식단플래너 설계서



|  |  |  |
| --- | --- | --- |
| 과목 | 소프트웨어 공학 | |
| 학과 | 컴퓨터공학과 | |
| 제출일 | 2022.05.06 | |
| 분반 | 002분반 | |
| 학번 및 성명 | A2조 | |
| 20173023 | 황진호(조장) |
| 20183073 | 이상원 |
| 20203001 | 하상인 |
| 20203032 | 송현섭 |

* 목차

1. **설계서 내용**
2. **클래스 다이어그램 : 전체**
3. **클래스 명세서** 
   1. **영양성분조회기능**
   2. **물알람기능**
   3. **식단관리기능**
   4. **식단정보화면**
   5. **물정보 화면**
   6. **물관리기능**
   7. **사용자정보(수정,삭제,조회,계산)**
   8. **사용자정보 수정 화면**
4. **시퀀스 다이어그램**
   1. **영양성분조회 시퀀스다이어그램**
   2. **식단관리기능 시퀀스다이어그램**
   3. **물관리기능 시퀀스다이어그램**
   4. **사용자 수정,삭제,조회,계산 시퀀스다이어그램**
5. **DB설계**

**1.설계서 내용**

코틀린 언어로 안드로이드 SDK를 사용하여 작성하기에 구글이 권장하는 디자인패턴인 MVVM+Android Architecture Components와 데이터베이스 관리를위한 Room 라이브러리를 이용하여 구현하는것으로 설계 하였습니다. 또한 외부API요청에는 서드파티인 Retrofit과 통신으로 okHttpClient 라이브러리로 설계하였습니다.

MVVM은 화면을 담당하는 View, 데이터가 저장되는 Model, 화면의 데이터SIUD의 요구를 처리하고 기능을 수행하는 ViewModel 으로써, 화면의 뷰에 들어온 데이터를 observer 패턴과 LiveData로 별도의 boilerplate를 줄여 즉시 ViewModel에게 데이터의 변경을 알리고 SIUD 및 기능을 처리합니다.

View는 싱글액티비티 패턴으로, 액티비티의 특성상 리소스가 무거워서 액티비티를 껍데기로 하나를 두고, 그위에 프래그먼트를 두어 메인화면의 역할과 제어권한을 넘겨줍니다. 이후 다수의 프래그먼트들로 부분화면을 구성하고 Navigation 라이브러리를 이용하여 화면을 전환합니다.

뷰의 전환이 일어나면 이전뷰는 메모리에서 제거되고 수명주기상 destroy 되기때문에 모든데이터를 ViewModel 인 컨트롤클래스에서 관리합니다.

이후 실제 데이터가 저장되는 곳은 Model 인 데이터베이스 입니다.

