참빛설계학기 주차별 활동보고서

주 차	5주차

팀 명	편--안
팀 원	오윤제

프로젝트명	NoSQL-based PC Café Integrated Management System		
활동기간	2019.9.23 ~ 2019.9.27	활동시간	6시간
금주목표	Point 충전에 대한 내용 구현, Main Page에 대한 추가적 설계, 로그아웃, PC종료 Database에 대한 정리		

활동내용

1. Menubar 추가 구현

[Back-end]

- Realtime Database 기반의 정보 출력
  -> 현재 접속 중인/선택 중인 PC Cafe에 대한 출력
  -> `myRef.child(myUid).child("PCcafe").addValueEventListener` : Firebase Database 중 PCCafe에 있는 데이터 불러오기
  -> `override fun onDataChange(p0: DataSnapshot)` : 변경된 값의 Database를 자동호출
  -> `val value = p0?.value` : Textview에 출력 할 수 있게 해줌 (`"$value"`)

  -> 현재 접속 중인/선택중인 PC Cafe 지점에 대한 출력
  -> `myRef.child(myUid).child("local").addValueEventListener` : Firebase Database 중 local에 있는 데이터 불러오기
  -> `override fun onDataChange(p0: DataSnapshot)` : 변경된 값의 Database를 자동호출
  -> `val value = p0?.value` : Textview에 출력 할 수 있게 해줌 (`"$value"`)

  -> 현재 보유 중인 Point에 대한 정보 출력 및 충전 창 추가
  -> `myRef.child(myUid).child("point").addValueEventListener` : Firebase Database 중 point에 있는 데이터 불러오기
  -> `override fun onDataChange(p0: DataSnapshot)` : 변경된 값의 Database를 자동호출
  -> `val value = p0?.value` : Textview에 출력 할 수 있게 해줌 (`"$value"`)

[Front-end]

- Menubar.xml에 출력 Textview 및 Image Button 추가 [1]
  -> charge : Point 충전을 위한 Image Button
  -> showPCCafe/pc_cafe : 현재 접속한 PC Cafe를 보여주는 Textview
  -> local : 현재 접속한 PC Cafe의 지점을 보여주는 Textview
  -> showPoint/point : 현재 소유 중인 Point를 보여주는 Textview

2. Point 충전 Page [Back-end]

- 충전 버튼에 대한 구현

-> `val database : FirebaseDatabase = FirebaseDatabase.getInstance()`

-> `val myRef : DatabaseReference = database.getReference("member").child(auth.currentUseruid.toString()).child("point")` : 각 유저의 point Database 가져오기

-> `override fun onDataChange(p0: DataSnapshot)` : 변경된 값의 Database를 자동호출

-> `val value = p0?.value`
`sum = value.toString().toInt() + point.text.toString().toInt()`
`myRef.setValue(sum.toString())`
`point.setText("0")` : 현재 소지하고 있는 Point에 충전하는 Point 추가

-> 버튼 클릭 후 입력된 값 (Point) 만큼 충전

- 충전 Log 기록 남기기

-> `val current = LocalDateTime.now()`
`val formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss")`
`val formatted = current.format(formatter)`
`val logRef : DatabaseReference = database.getReference("log").child(formatted.toString()).child(auth.currentUseruid.toString())`
`logRef.setValue(point.text.toString().toInt().toString() + " point")` : 충전한 Log 기록은 Database에 저장

3. Point 충전 Page [Front-end]

- Cardview [2]

-> Point 입력 Textview 추가

-> 충전을 위한 Image Button 추가

4. PC 종료 [Back-end]

- PC 종료 메뉴 선택 Case에 대한 결과
- PC가 접속되어 있는 상태 확인

-> var pc_check 변수 선언 후 10000으로 초기화, 초기 예외 처리를 위한 변수 설정

-> 접속되어있을 때 (pc_check 값이 10000 일 때 = 앱에 처음 들어갔을 때) PC 종료 버튼을 누른 후 Database 변동 : 사용 여부 path using = "X"
접속자 정보 path uid = ""
시작 시간 path time_start = ""
사용 중인 좌석 path seat_using = ""

