프로젝트 기술서

프로젝트 개요

프로젝트 명 Sokoban 게임

개발 기간 2018. 10.15 (1일)
개발인원 1명
담당역할 개발

개발환경

OS	Windows 7, 10
Development Tool	Visual Studio
Language	C

프로젝트 설명

C언어의 끝에 알고리즘을 만드는 연습과 C언어를 마무리 짓는 프로그램으로 소코반게임을 만들었다. 상자를 목적지에 옮기는 단순한 게임으로, command창에서 실행된다.

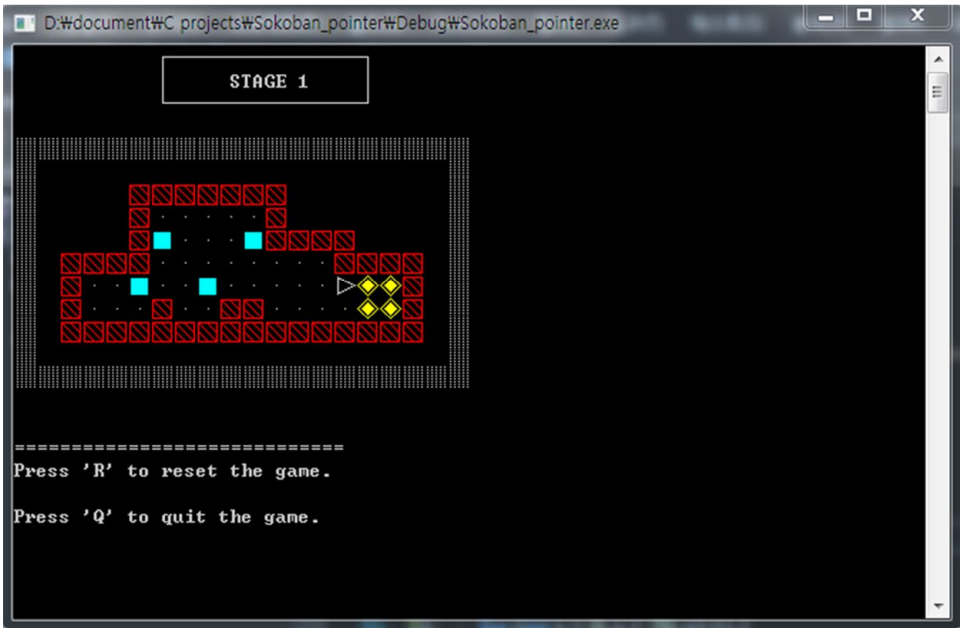
주요 코드

```
void keyInput()
{
    int xOrigin = xAxis, yOrigin = yAxis;
    char cmd, key;
    cmd = getch();
    key = getch();
    int xch = 0, ych = 0;
    switch (key) {
        case LEFT:
            xch--;
            break;
        case RIGHT:
            xch++;
            break;
        case UP:
            ych--;
            break;
        case DOWN:
            ych++;
            break;
    }
    if (Map[yAxis + ych][xAxis + xch] ==
SPACE || Map[yAxis + ych][xAxis + xch] ==
GOAL) {
        xAxis += xch;
        yAxis += ych;
    }
    else if (Map[yAxis + ych][xAxis + xch]
== BLACK && (Map[yAxis + (2 * ych)]
[xAxis + (2 * xch)] == SPACE || Map[yAxis
+ (2 * ych)][xAxis + (2 * xch)] == GOAL)) {
        Map[yAxis + (2 * ych)][xAxis + (2 *
xch)] = BLACK;
        xAxis += xch;
        yAxis += ych;
    }
    if ((yOrigin == Goalpoint[0][0] &&
xOrigin == Goalpoint[0][1]) ||
        (yOrigin == Goalpoint[1][0] &&
xOrigin == Goalpoint[1][1]) ||
```

```
        (yOrigin == Goalpoint[2][0] &&
xOrigin == Goalpoint[2][1]) ||
        (yOrigin == Goalpoint[3][0] &&
xOrigin == Goalpoint[3][1])){
        Map[yOrigin][xOrigin] = GOAL;
    }else{
        Map[yOrigin][xOrigin] = SPACE;
        Map[yAxis][xAxis] = POSITION;
    }
```

```
    switch (cmd) {
        case 'r':
            Reset();
            break;
        case 'q':
            exit(0);
            break;
    }
}
```

실행 결과



프로젝트 기술서

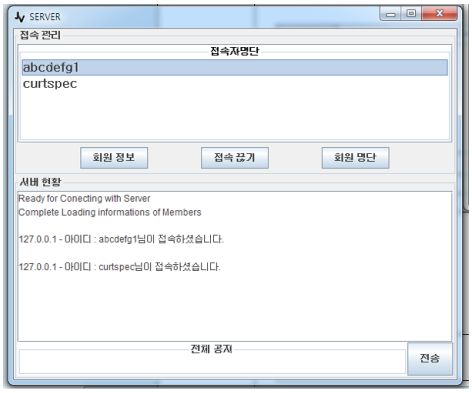
프로젝트 개요

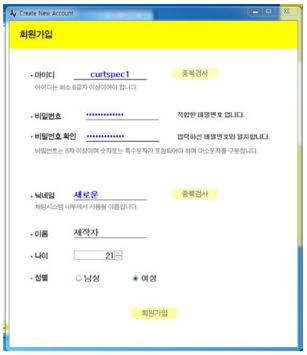
프로젝트 명	ID 기반 Chatting 프로그램	
개발 기간	2018. 11.19 ~ 2018. 11. 30 (2주)	
개발인원	1명	
담당역할	기획, 개발, 디자인	
개발환경	**OS**	Windows 7, 10
	Development Tool	JDK 8
		Eclipse
	Language	JAVA

프로젝트 설명

객체지향 프로그래밍을 배운 후에 배운 것들을 실습해보는 기회이자, 간단한 TCP통신을 구현하여 1:1 통신을 가능하게 하는 프로그래밍이다. 서버가 따로 존재한다는 가정하에 한대의 컴퓨터가 서버와 클라이언트의 역할을 모두 수행한다. 아이디를 기반으로 유저를 검색하고 메세지를 보낼 대상을 아이디를 기반으로 지정할 수 있다. 서버에서는 접속해있는 유저들의 상태를 확인 할 수 있으며 접속 상태를 변경하고 공지를 보낼 수 있다.

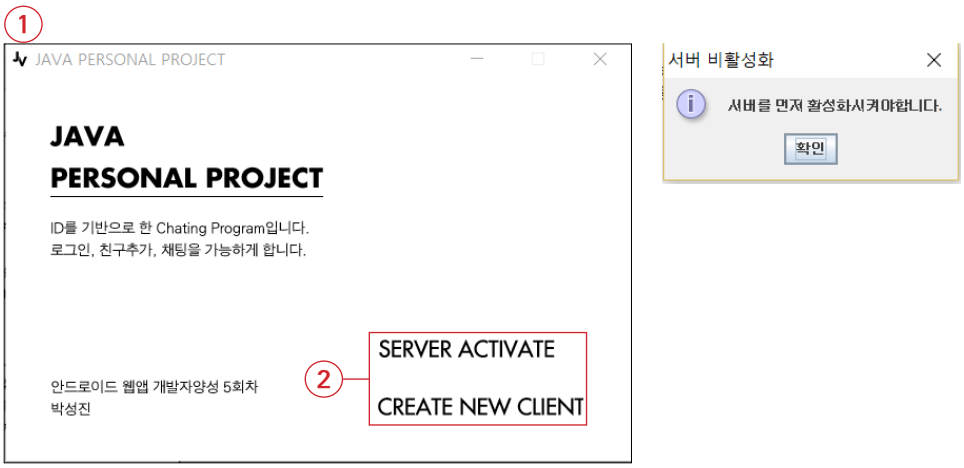
주요 화면 구성





화면 구성

MainFrame



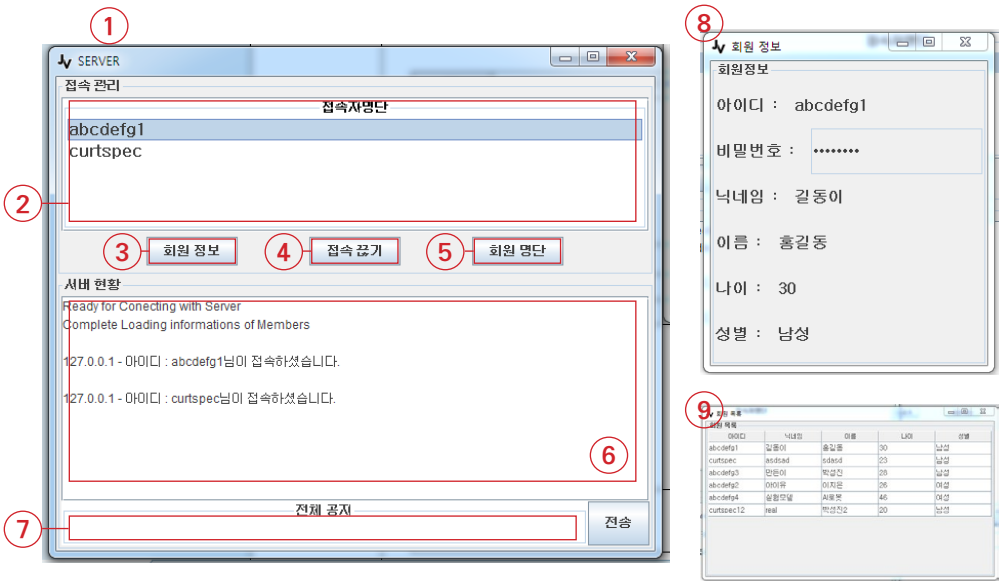
화면 기술 설명

1. public static void main(String[] args){ }이 위치한 MainFrame class가 정의되어 있고, process가 시작되면 JFrame을 상속받은 MainFrame이 생성된다. MainFrame 안에 JPanel을 상속받은 class를 만들고 paintComponent를 오버라이드하여 배경의 이미지를 넣어준다.

2. 서버활성화 버튼과 클라이언트 생성 버튼에 각각 ActionListener를 추가해주고 만약, 서버없이 클라이언트만 존재하는 것을 방지하기 위해서 클라이언트 생성버튼을 누를 때, 서버가 활성화 되어 있는지를 확인한다. 만약 서버가 없는 상황이라면 다이얼로그가 생성되어 메세지를 보여준다.