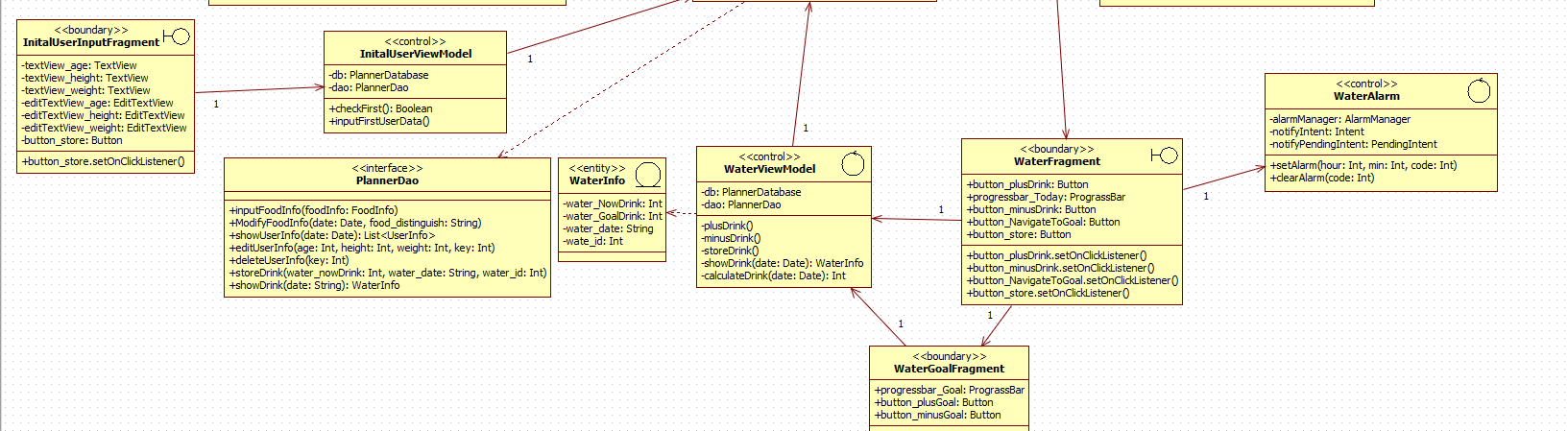
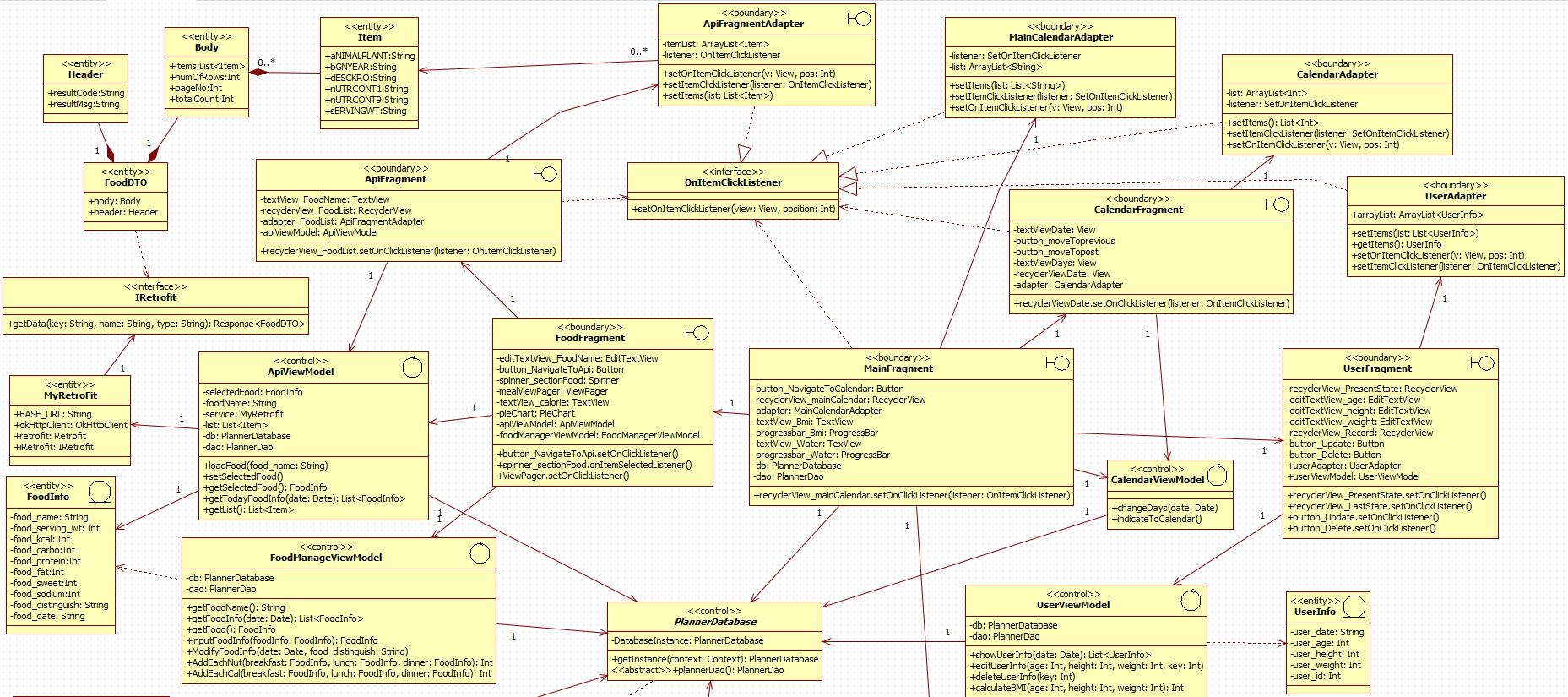
시스템은 개별 클라이언트와 데이터베이스로 구성되는 구조 입니다.