-> PC 로그아웃 Toast 출력 후 pc_check을 0으로 변경

5. 로그아웃 [Back-end]

- 로그아웃 메뉴 선택 Case에 대한 결과
- `mAuth.signOut()` : 을 사용한 계정 sign out

-> Firebase에 정의된 함수 사용

-> 로그아웃 성공 Toast 출력

-> MainActivity로 창 바꿈

6. 즐겨찾기

[Back-end]

- favorite path에 들어간 Database 호출
  -> `val database: FirebaseDatabase = FirebaseDatabase.getInstance()` : Firebase Database 사용
  -> `val memberRef = database.getReference("member").child(auth.currentUseruid.toString()).child("favorite")` : uid가 매치하는 member의 favorite path에 들어가는 Database 호출 (즐겨찾기에 목록에 넣어진 PC Cafe List)

- 반복문을 통한 onDataChange 확인
  -> List를 view로 만들 adapter를 생성 (여기에 어댑터 선언하는거 넣으주시면 될 듯)
  -> `adapter = ArrayAdapter(this, android.R.layout.simple_list_item_single_choice, dataList)` : Adapter 선언을 통한 List 항목에 기능 추가
  -> for문을 사용하여 즐겨찾기에 있는 key값 수만큼 반복문 동작
  -> 해당 key값을 list의 데이터로 추가
  -> iterator를 사용한 key값 호출
  -> List에 변화 있을 시 adapter에 데이터의 최신화를 알림

```
override fun onDataChange(p0: DataSnapshot) {
    var value = p0.children.asIterable()
    var iter = p0.children.asIterable()
    for(i in 0..p0.childrenCount.toInt()-1) {
        dataList.add(iter.iterator().next().key.toString())
        adapter?.notifyDataSetChanged()
        value = iter
        iter = value.asIterable()
        }
    }
```

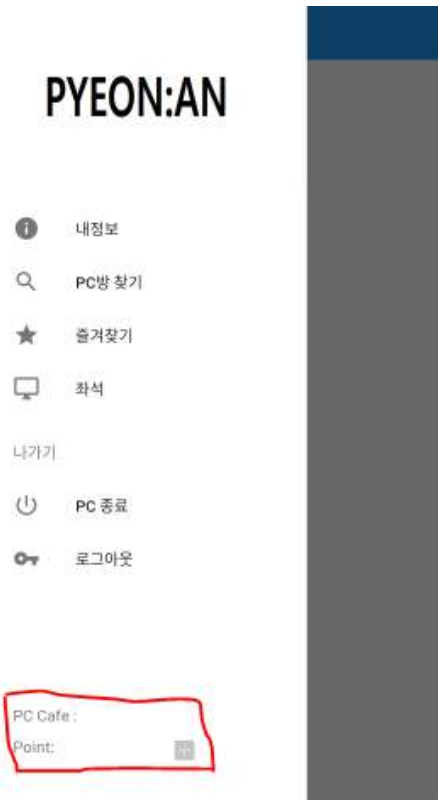
- List내 삭제 기능
  -> lstMain : 선택된 아이템에 대한 기능을 위한 변수
  클릭시 선택된 PC Cafe의 이름이 Toast를 통해 출력
  -> select 버튼을 통한 삭제할 PC Cafe 선택
  -> delete 버튼을 클릭해 선택된 PC Cafe 삭제
  -> `myRef.child(pccafe!!.toString()).removeValue()` : Firebase의 favorite 항목의 내용 삭제

[Front-end]

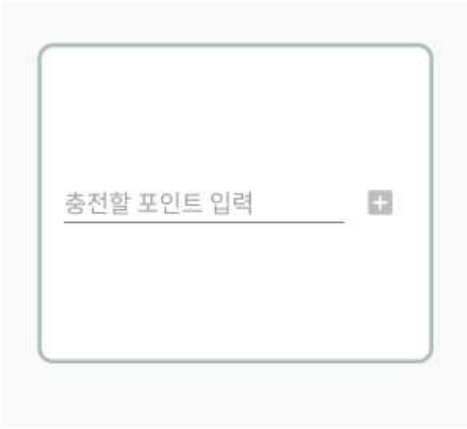
- LinearLayout 생성
- ListView 작성
- 선택 버튼 및 삭제 버튼 추가
- 초기 버전의 즐겨찾기 페이지 [3]
  -> 추후 UI 변경 예정
- Swipe and Delete (밀어서 삭제 시도)
- Swipe에 대한 Kotlin 코드 검색