화면 구성

ServerFrame



화면 기술 설명

1. JFrame을 상속받은 ServerFrame Class를 정의하고, 내부 Class로 ServerThead를 정의하여 JFrame의 생성과 동시에 ServerThead의 run()호출된다.

2. ServerSocket을 통해 들어온 클라이언트의 명단을 ArrayList의 형태로 저장하고 InputSream을 통해서 들어온 회원정보를 JList를 통해서 보여준다.

3. List에서 특정 회원을 선택한 후 회원 정보를 누르면 8번 화면이 뜬다. ServerThead가 실행되면 FileInputStream을 열고 보조 스트림으로 ObjectInputStream을 사용하여 Local에 저장되어 있는 회원의 명단을 읽어온다. 회원정보는 각각의 객체로 관리되어 Collection Framework에 담겨 특정 회원의 아이디로 정보를 가져와서 나타낸다.

4. ServerSocket에서 client가 접속할 때 전달하는 아이디 정보와 생성되는 socket을 각각 Key와 Value로 저장하여 Map에 저장하고 JList에서 선택한 아이디로 socket을 찾아와서 연결을 끊는다.

5. 누를 시에 9번 창이 뜨고, FileInputStream으로 읽어들인 회원들의 정보를 모두 보여준다.

6. Server의 Socket이 만들어지거나 client가 접속하는 등 새로운 상황이 발생할 시에 TextArea에 내용을 append하여 표시하여 서버운영자로 하여금 서버의 상황을 한눈에 파악할 수 있게 한다.

7. Client로 부터 온 메세지와 Server에서 보내는 전체 공지는 메세지를 받는 대상이 다르다. 때문에 가지고 있는 모든 socket의 output stream을 열어서 메세지를 전달하는 method와 특정 타겟의 socket으로 output stream을 열어서 메세지를 전달하는 method를 따로 만들어서 실행한다.

화면 구성

LoginFrame

① ②

화면 기술 설명

1. JFrame을 상속받고 paintComponent를 오버라이드하여 배경과 메인로고를 그려낸다. 아이디와 비밀번호 TextFeild에 각각 KeyListener를 추가해서 아이디의 TextFeild에서는 다음 TextFeild에 Focus를 전달하고 비밀번호의 TextFeild에서는 로그인 작업을 진행하게 한다.

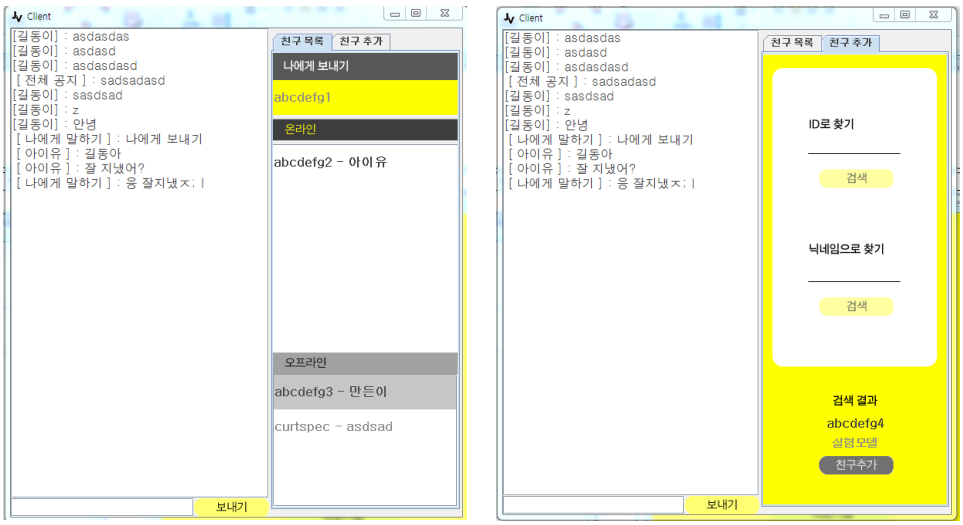
2. 아이디 중복검사, 닉네임 중복검사를 누르면 서버로 보내고 서버에서 boolean을 리턴받아 중복여부를 검사한다. 만약 사용중인 아이디라면, 다이얼로ㄷ그를 또한 비밀번호를 작성할 때 KeyReleased를 오버라이드하여 키보드가 눌러질때 마다 생성 규칙과 부합하는지 검사하여

표시한다. 비밀번호 확인 칸도 마찬가지로 두번의 비밀번호 입력이 동일한지 검사하여 표시한다.
회원가입 버튼을 누를 때 모든 입력 칸을 조사하여 입력되지 않은 정보가 있다면 알려준다.
모든 정보가 올바르게 입력되었다면 OutputStream을 통해서 서버로 데이터를 전송한다.

화면 구성

LoginFrame

① ②



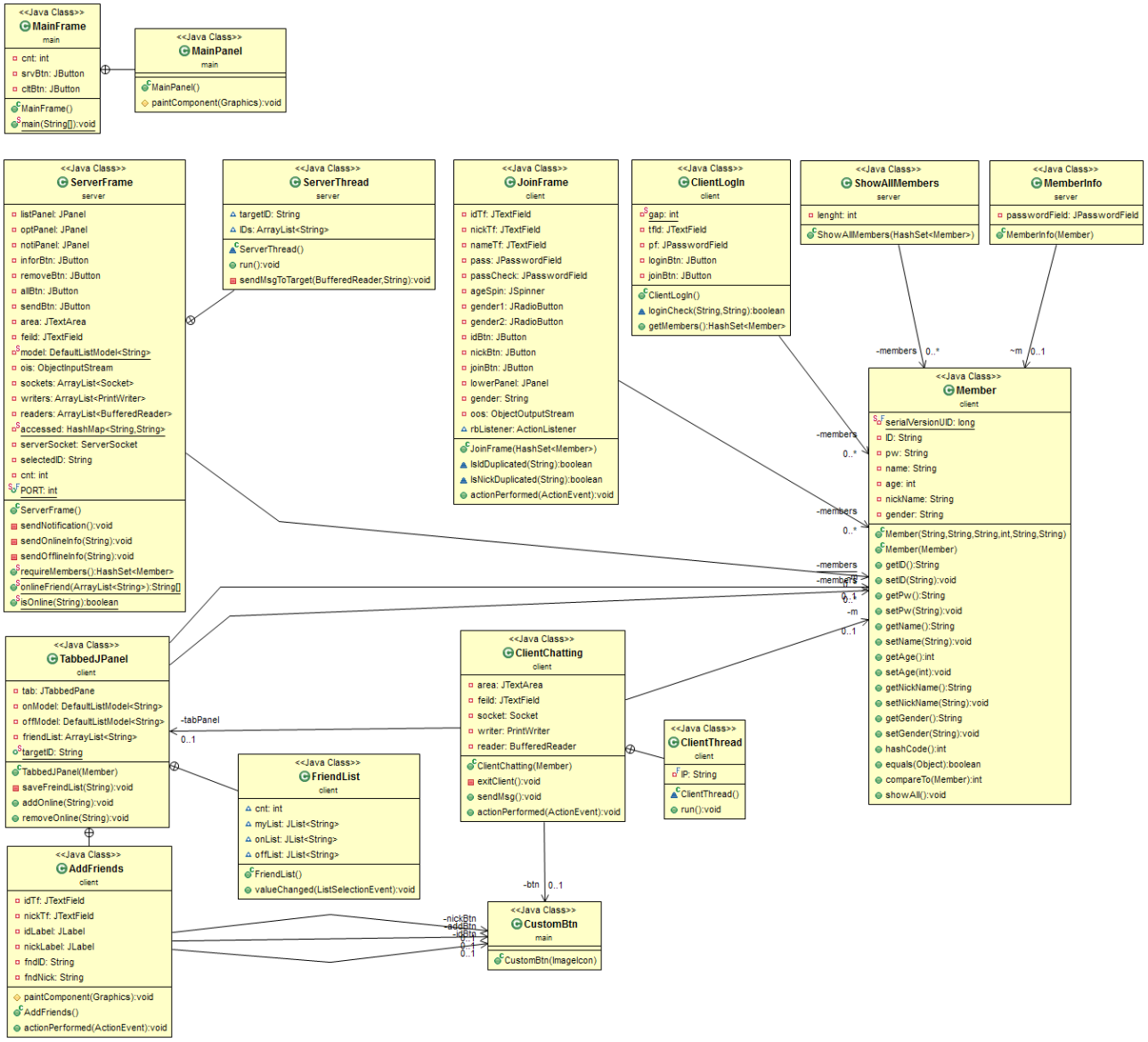
화면 기술 설명

1. Client 접속화면은 크게 2가지 영역으로 구분된다. chatting 영역과 TabbedPanel 으로 만들어진 친구목록 영역이다. WindowListner를 추가해서 Frame이 생성되면 local drive에 저장되어 있는 이때 까지의 채팅 내용을 FileInputStream을 통해 읽어들이고 내용을 모두 화면에 표시한다. Frame이 꺼질 때에는 이때까지의 채팅 내용을 다시 저장한다. 그리고 친구목록도 local에 저장되어 같은 방식으로 읽어들인다.

ClientFrame이 생성되면 Client Thread가 생성되고 이 Thread에서 Network 작업을 전담해서 수행한다. Socket을 생성하고 Server와 연결한다, 연결이 성공하면 접속된 Client의 id를 전송하고 친구 목록을 모두 전달하면 server에서는 온라인인 친구들의 목록만을 리턴한다. 메세지를 보내는 대상은 스스로에게 하거나 친구 중 한명을 골라서 보낼 수 있다.

2. 친구를 검색하는 방법은 2가지이다. 아이디로 검색하는 방법과 닉네임으로 검색하는 방법이 있다. 두 개의 정보 모두 중복될 수 없는 정보이기에 Server로 보내고 검색에 성공한 경우에 해당 회원의 아이디와 닉네임을 리턴한다.

