

TermProject

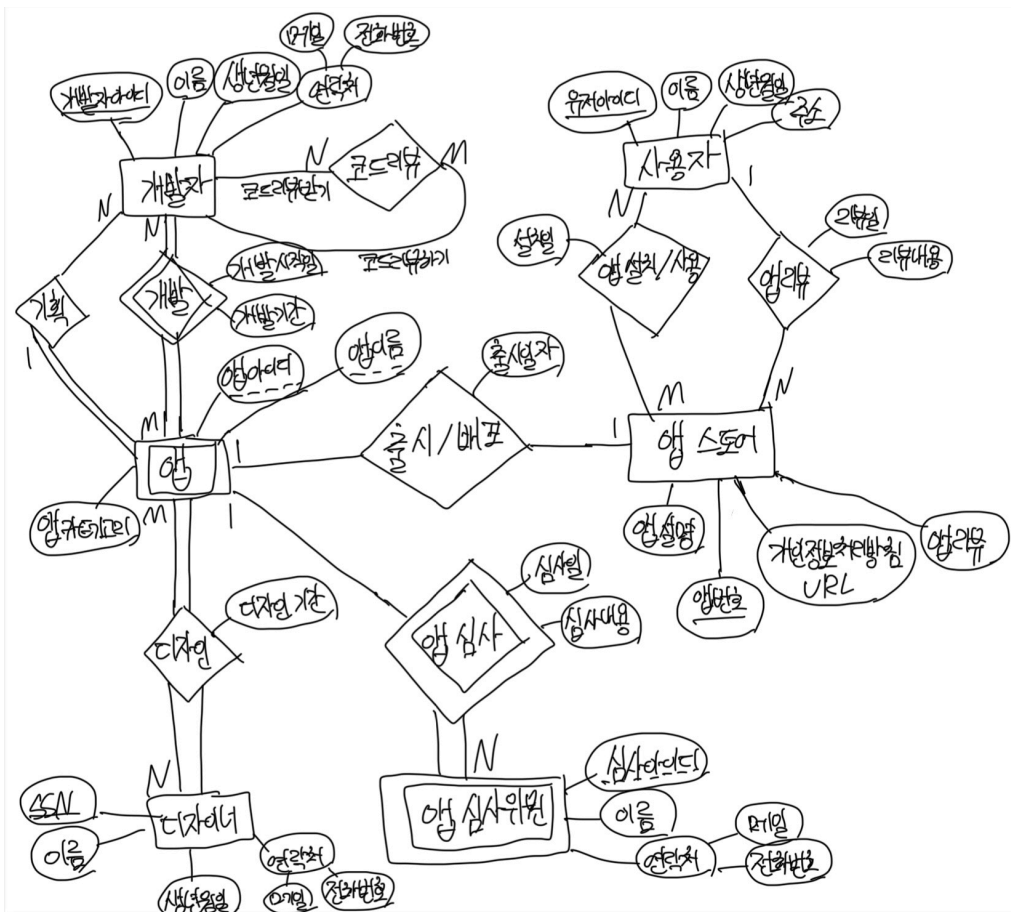
2019038046 김도현

1. Mini-World Problem: 1인 iOS앱 출시

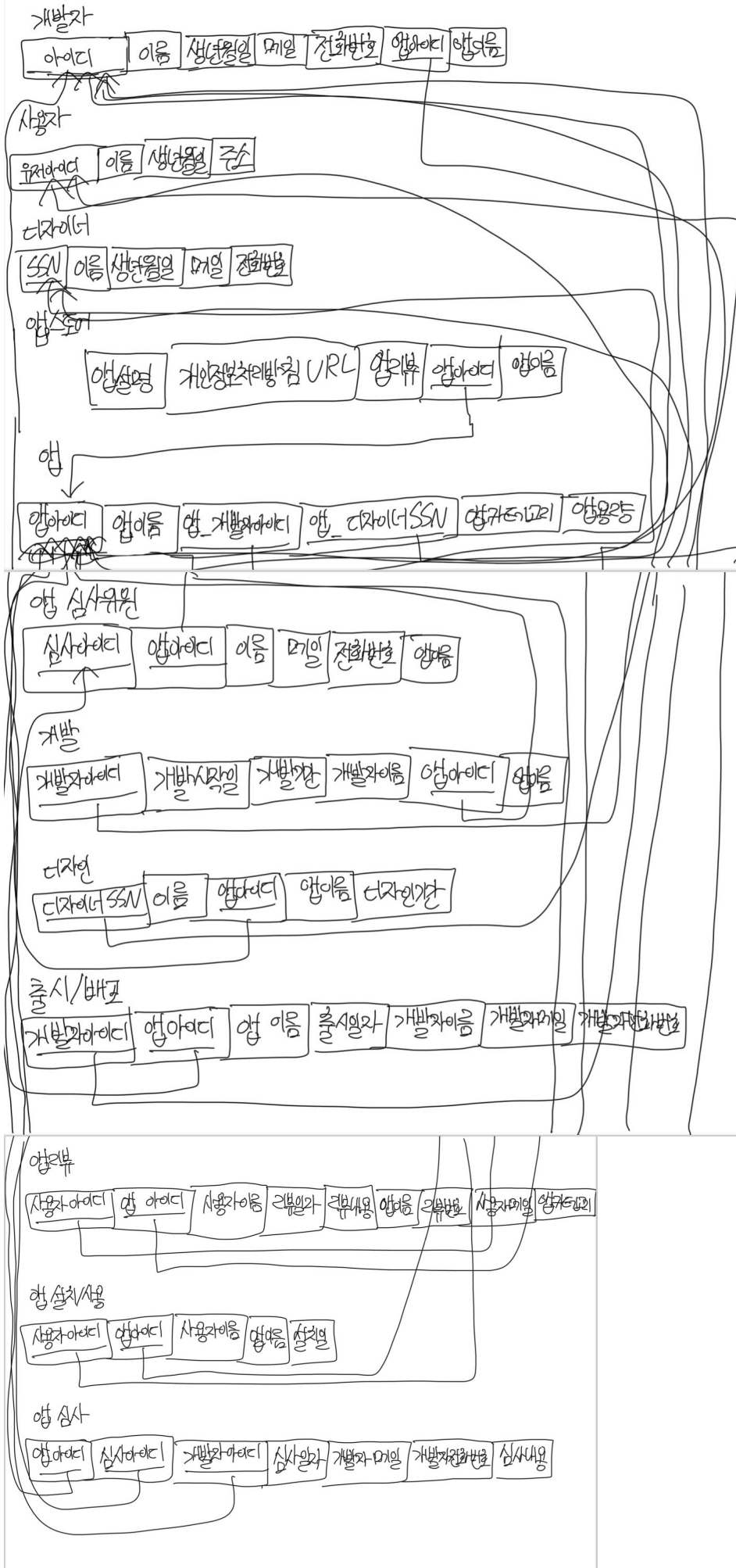
2. 요구사항

- iOS앱 출시는 앱, 앱스토어로 구성되어 있다.
- 각 앱에는 앱아이디, 앱이름, 앱을 개발하는 개발자의 개발자 아이디, 디자이너의 주민번호(SSN)가 있다.
- 앱스토어에 앱을 출시하려면 개발자 아이디, 앱 아이디, 앱 이름, 출시일자, 전화번호, 메일주소를 입력해야 한다.
- 앱은 앱 아이디로 식별한다.
- 개발자는 개발자 아이디로 식별한다.
- 디자이너는 주민번호(SSN)로 식별한다.
- 사용자는 사용자 아이디로 식별한다.
- 개발자는 여러 앱을 개발할 수 있고, 하나의 앱을 여러명의 개발자가 개발할 수 있다.
- 개발자는 서로 코드리뷰를 받을 수 있다.
- 개발자가 앱을 개발하면 개발에 대한 개발자 아이디, 개발자 이름, 개발일자, 개발기간 정보를 저장한다.
- 앱을 출시/배포하면 출시에 대한 개발자 아이디, 개발자 이름, 앱 이름, 앱 아이디, 출시일자 정보를 저장한다.