활동사진

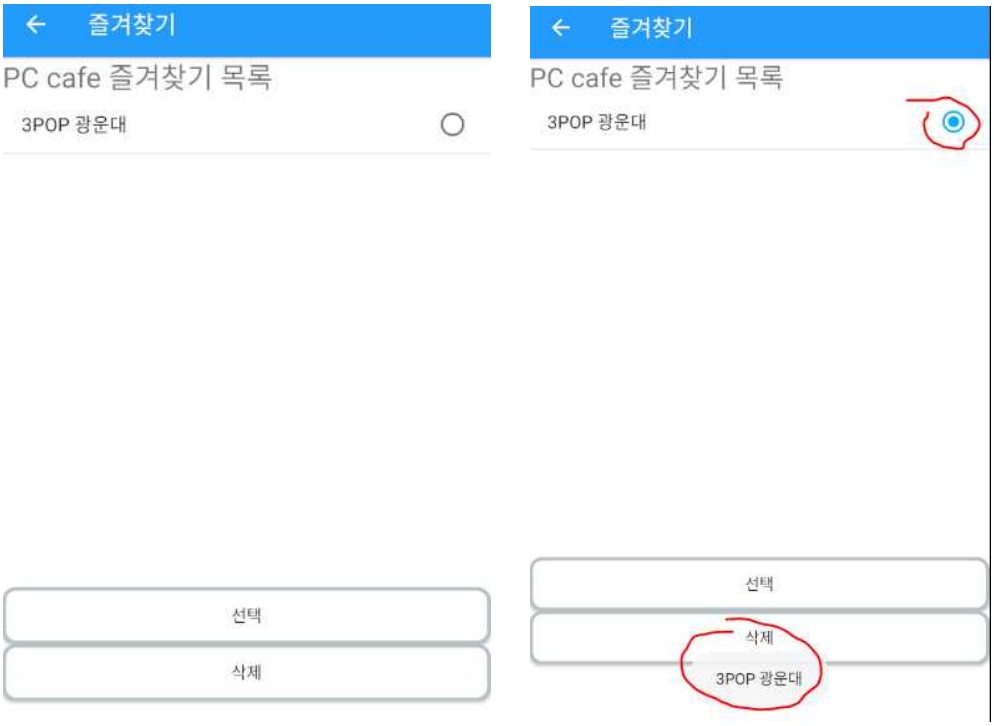
[1]