클래스 다이어 그램


<<Java Class>>
MainFrame
main
cnt: int
srvBtn: JButton
cltBtn: JButton
MainFrame()
main(String[]):void
<<Java Class>>
MainPanel
main
MainPanel()
paintComponent(Graphics):void
<<Java Class>>
ServerFrame
server
listPanel: JPanel
optPanel: JPanel
notiPanel: JPanel
inforBtn: JButton
removeBtn: JButton
allBtn: JButton
sendBtn: JButton
area: JTextArea
feild: JTextField
model: DefaultListModel<String>
ois: ObjectInputStream
sockets: ArrayList<Socket>
writers: ArrayList<PrintWriter>
readers: ArrayList<BufferedReader>
accessed: HashMap<String,String>
serverSocket: ServerSocket
selectedID: String
cnt: int
PORT: int
ServerFrame()
sendNotification():void
sendOnlineInfo(String):void
sendOfflineInfo(String):void
requireMembers():HashSet<Member>
onlineFriend(ArrayList<String>):String[]
isOnline(String):boolean
<<Java Class>>
ServerThread
server
targetID: String
IDs: ArrayList<String>
ServerThread()
run():void
sendMsgToTarget(BufferedReader,String):void
<<Java Class>>
JoinFrame
client
idTf: JTextField
nickTf: JTextField
nameTf: JTextField
pass: JPasswordField
passCheck: JPasswordField
ageSpin: JSpinner
gender1: JRadioButton
gender2: JRadioButton
idBtn: JButton
nickBtn: JButton
joinBtn: JButton
lowerPanel: JPanel
gender: String
oos: ObjectOutputStream
rbListener: ActionListener
JoinFrame(HashSet<Member>)
IsIdDuplicated(String):boolean
IsNickDuplicated(String):boolean
actionPerformed(ActionEvent):void
<<Java Class>>
ClientLogIn
client
gap: int
tfId: JTextField
pf: JPasswordField
loginBtn: JButton
joinBtn: JButton
ClientLogIn()
loginCheck(String,String):boolean
getMembers():HashSet<Member>
<<Java Class>>
ShowAllMembers
server
lenght: int
ShowAllMembers(HashSet<Member>)
<<Java Class>>
MemberInfo
server
passwordField: JPasswordField
MemberInfo(Member)
-members
0..*
~m
0..1
<<Java Class>>
Member
client
serialVersionUID: long
ID: String
pw: String
name: String
age: int
nickName: String
gender: String
Member(String,String,String,int,String,String)
Member(Member)
getID():String
setID(String):void
getPw():String
setPw(String):void
getName():String
setName(String):void
getAge():int
setAge(int):void
getNickName():String
setNickName(String):void
getGender():String
setGender(String):void
hashCode():int
equals(Object):boolean
compareTo(Member):int
showAll():void
<<Java Class>>
TabbedJPanel
client
tab: JTabbedPane
onModel: DefaultListModel<String>
offModel: DefaultListModel<String>
friendList: ArrayList<String>
targetID: String
TabbedJPanel(Member)
saveFreindList(String):void
addOnline(String):void
removeOnline(String):void
<<Java Class>>
ClientChatting
client
area: JTextArea
feild: JTextField
socket: Socket
writer: PrintWriter
reader: BufferedReader
ClientChatting(Member)
exitClient():void
sendMsg():void
actionPerformed(ActionEvent):void
-tabPanel
0..1
<<Java Class>>
ClientThread
client
IP: String
ClientThread()
run():void
<<Java Class>>
FriendList
client
cnt: int
myList: JList<String>
onList: JList<String>
offList: JList<String>
FriendList()
valueChanged(ListSelectionEvent):void
<<Java Class>>
AddFriends
client
idTf: JTextField
nickTf: JTextField
idLabel: JLabel
nickLabel: JLabel
fndID: String
fndNick: String
paintComponent(Graphics):void
AddFriends()
actionPerformed(ActionEvent):void
-btn
0..1
-nickBtn
-addBtn
-idBtn
<<Java Class>>
CustomBtn
main
CustomBtn(ImageIcon)

프로젝트 기술서

프로젝트 개요

프로젝트 명 Spin Mob 슈팅게임

개발 기간 2019. 01.02 ~ 2019. 01. 09 (1주)

개발인원 1명

담당역할 레이아웃 설계, 개발

개발환경

OS	Windows 7, 10
Development Tool	android studio 3.3.2 Genymotion
Language	JAVA
DBMS	-

프로젝트 설명

교육받은 내용을 바탕으로 View를 상속받아서 Canvas객체를 이용한 화면 구현을 목표로 하는 교육 실습용 어플리케이션이다. Graphic은 대부분 'Dragon Fly'라는 어플리케이션의 디자인을 이용하여 화면을 구성했다.

화면 구성

MainActivity

화면 기술 설명

Activity의 onCreate에서 MediaPlayer 객체를 생성하고 setLooping으로 설정하여 배경음악의 반복재생을 설정한다. onResume에서 음악을 재생하고 onPause에서 일시정시하고 onDestroy에서 MediaPlayer를 정지하고 release하여 메모리를 관리한다.
각 button에는 onclick 속성으로 method를 연결하여 실행한다.

화면 구성

GameActivity

화면 기술 설명

Activity에서는 여러가지 버튼의 onClickListener 또는 onCheckedChangeListener를 추가하는 작업을 진행한다. 또한 배경음악을 MainActivity와 동일하게 각 LifeCycle의 callback method에 오버라이드 하여 재생 / 정지를 한다.

게임 플레이에 관한 코드를 가지고 있는 GameView라는 클래스를 SurfaceView를 상속받아서 정의한다. 또한 SurfaceView에 필요한 Callback을 implements하여 각 추상 method들을 오버라이딩하여 사용한다. Game에 필요한 로직을 담당하는 GameThead는 Thead를 상속받아 만들고 surfaceView가 만들어지고 실행되는 surfaceChanged를 오버라이딩하여 GameThead를 run시킨다. TouchEvent가 발생하면 View에서 event를 받고 GameThead에 그 정보를 전달한 후 event를 소비한다. GameThread에서는 전달받은 TouchEvent의 정보를 바탕으로 동작을 수행한다. 조이패드위에서 TouchEvent가 발생하면 TouchMove시에 각도를 계산하여 플레이어의 캐릭터를 움직인다. 화면에 나타나는 것들은 모두 각각의 객체로 만들어져있다. run()에서 반복문을 통해서 만들어내고, 움직이고, 충돌을 계산하는 과정을 진행한다.

SoundPool객체를 만들어서 각각의 상황에 따라 효과음을 발생시킨다. 또한 Vibrator를 통해서 플레이어의 체력이 깍이는 경우에 진동을 주게한다. 이때 manifests에서 진동에 관한 권한을 부여한 후에 사용한다.

화면 구성

GameActivity

① ②

1. 스코어 아래에 있는 종료버튼을 누를 때 보이게 되는 view이다. AlertDialog로 보여주는 것이 아니라 ViewGroup위에서 만들어 viewport바같에 배치하여 보이지 않게 한 후, onClick의 call back method에서 animation객체를 통해 아래로 내려오게 만든다.

2. 화면의 우측에 있는 설정버튼을 누르면 같은 방식으로 Dialog가 보여지게 된다. Toggle버튼을 누르게 되면 public staic 변수의 값이 바뀌게 되어 배경음악, 효과음, 진동을 제어하게 된다.

화면 구성

GameOverActivity

화면 기술 설명

게임이 끝나면 게임 점수를 기반으로 기록을 대조하여 챔피언 점수와 나의 점수르 표시한다. 위의 아이콘을 터치하면 묵시적 인텐트를 통해 갤러리앱이 실행되어 사진을 가져와서 Glide를 통해 화면에 나타낸다. 기본적으로 최고점과 이미지의 url에 대한 정보는 SharedPreference에 저장되어 있으며 onCreate에서 SharedPreference에 정보의 유무를 검사하고 가져오는 작업을 진행한다.

프로젝트 기술서

프로젝트 개요

프로젝트 명 PriceTag

개발 기간 2018. 12. 20 (하루)
개발인원 1명
담당역할 기획, 디자인, 개발

개발환경

OS	Windows 7, 10
Development Tool	android studio 3.3.2
Language	JAVA
DBMS	–

프로젝트 설명

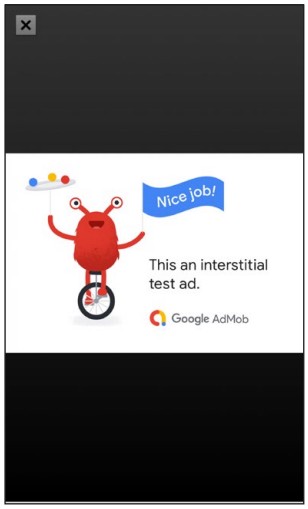
Xml pullparser를 이용해서 Open API를 활용한 Application을 만들어보는 프로젝트이다. 서울시의 공공데이터를 이용했으며 특정 시장(마트포함)의 농수산물 가격을 확인할 수 있는 어플리케이션이다.

화면 구성

화면 기술 설명

onCreate에서 AdMob을 초기화 하고 Banner 광고를 레이아웃에 추가하여 사용한다. 어느곳의 마트 정보를 가져올지를 정하는 과정이며 각 spinner들은 entries 속성을 이용해서 만들어낸다. Next버튼을 통해 다음 화면의 데이터가 바뀌진다.

화면 구성



화면 기술 설명

서울시에서 제공하는 open API를 분석하기 위해서 앞에서 받은 마트의 정보를 사용해서 URL에 GET방식으로 data를 전달하고 이때 받는 정보는 XmlPullParser 객체를 통해서 한줄씩 가져오게 된다. xml의 태그명과 구조를 파악한 뒤 분석하는 코드를 통해 얻어온 데이터를 편집하여 저장한다. 저장된 데이터는 ListView에 담기게 된다. 상단의 보기방식을 바꾸게 되면 GridView로 바뀌어 화면에 보이게 된다. 또한 onDestroy에서 50%의 확률로 전면광고를 띄운다.