데이터베이스

클라이언트

서버와 로그인이 없는이유는 식단플래너 라는 주제의 특성상, 개인이 본인의 식단 및 BMI를 관리하기 때문에 서버와의 통신을 할 데이터가 없기 때문 입니다.

2. 클래스 다이어그램 (위에서부터 윗부분 아랫부분 입니다.)



3. 클래스 명세서

3.1 영양성분 조회 기능(진호)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스명 | | ApiViewModel | | |
| 클래스 개요 | | 공공데이터포탈의 서버에 데이터를 API로요청하여 응답받는 기능을 하는 클래스 | | |
| Stereotype | | <<Control>> | | |
| Public  Interface  Public | Operations | Operation(Parameter) | Return | 설명 |
| loadFood(food\_name : String) | String | 외부시스템으로 해당음식명을 가지고 영양성분조회를 요청하면 탄수화물,단백질,지방,나트륨,당,칼로리를 응답받는 메소드 |
| SetSelectedFood() | void | 선택된 음식정보를 selectedFood속성값으로 설정하는 메소드 |
| getSelectedFood() | FoodInfo | 영양성분조회 화면에서 선택된 식단에 관한 객체를 반환하는 메소드 |
| getTodayFoodInfo(date:Date) | List<FoodInfo> | 오늘에 해당하는 아침,점심,저녁의 등록된 식단정보를 가져오는 메소드 |
| getList() | List<Item> | API요청 결과로 나오는 해당 식단단명에 대한 모든연관식단들을 가져오는 메소드 |
| Attributes | Attribute | Type | 설명 |
| selectedFood | FoodInfo | 영양성분조회 화면에서 선택된 식단에 관한 객체 |
| foodName | String | 사용자가 입력한 식단이름을 담는변수 |
| service | MyRetrofit | 외부시스템에 api를 통해 요청하고 반환받은 객체 |
| list | List<Item> | API요청 결과로 나오는 해당 식단단명에 대한 모든연관식단들을 담는 변수 |
| db | PlannerDatabase | 데이터베이스 객체 |
| dao | PlannerDao | 데이터베이스에 사용할 쿼리문들이 담겨잇는 인터페이스 객체 |

3.2 물 알람 기능

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스명 | | WaterAlarm | | |
| 클래스 개요 | | 물 알람 기능을 하는 클래스 | | |
| Stereotype | | <<Control>> | | |
| Public  Interface  Public | Operations | Operation(Parameter) | Return | 설명 |
| setAlarm(hour:Int, min:Int, code:Int) | void | 시,분,초와 알람을 구분해주는 code값을 매개변수로 받아 알람을 설정해준다. |
| clearAlarm(code:Int) | void | code를 매개변수로 받아 설정된 알람을 제거해주는 메소드 |
| Attributes | Attribute | Type | 설명 |
| alarmManager | AlarmManager | 알람을 설정해주기위해 사용하는 클래스 객체 |
| notifyIntent | Intent | 알람을 설정하기위해 클래스와 클래스간의 관계를 지정해주는 시스템에서 제공되는 클래스로 어떤클래스에서 어떤클래스로 메세지를 보낼것인가를 결정함 |
| notifyPendingIntent | PendingIntent | 특정시간에 메세지가 송수신하도록 도와주는 클래스 |

3.3 식단 관리 기능 (상원)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스명 | | inputFood | | |
| 클래스 개요 | | 식단 관리 기능 | | |
| Stereotype | | Control Class | | |
| Public  Interface  Public | Operations | Operation(Parameter) | Return | 설명 |
| getFoodInfo() | FoodInfo | 사용자가 입력한 식단정보를 가져온다 |
| getFoodName() | String | 사용자가 입력한 식단이름을 가져온다. |
| inputFoodInfo() | unit | 식품정보 입력 기능 |
| ModifyFoodInfo() | unit | 식품정보 수정 기능 |
| AddEachNut(each\_nut\_breakfast:  FoodInfo, each\_nut\_lunch: FoodInfo, each\_nut\_dinner: FoodInfo) | Int | 아침,점심,저녁의 각각의 영양소 정보를 가져와서 모두 합산 혹은 감산 하여 계산된 영양소를 반환한다. |
| AddEachCal(each\_cal\_breakfast:FoodInfo,each\_cal\_lunch:FoodInfo,each\_food\_dinner:FoodInfo) | Int | 아침,점심,저녁의 각각의 칼로리 정보를 가져와서 모두 합산 혹은 감산하여 계산된 칼로리를 반환한다. |
| Attributes | Attribute | Type | 설명 |
| Food | FoodInfo | 식품정보 객체 |