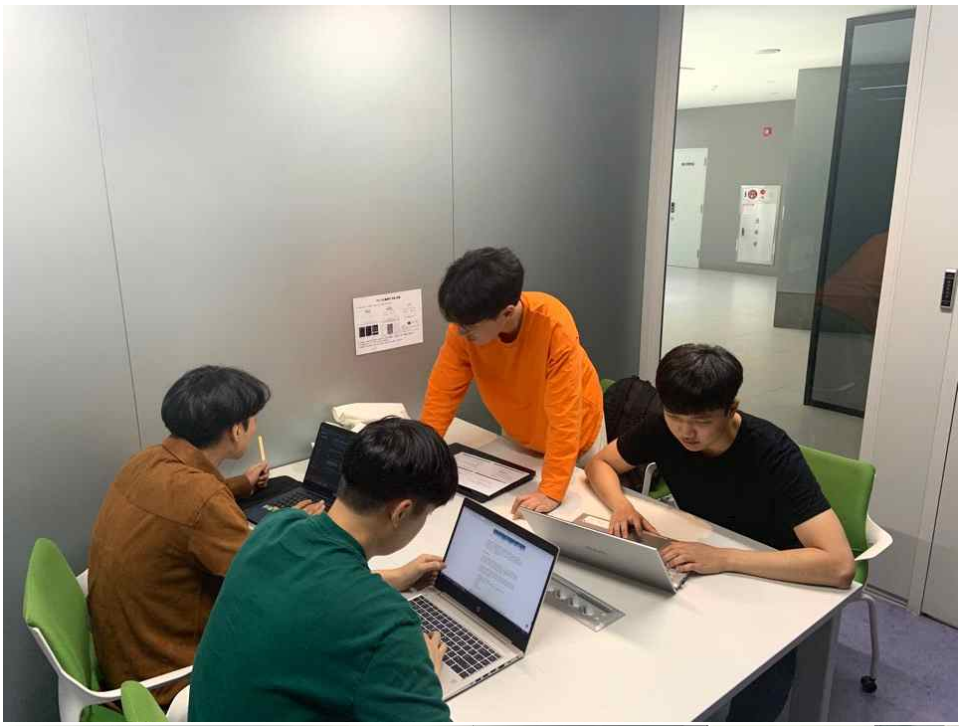
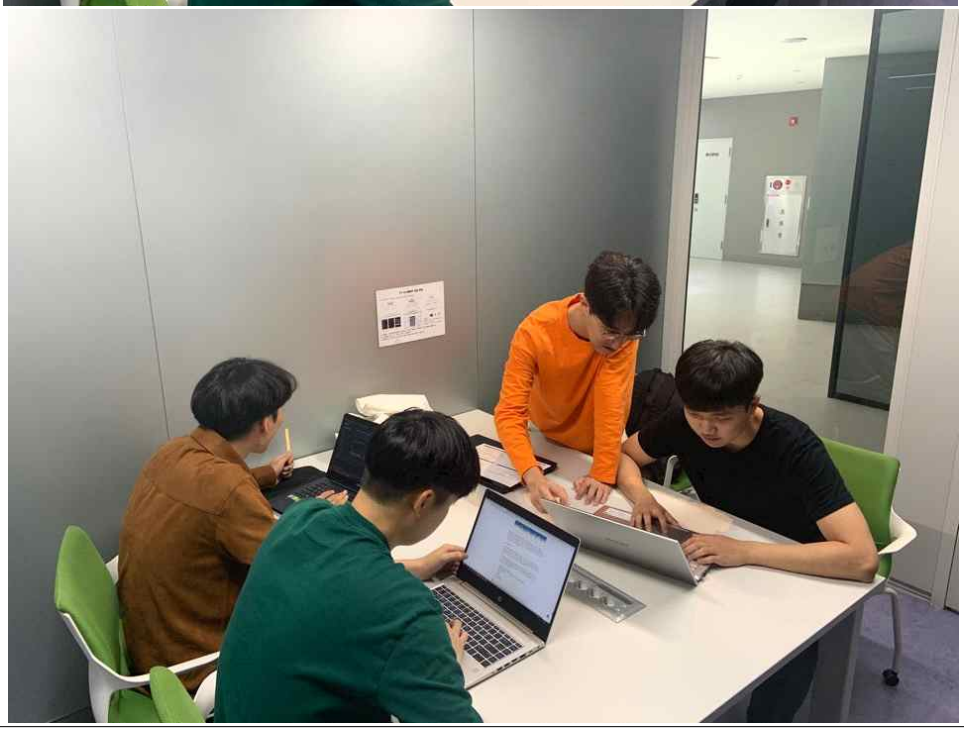

PYEON:AN
내정보
PC방 찾기
즐겨찾기
좌석
나가기
PC 종료
로그아웃
PC Cafe :
Point:

[2]


충전할 포인트 입력

[3]


즐겨찾기
PC cafe 즐겨찾기 목록
3POP 광운대
선택
삭제
즐겨찾기
PC cafe 즐겨찾기 목록
3POP 광운대
선택
삭제
3POP 광운대

	주간 활동 및 회의 사진  
문헌자료/ 참고자료	(열혈코딩) 안드로이드 스튜디오로 만나는 코틀린 : 안드로이드 스튜디오와 코틀린을 이용한 이지 코딩 - 박성완 Firebase로 안드로이드 SNS 앱 만들기 - 하울
차후 계획	좌석 선택창 구현 및 선택된 좌석에 대한 Database 설정 및 Application 보수
지도교수 총평	

2019. . .

지도교수 (인)