프로젝트 기술서

프로젝트 개요

프로젝트 명 HOMA 원룸관리 어플리케이션

개발 기간 2019. 02.18 ~ 2019. 03. 15 (4주)

개발인원 1명

담당역할 기획, 개발, 디자인, 서버, 관리

개발환경

OS	Windows 7, 10
Development Tool	android studio 3.3.2
	Sublime Text 3
	Adobe Illustration cs6
	Adobe PhotoShop cs6
Language	JAVA, PHP
DBMS	MYSQL

프로젝트 설명

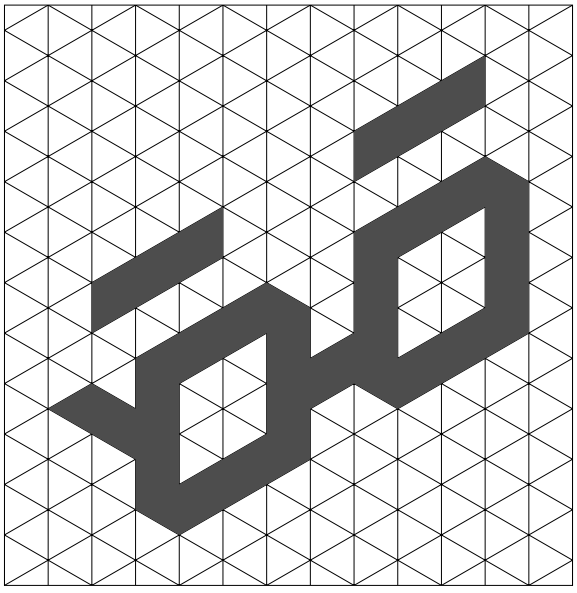
일정 규모 이상의 기업들에서는 사내 인트라넷이나 내부 DB를 통해서 원활한 데이터, 자료의 관리가 이루어 지고 있다. 그러나 개인이 진행하는 소규모 임대업의 경우에는 맞춤형 시스템을 제작하기에 무리가 있다. 이러한 불편함을 조금이나마 해소시켜주기 위해 제작을 시작하게 되었다. 임대건물의 정보, 임차인들의 정보 모두 저장하고 한 눈에 볼 수 있는 어플리케이션이다.

디자인 및 설계

Logo

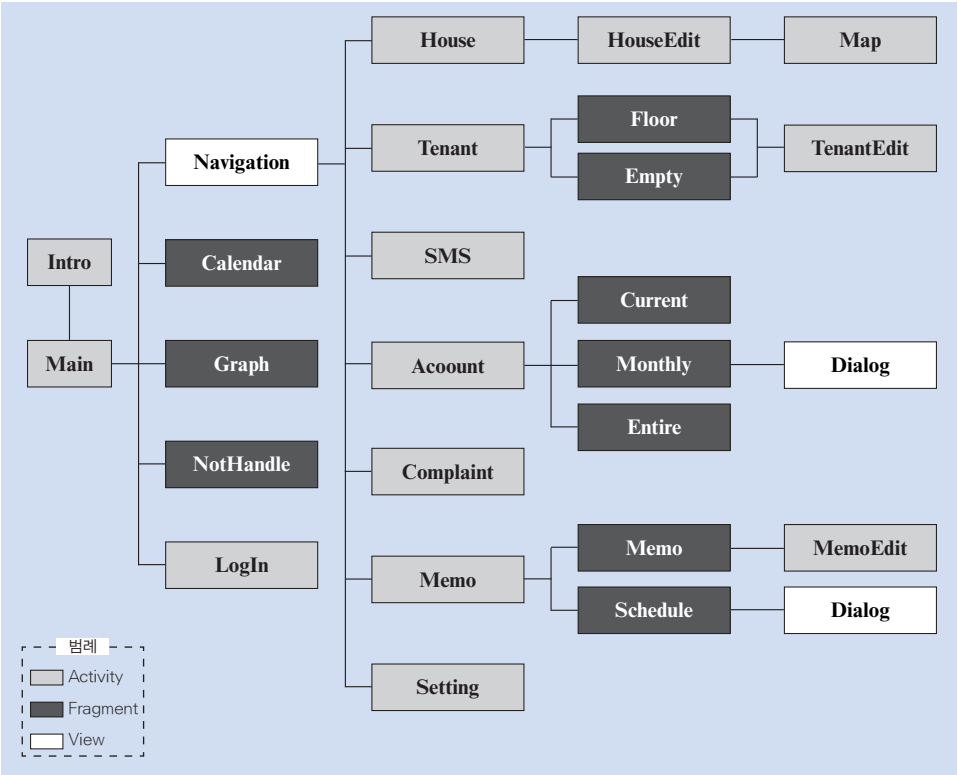
logo text : curt
typeface : DIN

HOMA



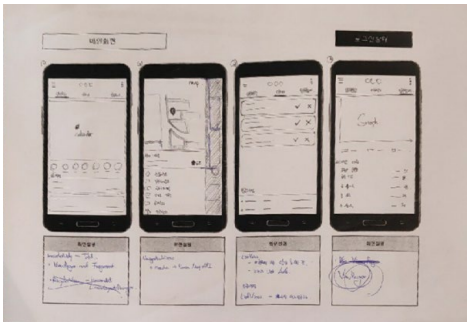
디자인 및 설계

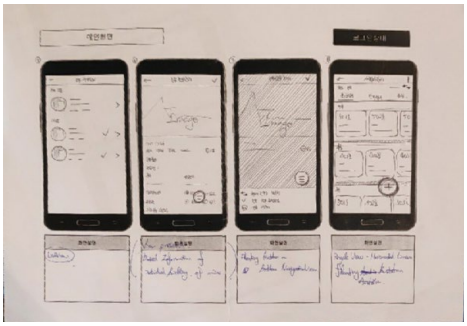
Activity 구조도

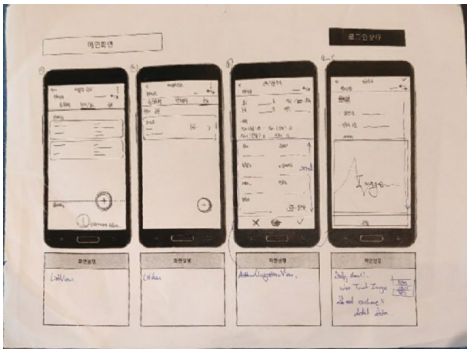


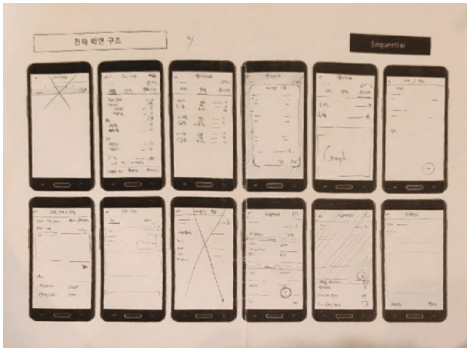
디자인 및 설계

와이어 프레임










HOMA APP - ver 1.0
ImageView
건물의 전경을 보여주는 ImageView로,
클릭시 변경 또는 등록 할 수 있다.
해당 건물의 상세정보를 표시한다.
FloationActionButton.
확인을 누르면 편집된 정보로 갱신한다.
해당 건물정보를 모두 삭제하는 option menu
ImageView
현재 등록된 주소를 기반으로 map을 통해서
보여주는 기능
시스템명 Android Mobile Phone & Tablet
서비스 구분 사용자(Client)
개발방법 NativeApp
화면 구분 HouseEditActivity
9 / 20

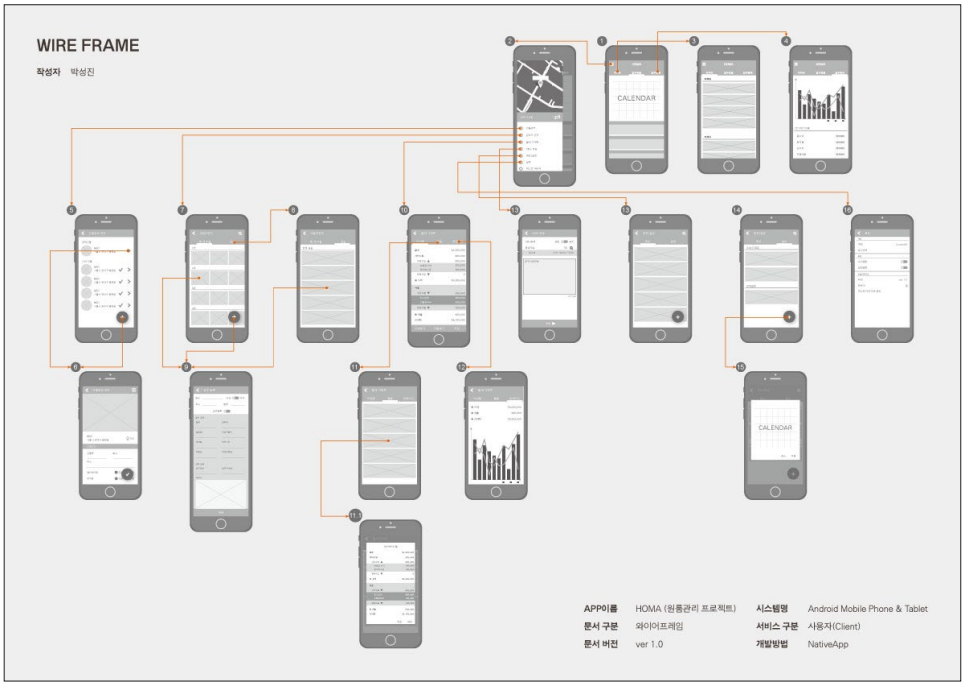

HOMA APP - ver 1.0
ListView
현재 건물로 설정되지 않은 다른 건물들의
리스트를 보이준다. 체크를 누르면 현재
건물로 변경되며 화살표를 누르면
상세편집 화면으로 넘어간다.
현재 건물로 설정된 건물의 간략한 정보를
보여주며 화살표를 누르면 상세편집 화면으로
넘어간다. 또한 기존에 있던 리스트에
더해서는 정보를 모두 표시한 상태로
보여준다.
FloationActionButton.
누르면 새로운 건물을 추가할 수 있는 상세
편집화면으로 넘어간다.
시스템명 Android Mobile Phone & Tablet
서비스 구분 사용자(Client)
개발방법 NativeApp
화면 구분 HouseActivity
8 / 20


HOMA APP - ver 1.0
NavigationView - header (NaverMap)
네이버 Map 을 이용해서 사용자가 등록한
현재 건물의 위치를 표시한다.
NavigationView - menu
세부 메뉴들이고 각 Activity로 이동하는
컴포즈 역할
언제든지 사용자의 건물을 중에 다른 건물로
변경이 가능하게 한다.
HouseActivity로 이동
시스템명 Android Mobile Phone & Tablet
서비스 구분 사용자(Client)
개발방법 NativeApp
화면 구분 MainActivity - NavigationView
5 / 20