3.4 식단정보 화면

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스명 | | FoodFragment | | |
| 클래스 개요 | | 식단, 칼로리, 영양성분이 보여지는 화면 클래스 | | |
| Stereotype | | Boundary Class | | |
| Public  Interface  Public | Operations | Operation(Parameter) | Return | 설명 |
| Spinner\_sectionFood.onItemSelectedListener() | void | 아침,점심,저녁을  선택할 수 있다. |
| Button\_NavigateToApi.  SetOnClickListener() | void | 식단을 입력할 수 있게 해준다. |
| ViewPager.  setOnClickListener() | void | 식단을 수정할 수 있게 해준다. |
| Attributes | Attribute | Type | 설명 |
| EditTextView\_FoodName | EditTextView | 식품검색을 할 수 있다. |
| mealViewPager | ViewPager | 입력한 식단을 보여준다. |
| TextView\_calorie | textView | 현재까지 섭취한 칼로리 총량 |
| pieChart | PieChart | 각 영양성분의 비율을 표시한다. |
| foodManagerViewModel | ApiViewModel | 식단화면내의 기능을 관리하는 클래스스 객체 |
| apiViewModel | ApiViewModel | API기능을 관리하는 클래스 객체 |
| Button\_NavigateToApi | Button | API화면으로 이동하는 버튼 |
| Spinner\_sectionFood | Spinner | 아침, 점심, 저녁 선택 기능 |

3.5 물정보 관리 기능(상인)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스명 | | WaterViewModel | | |
| 클래스 개요 | | 물정보 뷰 클릭 시 보여지는 물정보 화면 클래스 | | |
| Stereotype | | Control | | |
| Public  Interface  Public | Operations | Operation(Parameter) | Return | 설명 |
| plusDrink() | void | 물 섭취량 100ml증가 |
| MinusDring() | void | 물 섭취량 100ml감소 |
| StoreDrink() | void | 물정보 저장 |
| showDrink(date: Date) | WaterInfo | 물 정보를 가져온다 |
| calculateDrink(date: Date) | int | 물 정보를 계산한다 |
| Attributes | Attribute | Type | 설명 |
| db | PlannerDatabase | 데이터베이스 변수 |
| dao | PlannerDao | 다오 변수 |

3.6 물정보 화면

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스명 | | WaterFragment | | |
| 클래스 개요 | | 물 정보 뷰의 기능 클래스 | | |
| Stereotype | | Boundary | | |
| Public  Interface  Public | Operations | Operation(Parameter) | Return | 설명 |
| button\_plusDrink.setOnClickListener() | void | 100ml 증가 버튼 리스너 |
| button\_minusDrink.setOnClickListener() | void | 100ml 감소 버튼 리스너 |
| button\_NavigateToGoal.setOnClickListener() | void | 물 목표량을 설정하는 버튼 리스너 |
| button\_store.setOnClickListener() | void | 물 정보 저장 버튼 리스너 |
| Attributes | Attribute | Type | 설명 |
| button\_plusDrink | Button | 물 100ml 증가 버튼 |
| button\_minusDrink | Button | 물 100ml 감소 버튼 |
| progressbar\_Today | Progressbar | 물정보 프로그레스바 |
| button\_NavigateToGoal | Button | 물 목표량 설정 버튼 |
| button\_store: | Button | 물 정보 저장 버튼 |

3.7 사용자 정보 기능(수정,삭제,조회,계산)

