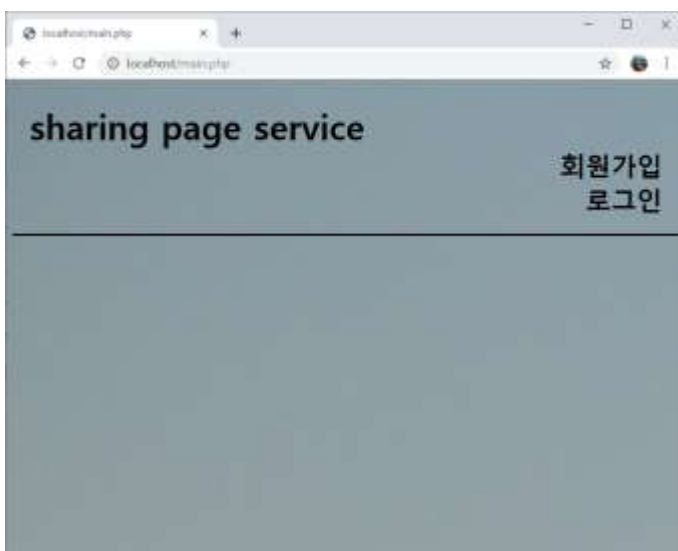
이렇게 유저테이블과 게시글 테이블을 정규화 마쳤고 이 테이블들은 모두 이행적 함수 종속 없어 제 3정규형까지 만족하고 이상이 발생하지 않아 여기까지 정규화를 진행했습니다.

정규화 과정은 이런 식으로 진행했고 프로그램을 개발하면서 수정, 추가되기도 했습니다. 게시글(보드) 테이블에 일정의 타입을 추가했고 정규화를 통해 Work\_type테이블을 추가했습니다. 댓글의 테이블의 경우 처음 테이블을 정의하면서 제 3정규형까지 만족하게 정의했기 때문에 수정이 없었습니다.

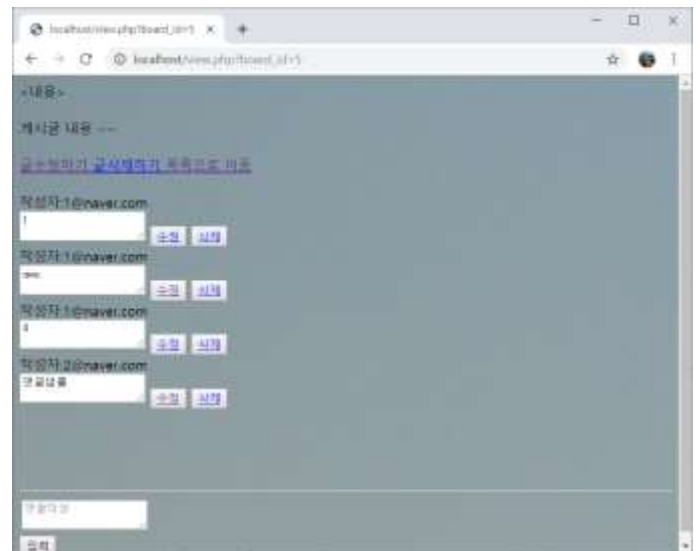
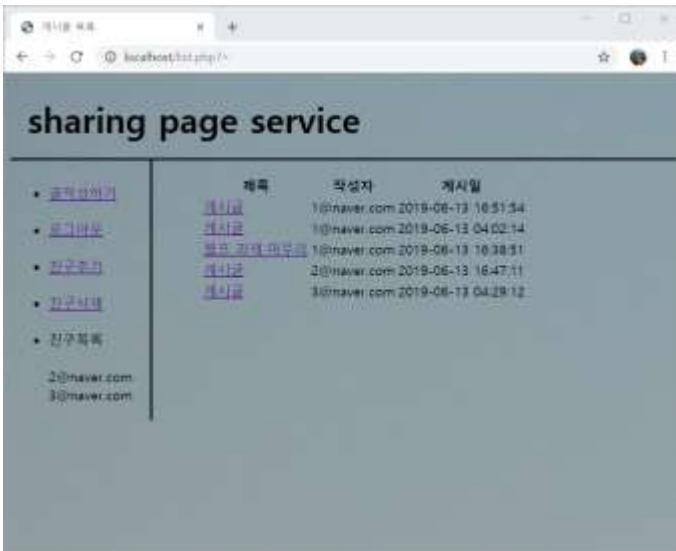
## ○ 프로그램 실행 모습



위와 같은 형태로 개발하려 했으나 데이터베이스 설계보다 프론트엔드 구현에 너무 집중되어 간단한 UI로 변경 후 데이터베이스 설계에 집중했습니다.



메인 화면과 회원 가입 화면입니다.



게시글과 친구목록이 나타나고 글 작성, 친구추가, 삭제, 로그아웃 버튼이 있습니다.

게시글에 들어가면 게시글 내용이 보이고 글 수정, 삭제, 목록으로 이동 버튼 아래 댓글과 댓글 작성이 보이는 모습입니다.

## ○ 예외 처리 과정

### 1. 회원가입 창

입력한 이메일이 제대로 된 이메일 주소 형식 인 경우 제대로 된 이메일 형식이 아니라는 메시지가 나타납니다. 입력이 공백인 경우도 경고 화면으로 전환됩니다

회원 가입 시 기존 가입한 이메일인지 확인 여부 등을 확인하고 모두 정상일 경우에만 회원가입이 처리됩니다.

### 2. 로그인 창

로그인과 비밀번호가 제대로 입력되었는지 여부를 확인하고

입력된 정보가 데이터베이스의 User테이블에 존재하는 경우에만 로그인이 됩니다.

없는 정보라면 로그인 정보가 일치하지 않는다는 화면이 나타납니다.

### 3. 친구 추가, 삭제

아이디 입력 시 존재하는 아이디 인지 확인하고 추가, 삭제 해당 아이디가 없는 경우에 오류를 출력합니다.

## ○ 역할 분담

박상우 - 개발일정 수립, 기획서, 리포트 및 ppt 작성, 시스템 구상도, 요구사항 정의서 등 기술서 작성, 데이터베이스 설계 및 정규화 과정, 정규화 과정 중 이상 확인, 최초 UI디자인 구현, 프로그램 테스트

정지운 - 개요 및 필요성 분석, 프로젝트 기획, 제안서작성, 게시판구현, 시장 분석, 시스템 구상도, 데이터베이스 설계, 유사 서비스 경쟁업체 분석, HTML, CSS, PHP, MYSQL 을 이용한 프로그램 개발

## ○ 향후 계획

최초 기획했던 UI로 프론트엔드 개발 진행, fullCalendar 자바스크립트 라이브러리 활용 HTML, CSS, JS (Bootstrap, JQuery 등 프레임워크, 라이브러리 공부), 일정 분석을 통한 광고 등의 수익 창출 모델 기획, 마지막으로 AWS, Azure등의 클라우드 서비스를 이용한 최종 배포로 프로젝트 최종 마무리할 계획입니다.