HOMA APP - ver 1.0
ListView
선택된 사람들의 리스트를 보여준다.
EditText
문자의 내용을 쓰는 공간이며 SMS로 한정한
기능이므로 글자수 제한이 있다.
Switch.
일반 선택시 번호가 등록된 세입자들 중에서
몇명을 선택해서 보낼 수 있고, 공지 선택시
번호가 등록된 모든 사람들이 선택된다.
시스템명 Android Mobile Phone & Tablet
서비스 구분 사용자(Client)
개발방법 NativeApp
화면 구분 SMSActivity
16 / 20


HOMA APP - ver 1.0
MPAndroidChart.
이번달의 수입과 이번달의 지출, 순이익을
그래프화해서 보여준다.
해당 월에 대한 정보를 그래프 뿐만아니라
텍스트로도 보여줌으로서 조금 더 정확한
정보를 제공한다.
시스템명 Android Mobile Phone & Tablet
서비스 구분 사용자(Client)
개발방법 NativeApp
화면 구분 MainActivity - GraphFragment
7 / 20

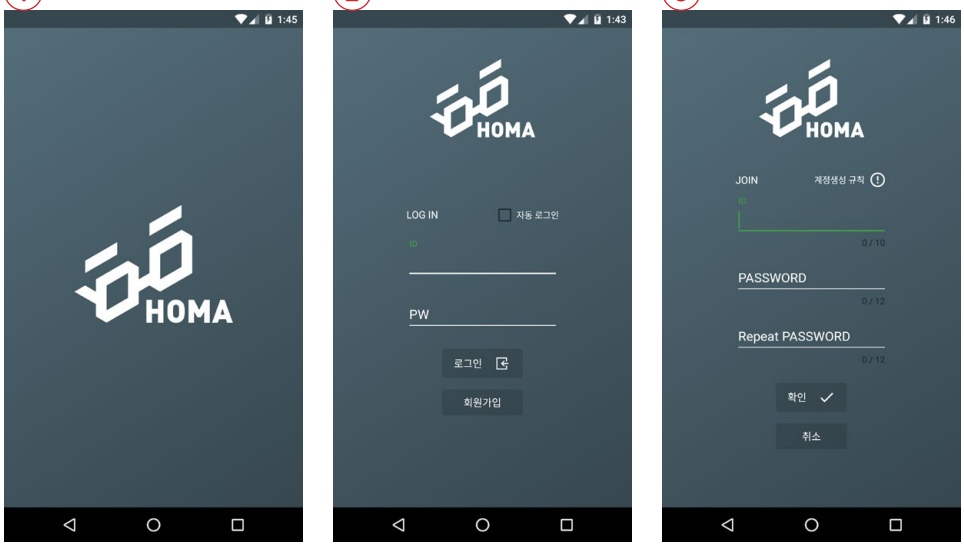

HOMA APP - ver 1.0
ListView
고정수입의 내역을 보여준다.
BottomNavigationView
수입이나 지출을 추가해서 계산할 수 있게
한다. 또한 현재의 화면을 스크린샷으로
저장하게 한다.
ListView
고정지출의 내역을 보여준다.
시스템명 Android Mobile Phone & Tablet
서비스 구분 사용자(Client)
개발방법 NativeApp
화면 구분 AccountActivity - CurrentFragment
13 / 20


WIRE FRAME
작성자 박성진
CALENDAR
CALENDAR
APP이름 HOMA (원룸관리 프로젝트)
문서 구분 와이어프레임
문서 버전 ver 1.0
시스템명 Android Mobile Phone & Tablet
서비스 구분 사용자(Client)
개발방법 NativeApp

화면 구성

IntroActivity &
Login Activity

① ② ③

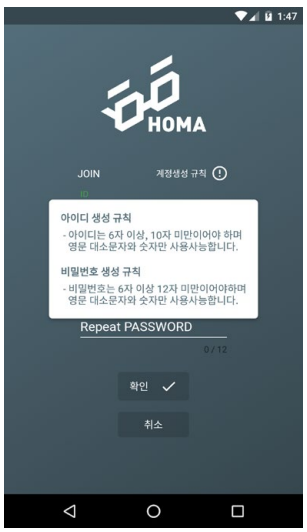


화면 기술 설명

1. IntroActivity의 onCreate에서 SharedPreference의 저장된 자동 로그인 여부를 확인한다. 만약 자동 로그인 상황이라면 특정 회원의 정보를 서버의 DB에서 가져오는 작업을 하고 MainActivity를 실행시킨다. 만약 자동로그인 상황이 아니라면 서버에서 회원의 명단을 가져와서 LoginActivity를 실행시킨다.

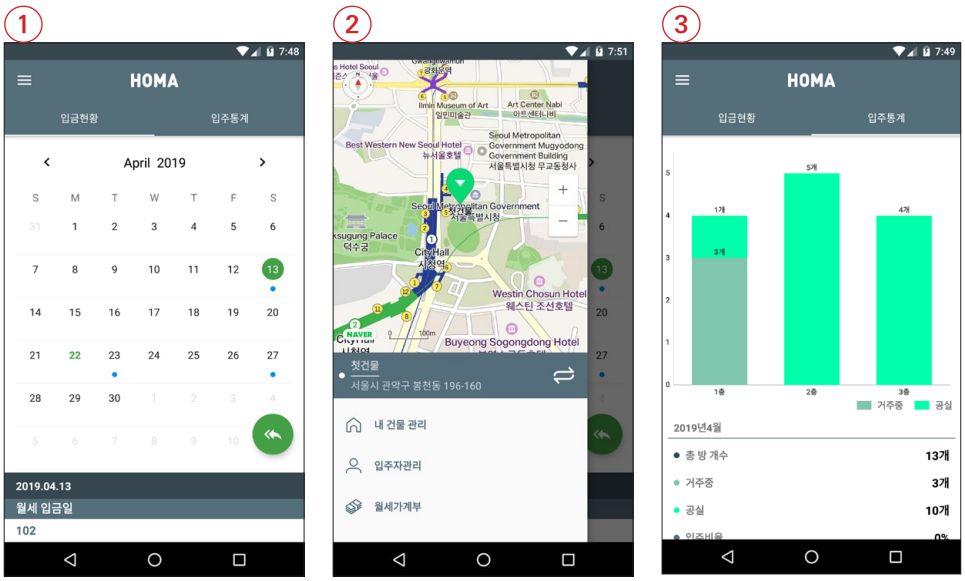
2. IntroActivity에서 전달받은 회원들의 리스트를 이용해서 아이디 비밀번호를 검증한다. 회원가입 버튼을 누르면 로그인에 관한 view들이 animation되어 viewport 바같으로 밀려나게 되고, 회원가입에 관한 view들이 화면의 왼쪽에서 나오게 된다.
  로그인에 성공하면 해당 회원의 정보를Server에서 가져오는 작업을 진행하고 MainActivity를 시작한다.

3. 아이디 생성규칙에 부합하는지 유효성 검사를 진행하고 물음표를 터치하면 view가 popup 되어 규칙을 보여준다. 유효성 검사가 끝나면 server에 회원을 등록하고 로그인에 관한 view들이 다시 나타난다.



화면 구성

MainActivity



화면 기술 설명

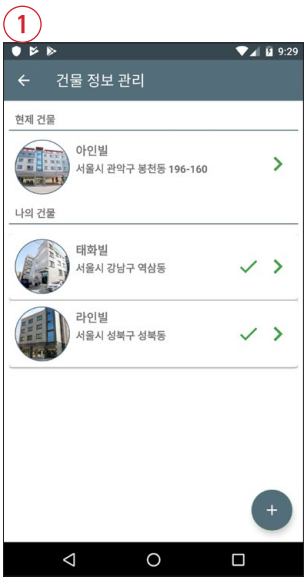
1. Main Activity의 onCreate에서 Fragment Pager Adapter를 통해 각 Fragment의 onViewCreate가 호출된다. public static으로 만들어진 변수들에 회원의 정보를 모두 보관하고 있는 class에서 달력에 표시될 정보를 받아온다. 날자별로 일정의 갯수를 확인하기 위해 반복문 처리하여 각 날자에 해당하는 데이터를 편집한다. 이후 일정의 개수에 따라 다른 아이콘을 배치하여 일정의 양을 볼 수 있게 한다. 일정을 표시할 때 사용되는 일정 객체는 Factory Pattern을 이용해서 생성시에 parameter에 따라 다른 기본값(다른 메모)을 가지게 만들면서 composite pattern으로 각기 다른 객체들을 손쉽게 달력에 표시할 수 있게 하였다. 달력의 날자를 선택하면 해당일의 일정이 밑에 위치한 ListView에 표시되고 Floating Action Button이 pop up되어 한번의 터치로 현제 날자로 돌아갈 수 있게 한다.

2. Navigation View의 Header View에 Naver의 외부 라이브러리인 NaverMap을 배치하여 사용자가 등록한 임대건물의 위치를 보여준다. Map Fragment에 onMapReady라는 call back method를 전달하고 parameter로 전달된 naver map객체를 통해서 map을 control한다. 지도의 하단에는 각 메뉴로 향하는 리스트들이 배치된다.

3. MPandroidChart 라는 Library를 사용하여 Graph를 만들고 건물의 임대상태에 대한 정보를 가져와서 표현한다, Activity가 resume되면 chart는 animate된다. 또한 상세 정보는 하단에 Text형태로 표시한다.

화면 구성

HouseActivity &
HouseEditActivity



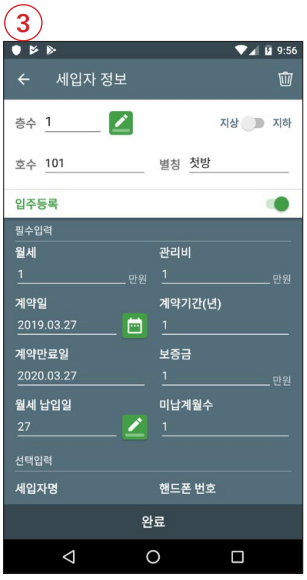
화면 기술 설명

1. 가지고 있는 건물의 목록을 ListView를 통해서 보여준다. 현제건물로 설정되면 모든 정보들이 현제 건물의 정보로 변환된다. 건물리스트의 우측에 있는 화살표를 누르면 특정 건물의 정보를 직렬화하고 intent를 통해서 HouesEditActivity로 전달한다.

2. 전달받은 정보를 기반으로 화면에 모든 정보를 표시한다. 이미지는 url 경로만을 제시하고 Glide Library를 통해서 보여주고 편집까지 한번에 할 수 있다. 새로운 건물을 등록하는 경우와 기존의 것을 편집하는 경우를 확인해서 toolbar의 삭제 menu를 보이게 할지를 결정한다.