- 디자이너가 앱을 디자인하면 디자인에 대한 디자이너 주민번호, 이름, 디자인 기간 정보를 저장한다.
- 디자이너는 여러 앱을 디자인할 수 있고, 하나의 앱을 여러명의 디자이너가 디자인할 수 있다.
- 사용자는 여러 앱을 설치/사용할 수 있고, 하나의 앱을 여러명의 사용자가 설치/사용할 수 있다.
- 사용자는 각 앱을 리뷰할 수 있고, 하나의 앱은 하나의 사용자가 리뷰할 수 있다.
- 사용자가 앱을 설치하면 설치에 대한 사용자 아이디, 사용자 이름, 앱 아이디, 앱 이름, 설치일 정보를 저장한다.
- 사용자가 앱을 리뷰하면 리뷰에 대한 사용자 아이디, 앱 아이디, 사용자 이름, 리뷰일자, 리뷰내용 정보를 저장한다.
- 앱스토어 앱심사위원은 하나의 앱을 심사할 수 있고, 하나의 앱은 여러명의 앱스토어 앱심사위원에 의해 심사될 수 있다.
- 앱스토어 앱심사위원이 앱을 심사하면 심사에 대한 앱 아이디, 심사 아이디, 개발자 아이디, 심사일자, 개발자 연락처, 심사내용 정보를 저장한다.
- 앱스토어는 앱 아이디, 앱 번호, 개인정보처리방침url, 앱 설명, 앱 리뷰 정보를 저장한다.
- 앱스토어 앱심사위원은 앱 심사아이디로 식별한다.