현섭

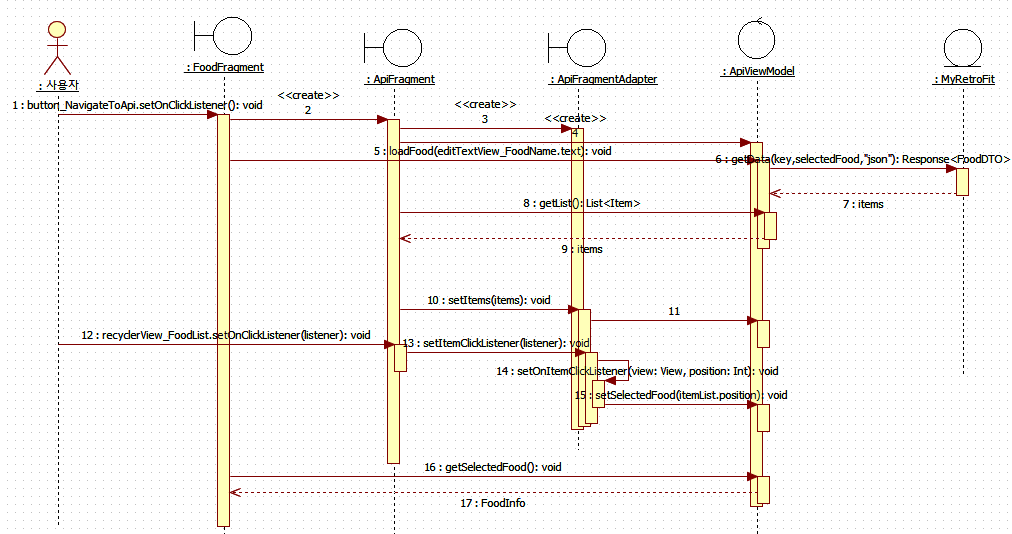
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스명 | | UserViewModel | | |
| 클래스 개요 | | 사용자 정보 수정 기능 클래스 | | |
| Stereotype | | Control Class | | |
| Public  Interface  Public | Operations | Operation(Parameter) | Return | 설명 |
| showUserInfo(date: Date | List<UserInfo> | 사용자 정보를 가져와서 화면에 출력한다. |
| editUserInfo(age: Int, height: Int, weight: Int) | void | 사용자 정보를 수정한다 |
| deleteUserInfo(key: Int) | void | 사용자 정보 기록을 삭제한다. |
| calculateBMI(age: Int, height: Int, weight: Int) | Int | BMI를 계산한다. |
| Attributes | Attribute | Type | 설명 |
| db | PlannerDatabase | 데이터베이스 변수 |
| dao | PlannerDao | 다오 변수 |