화면 구성

TenantActivity &
TenantEditAcrivity



화면 기술 설명

1. recylerView를 이용해서 각각의 층을 만들고 그 안에서 또 다시 recylerView를 통해서 각각의 방을 표시한다, 이를 통해 사용자로 하여금 층/ 호수별로 방을 한눈에 볼 수 있게 한다. Floating Action Button을 통해서 새로운 방을 등록할 수 있게 한다.

2. 현제 방 중에서 공실만을 골라서 정보를 가져와서 ListView에 보여준다. 공실의 등록 아이콘을 누르면 바로 등록하는 화면으로 넘어간다.

3. 방의 정보와 세입자들의 정보를 등록, 수정할 수 있는 Activity로서, 각 방의 정보를 모두 표시한다. 층수를 입력하는 버튼을 누르면 AlertDialog가 보여지고 그 안의 NumberPicker를 통해서 숫자를 선택하게 되는데 이때의 선택가능한 범위는 현제 건물의 층수이다, 달력모양의 아이콘을 누르면 DatePickerDialog를 만들어서 날자를 선택할 수 있게 한다. 입주등록 Switch를 통해서 상세정보 입력영역이 보여질지, 말지를 결정한다.

Tenent Activity의 onDestroy를 override 하여 server에 현제의 데이터로 update를 요청한다.

화면 구성

TenantActivity &
TenantEditAcrivity

화면 기술 설명

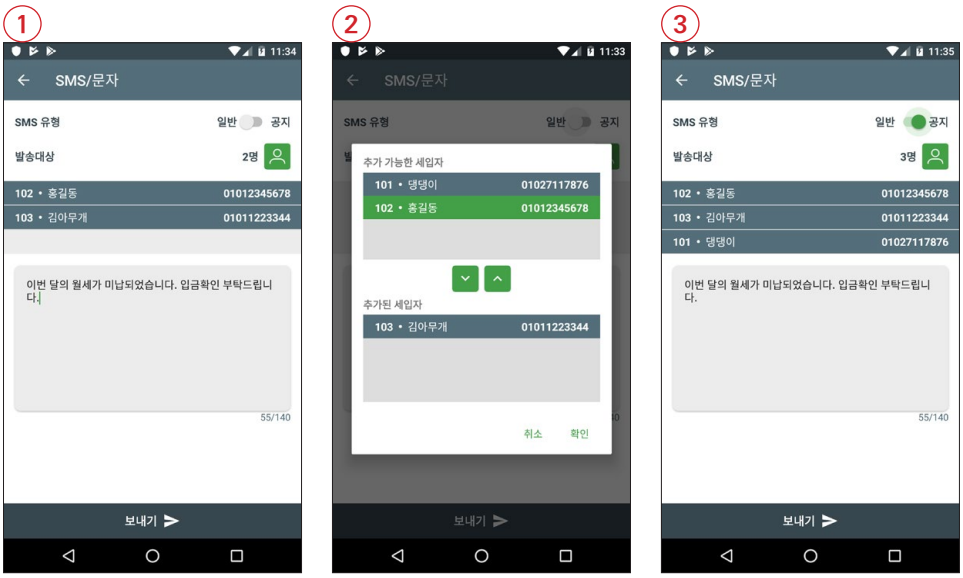
1. 이번 달의 수입과 지출을 한눈에 볼 수 있는 화면이다. 전체 세입자들의 월세를 모두 더해서 월세수익을 표시하고 기타 수입, 기타 지출을 표시한다. Bottom navigation Menu를 통해서 수입과 지출을 추가, 수정 할 수 있다, 화살표 모양의 토글버튼을 이용해서 세부 내용의 visiblity를 설정한다. 또한 세부내용의 개수가 추가/ 제거 될때 마다 ListView의 Height값을 변경하여 효과적인 화면 분배를 가능하게 한다.

2. Bottom Navigation Menu를 선택하면 뜨는 Dialog이며, 지출 / 수입의 유형을 선택할 수 있고 내용과 그에 해당하는 액수를 입력하고 추가버튼을 누르면 List에 추가된다.

3. BroadCast Reciver를 통해 날자가 변할 때 Service를 실행하고 달이 바뀌었을 때 그전의 데이터를 하나의 객체로 저장하고 새로운 객체를 현제 달로 설정한다, 이렇게 쌓인 데이터들을 종합하여 어플을 설치한 이후의 수입 지출을 Graph로 보여준다.

화면 구성

SMSActivity



화면 기술 설명

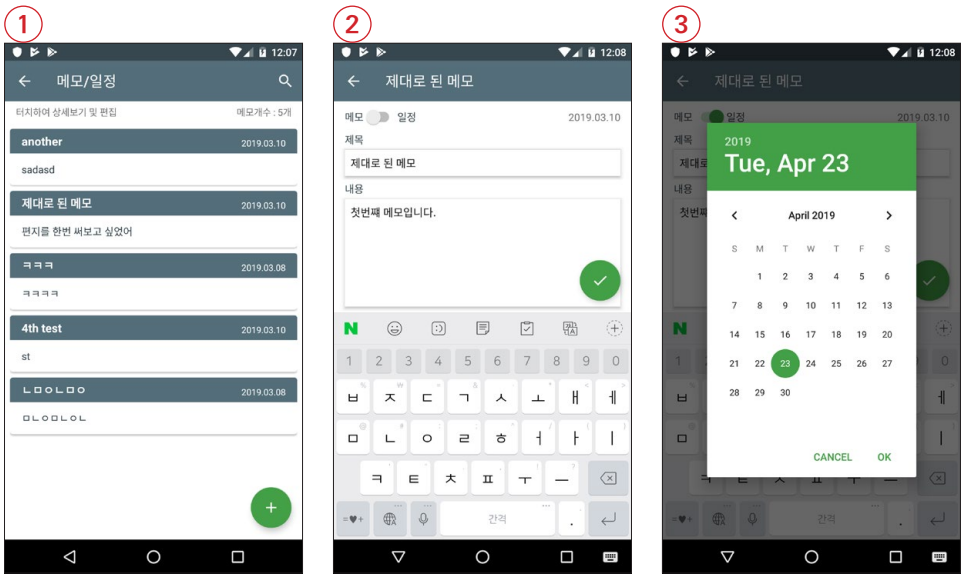
1. 세입자들을 편리하게 선택하고 문자메세지를 보낼 수 있는 화면이다. 보내기 버튼을 누르면 묵시적 intent를 통해서 SMS메세지전송 화면으로 이동한다. 일반모드는 보내는 사람들을 직접 선택해서 보내는 형식이며, 공지모드는 모든 세입자들이 자동으로 추가된다.

2. 발송 대상을 누르면 AlertDialog를 띄우고 선택 가능한 인원과 선택된 인원을 보여준다. 각각의 아이템이 boolean변수로 선택되었는지를 저장하고 선택된 경우, view의 background를 변경한다. 선택된 List의 아이템을 위로 올리거나 아래로 내리는 방식을 통해 보내는 사람을 선택한다.

3. 공지모드를 선택했을 때 모든 세입자들이 선택되어 보여진다.

화면 구성

MemoActivity &
MemoEditActivity



화면 기술 설명

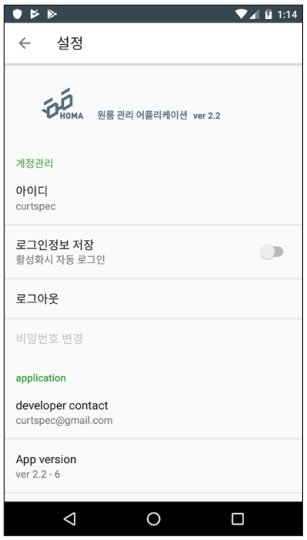
1. 모든 메모를 가져와서 ListView에 표시한다. FloatingActionButton을 눌러서 새로운 메모를 등록한다. Toolbar의 SerchView 를 통해서 메모를 검색하고 볼 수 있다.

2. 메모의 내용을 보거나, 수정 할 수 있는 화면이다. Edit text에 TextChangeListener를 추가하고 내용이 수정되면 Floating Action Button이 pop up되어 누를 수 있게 된다.

3. 일정모드로 바꾸게 되면 DatePickerDialog가 생성되어 특정 날자를 선택 할 수 있게 된다. 메모모드는 기본적으로 현제 날자를 가지고 있으며, 추가된 메모는 Main화면의 달력에도 표시되어 보여진다.