3. ER (Entity-Relationship) Diagram



4. ER-Relational Mapping을 통해 Relation Schema를 생성



5. Relation Schema에 대해 Normalization을 수행하여 최소 3NF의 형태가 되도록 한다.

개발자_정보

<u>아이디</u>	이름	생년월일	이메일	전화번호	아이디
------------	----	------	-----	------	-----

개발자_연

<u>아이디</u>	아이름
------------	-----

사용자

<u>유저아이디</u>	이름	생년월일	주소
--------------	----	------	----

디자이너_정보

<u>SSN</u>	이름	생년월일	이메일	전화번호	아이디
------------	----	------	-----	------	-----

디자이너_연

<u>아이디</u>	아이름
------------	-----

앱스토어

<u>앱아이디</u>	앱이름	개인정보처리방침 URL	앱버
-------------	-----	--------------	----

앱

<u>앱아이디</u>	앱이름	앱개발자아이디	앱디자이너 SSN	앱카테고리	앱용량
-------------	-----	---------	-----------	-------	-----

앱심사위원_정보

<u>심사아이디</u>	이름	이메일	정보
--------------	----	-----	----

앱심사위원_연

<u>아이디</u>	아이름
------------	-----

개발_정보

<u>개발자아이디</u>	개발자이름	개발자	개발자이름	앱이름
---------------	-------	-----	-------	-----

개발_연

<u>아이디</u>	아이름
------------	-----

디자인_정보

디자인SSN	이름	업종아이디	디자인기간
--------	----	-------	-------

디자인_업

업종아이디	업종명
-------	-----

출시/배포_개발자

개발자아이디	개발자이름	개발자메일	개발자전화번호	업종아이디
--------	-------	-------	---------	-------

출시/배포_업

업종아이디	업종명	출시일과
-------	-----	------

업종별_전체

사용자아이디	업종아이디	리뷰일과	리뷰내용	리뷰별
--------	-------	------	------	-----