화면 구성

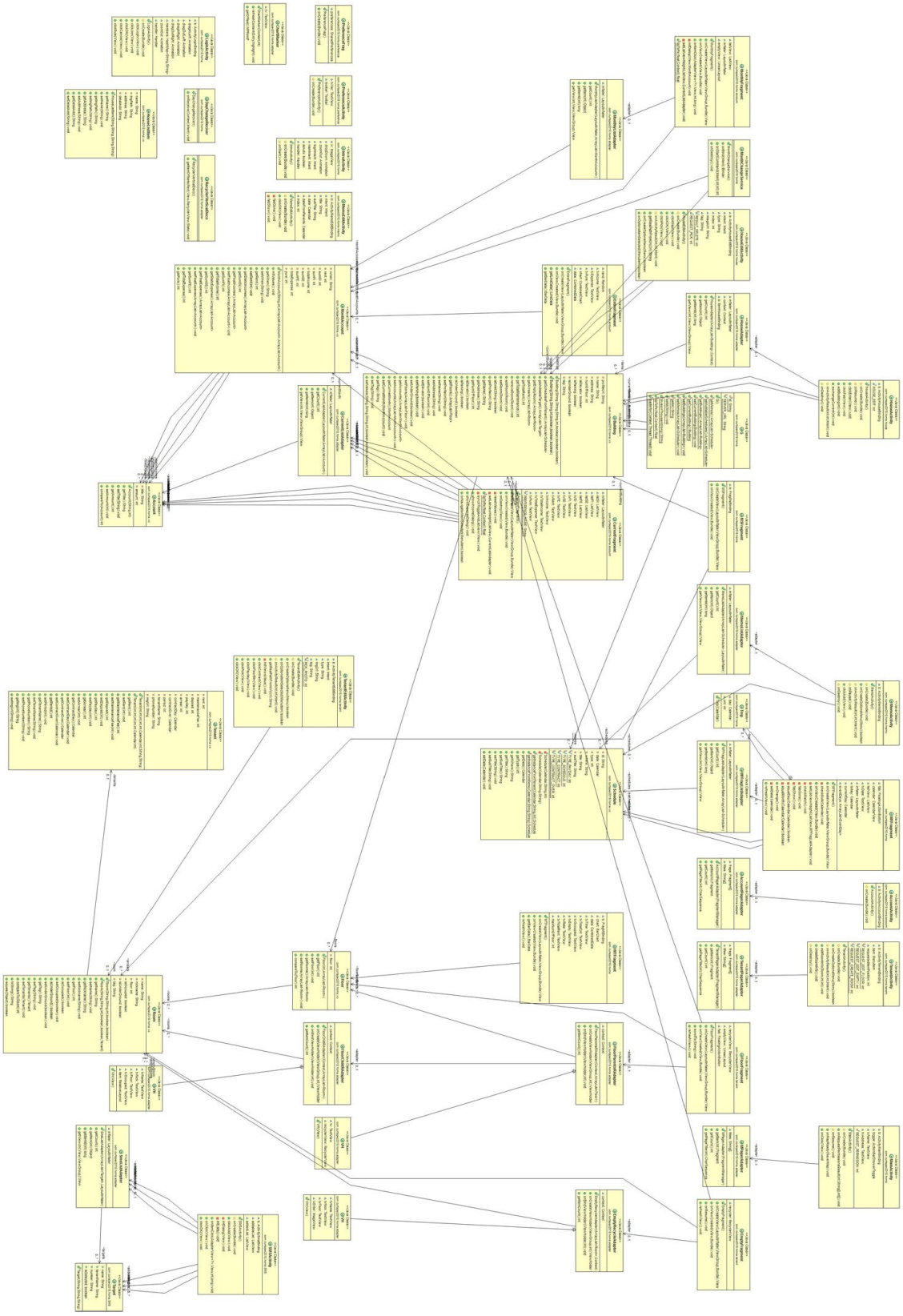
SettingActivity



화면 기술 설명

preferenceFragment를 상속받은 Class를 정의하고 Activity에 배치한다. Preference에 들어가는 내용은 xml로 만들어서 화면의 요소를 구성한다.

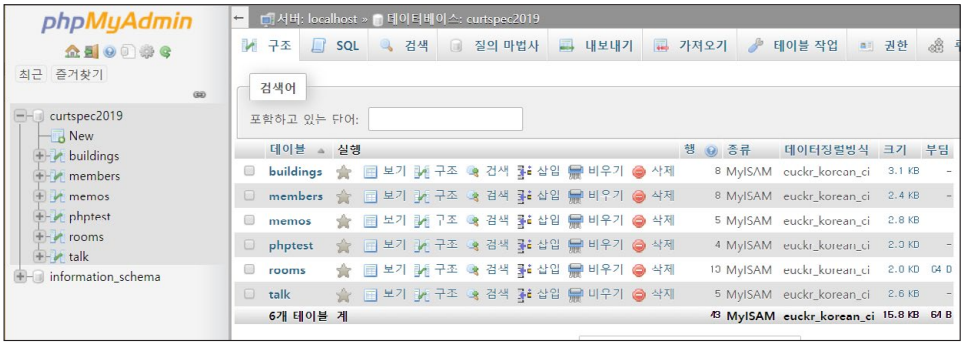
Class Diagram

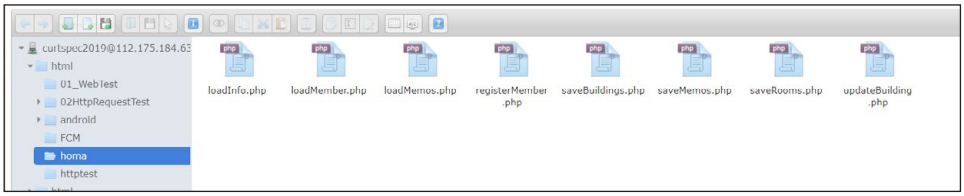


Link

https://firebasestorage.googleapis.com/v0/b/fir-storage-2ef34.appspot.com/o/diagram.jpg?alt=media&token=d523ea70-d724-4baf-99d6-903ae41bfe63

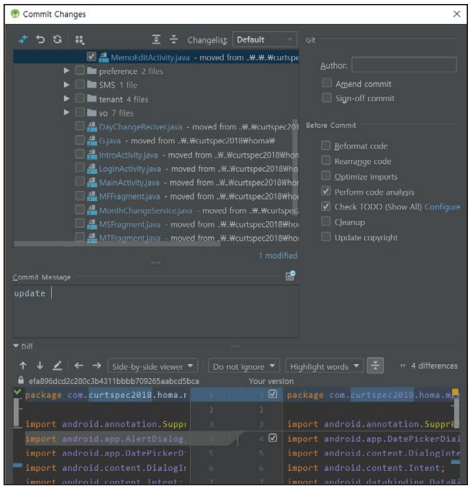
호스팅

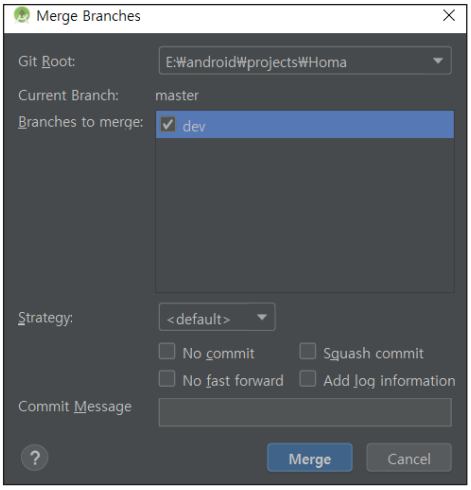


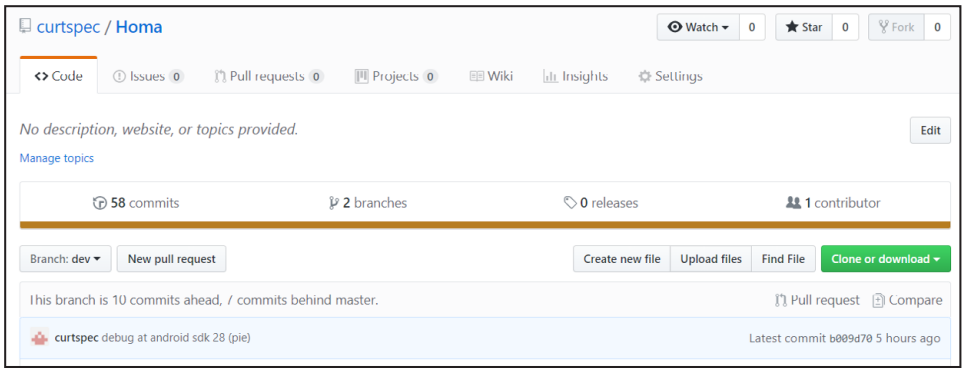


닷홈 무료 호스팅 업체를 선택하여 사용했습니다. 닷홈에서 제공하는 server에는 Apache, Mysql, php interperter가 기본적으로 설치 되어 있습니다. WebFTP에 필요한 php문서들이 저장되어 있으며, DB에는 이미지의 경로가 저장됩니다. php문서에서 MySQL과의 연동을 진행하고 질의를 통한 데이터의 CRUD를 빈번하게 작업하였습니다.

형상관리



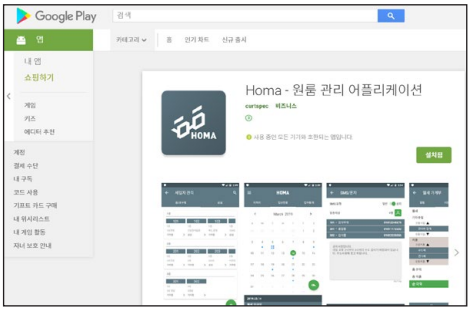




project를 진행하는 과정에서 commit, merge, 그리고 branch를 이용한 작업 등 버전관리에 관한 모든 작업은 android studio내부에서 이루어졌습니다. 또한 Cloud 기반 SCM을 위해 Github를 이용하여 여러 local에서 작업이 가능하도록 하였습니다.

Google Play store 출시

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.curtspec2019.homa"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 7
        versionName "2.3"
        testInstrumentationRunner "android.suppo
    }
```



build.gradle(앱 모듈)의 versionCode를 수정하고 generate Signed APK를 통해서 abb 또는 apk 생성한 후에 Google의 play store consol을 통해서 application을 업로드한다.
새버전 출시가 완료되면 검색을 통해 다운로드 할 수 있습니다.

Link https://play.google.com/store/apps/details?id=com.curtspec2018.homa

프로젝트 기술서

프로젝트 개요

프로젝트 명 Web page제작 – outback

개발 기간 2019. 04.08 ~ 2019. 04. 12 (1주)

개발인원 5명 (박성진, 정지헌, 이동훈, 추헌동, 김형우)

담당역할 팀장, 기획, 디자인, 개발 (header, footer 제작후 format배포)

개발환경

OS	Windows 7, 10
Development Tool	Visual Studio Code
Language	HTML, CSS, JavaScript
DBMS	–

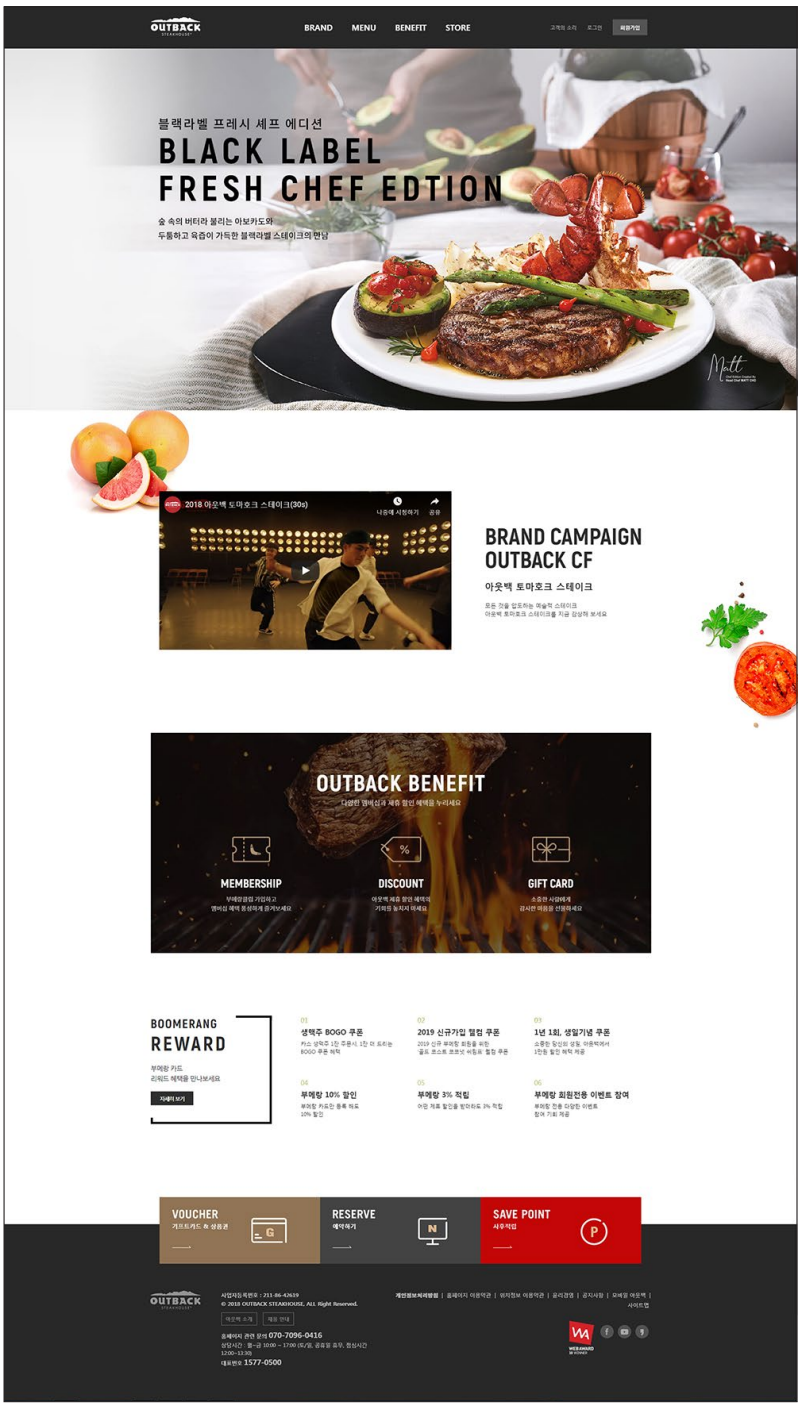
프로젝트 설명

이번 프로젝트는 새로운 컨텐츠와 새로운 디자인으로 만들어내는 것이 아니라, 기존의 컨텐츠와 디자인을 기반으로 한 웹사이트를 따라서 만드는 프로젝트이다. 모티브로 삼을 기존의 웹사이트는 아웃백의 사이트로 정했다. 아웃백 사이트에 있는 모든 기능과 메뉴를 구현하는것이 아니라 개개인의 제작 속도를 고려하여 만들 수 있는 메뉴와 페이지만을 구현한다.

기획서

화면 구성

index.html



화면 기술 설명

Semantic tag를 이용한 HTML작성후 CSS를 통해 Styling을 진행. 전체 Menu에 마우스를 올려놓으면 JQuery의 on()를 이용해서 숨겨져있던 세부메뉴들을 FadeIn, FadeOut 을 진행한다.

Main Image 슬라이드 영역은 크기를 설정하고 display를 row방향의 Flex로 설정하고 overFlow는 hidden으로 설정한다. 이후 JQuery의 animate를 이용해서 left 값을 변경시켜 에니메이션을 진행한다. 또한 setInterval을 이용해서 주기적으로 함수의 자동 호출을 진행한다.
FooterView에도 JQuery의 on 함수를 이용해서 animate를 걸어 마우스 이벤트에 반응하게 한다.

화면 구성

Brand_Story.html

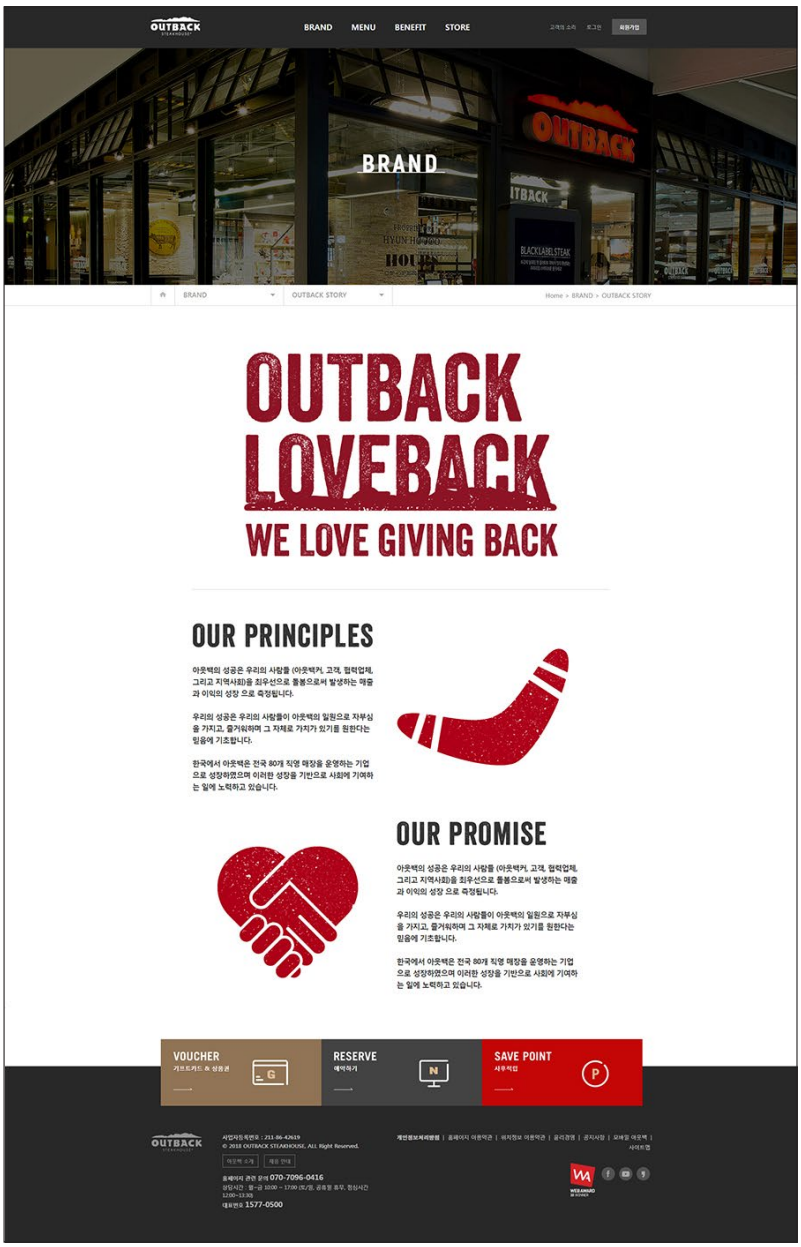
화면 기술 설명

Brand로고의 BackGround에 이미지를 넣어놓고 background-position을 fixed로 하여 스크롤을 내릴때에도 변화가 없게 설정한다.

중간에 dropdown되는 navigator에 JQuery의 on함수를 이용해서 SlideToggle을 실행한다. 글과 같이 있는 로고의 경우 Float속성에 left를 주어서 글과 어우러지게 배치될 수 있게끔 만들었다.

화면 구성

index.html



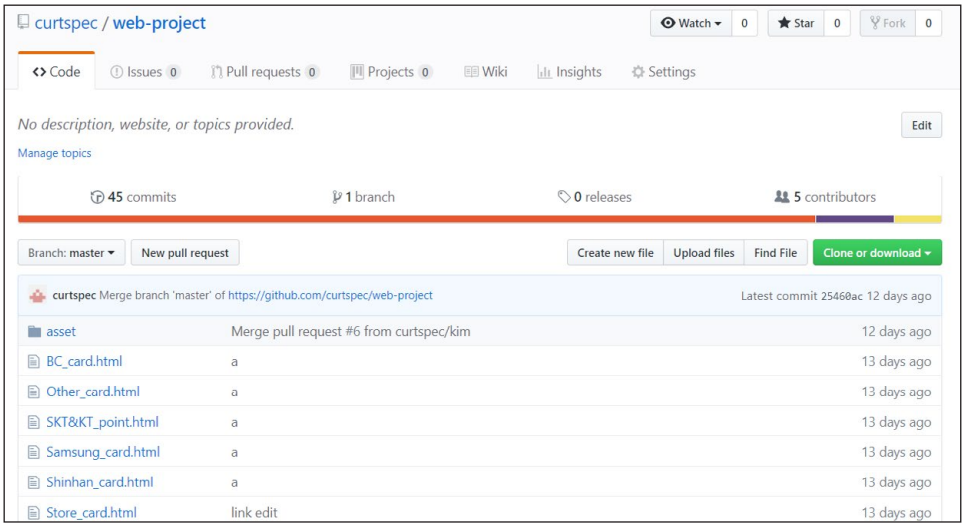
화면 기술 설명

Brand로고의 BackGround에 이미지를 넣어놓고 background-position을 fixed로 하여 스크롤을 내릴때에도 변화가 없게 설정한다.

중간에 dropdown되는 navigator에 JQuery의 on함수를 이용해서 SlideToggle을 실행한다. 글과 같이 있는 로고의 경우 Float속성에 left를 주어서 글과 어우러지게 배치될 수 있게끔 만들었다.

형상관리

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   asset/css/seongjin/csr.css

no changes added to commit (use "git add" and/or "git commit -a")
PS E:\web\web-project> git add asset/css/seongjin/csr.css
PS E:\web\web-project> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   asset/css/seongjin/csr.css

PS E:\web\web-project> git commit
[master 8bdeacc] Please enter the commit message for your changes. Lines starting
 1 file changed, 1 insertion(+)
PS E:\web\web-project>
```



여러명이 함께 진행하는 팀프로젝트의 특성상 형상관리의 중요성이 더욱 부각되었고 충돌이 발생하지 않도록 진행하였습니다. CUI환경에서 git 명령을 수행하며 github를 통해 push와 pull을 진행하였습니다. github에서 collaboration을 통해 팀원을 초대하고 수정권한을 부여하였습니다.