업종별_사용자

사용자아이디	사용자이름	사용자메일
--------	-------	-------

업종별_업

업종아이디	업종명	업종카테고리
-------	-----	--------

업종별/사용_전체

사용자아이디	업종아이디	실시간
--------	-------	-----

업종별/사용_사용자

사용자아이디	사용자이름
--------	-------

업종별_업

업종아이디	업종명
-------	-----

업종별_전체

업종아이디	심사아이디	개발자아이디	심사일과	심사내용
-------	-------	--------	------	------

업종별_개발자

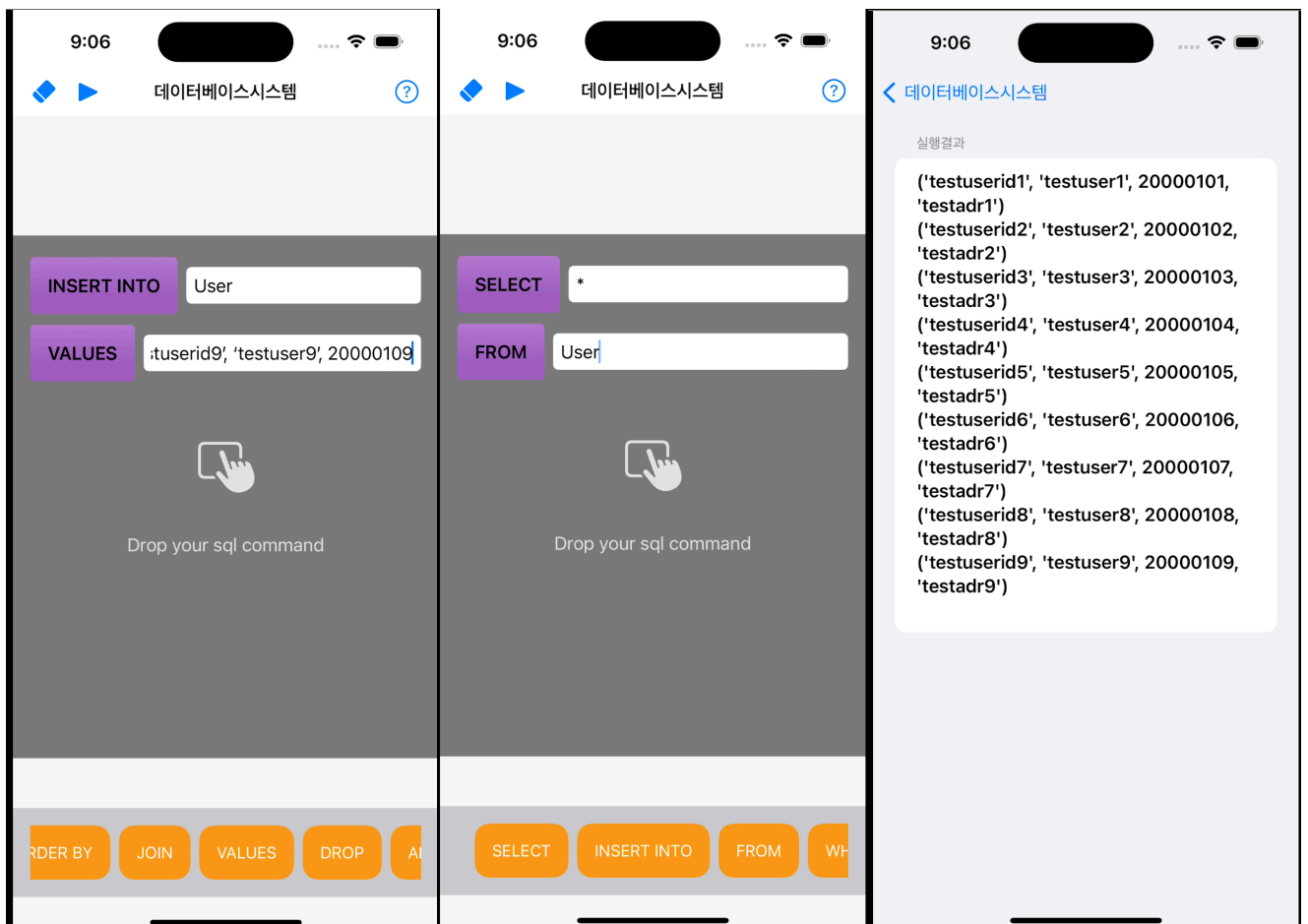
개발자아이디	개발자이름	개발자전화번호
--------	-------	---------

6. MySQL 데이터베이스 생성

```
mysql> CREATE DATABASE TermProject;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| TermProject |
| madang |
| mysql |
| performance_schema |
| smallbookstore |
| sys |
+-----+
7 rows in set (0.00 sec)
```

7. 프로그램을 통해 데이터베이스에 접속하여 사용자가 요청을 수행하도록 한다.
iOS인터페이스의 드래그앤 드롭과 사용자가 직접 입력하는 텍스트필드를 기반으로 구성하였습니다.



8. 프로그램은 Network를 통해 데이터베이스에 접속해야 한다.

- iOS에서 Network를 통해 파이썬 백엔드 서버에 접속하고 파이썬 백엔드 서버에서 Network를 통해 MySQL서버에 접속하는 방식으로 구성하였습니다. bridge.py파일과 ContentView.swift 파일을 참조해 주세요.

***참고 사항**

- 해당 프로젝트에서 iOS부분인 DBTermProject는 macOS 14, Xcode15, iOS 16버전이상에서 진행되었습니다.
혹시 테스트하신다면 참고해주시면 감사하겠습니다.

github 주소: <https://github.com/a-mystic/DB-School-lecture/tree/main/Term%20Project>

시연영상: <https://github.com/a-mystic/DB-School-lecture/blob/main/README.md>