3.8 사용자 정보 수정 화면

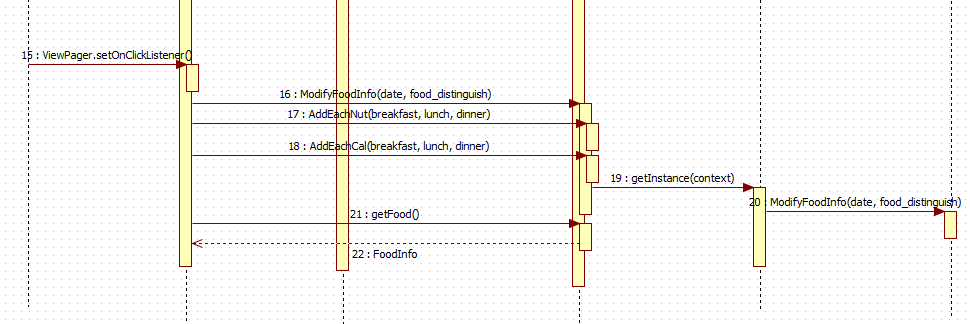
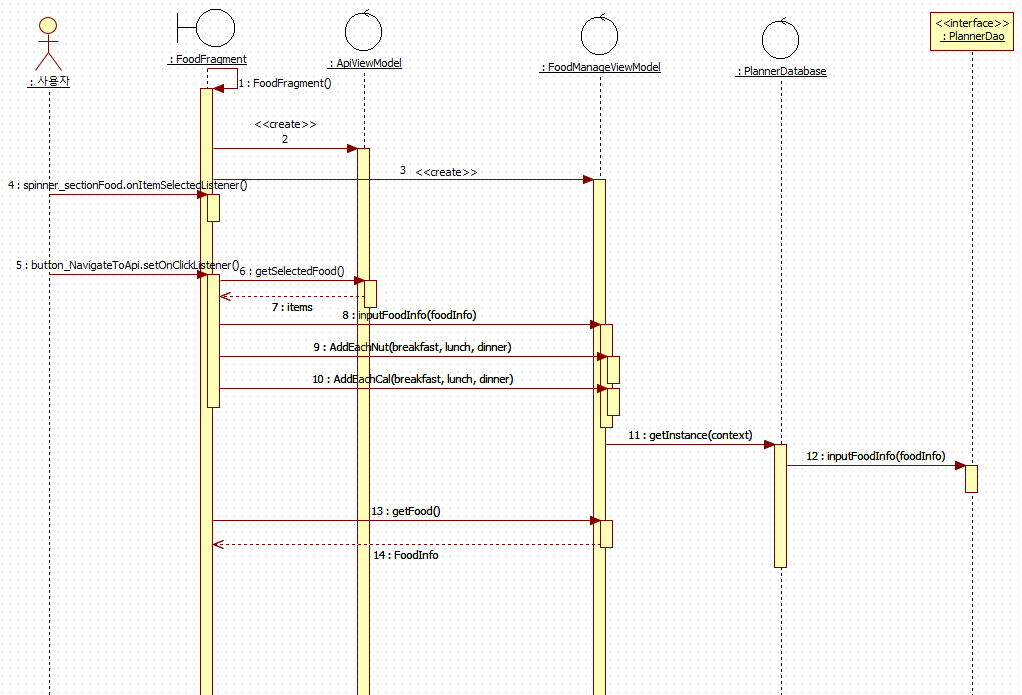
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스명 | | Input user information view | | |
| 클래스 개요 | | 사용자 정보 수정 뷰 클래스 | | |
| Stereotype | | Control Class | | |
| Public  Interface  Public | Operations | Operation(Parameter) | Return | 설명 |
| recyclerView\_Record.setOnClickListener() | void | 사용자 정보 기록을 선택한다다. |
| button\_Update.setOnClickListener() | void | 사용자 정보를 수정하는 버튼 클릭 리스너 |
| button.Delete.setOnClickListener() | void | 사용자 정보를 삭제하는 버튼 클릭 리스너 |
| Attributes | Attribute | Type | 설명 |
| recyclerView\_PresentState | RecyclerView | 현재 상태를 나타내는 뷰 |
| editTextView\_age | EditTextView | 몸무게 적는 칸 |
| editTextView\_height | EditTextView | 키 적는 칸 |
| editTextView\_weight | EditTextView | 몸무게 적는 칸 |
| recyclerView\_Record | RecyclerView | 사용자 정보 기록을 출력하는 뷰 |
| button\_Update | Button | 사용자 정보를 수정하는 버튼 |
| button\_Delete | Button | 사용자 정보 기록을 삭제하는 버튼 |
| userAdapter | UserAdapter | Adapter를 대신하는 속성 |
| userViewModel | UserViewMode | ViewModel을 대신하는 속성 |

4. 시퀀스 다이어그램

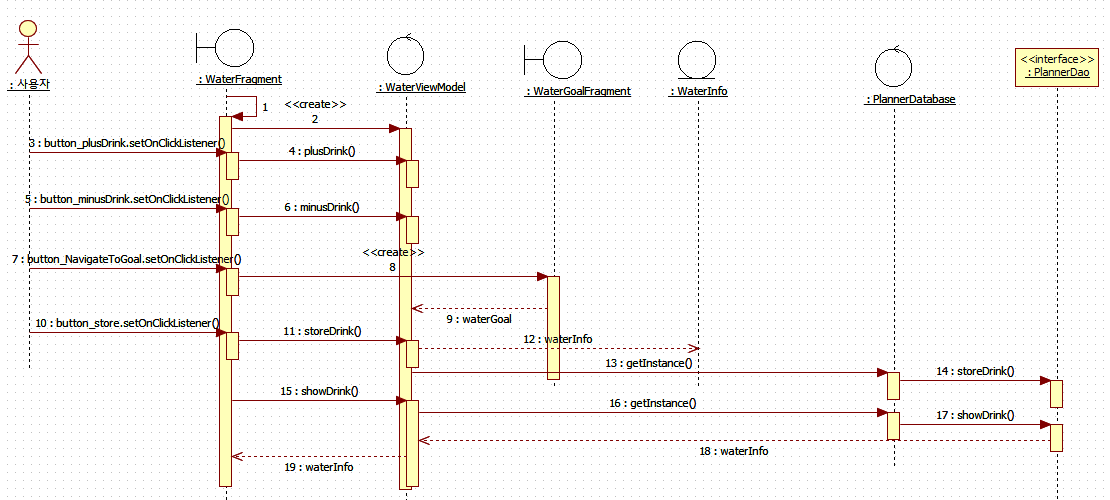
4.1 영양성분조회 시퀀스 다이어그램



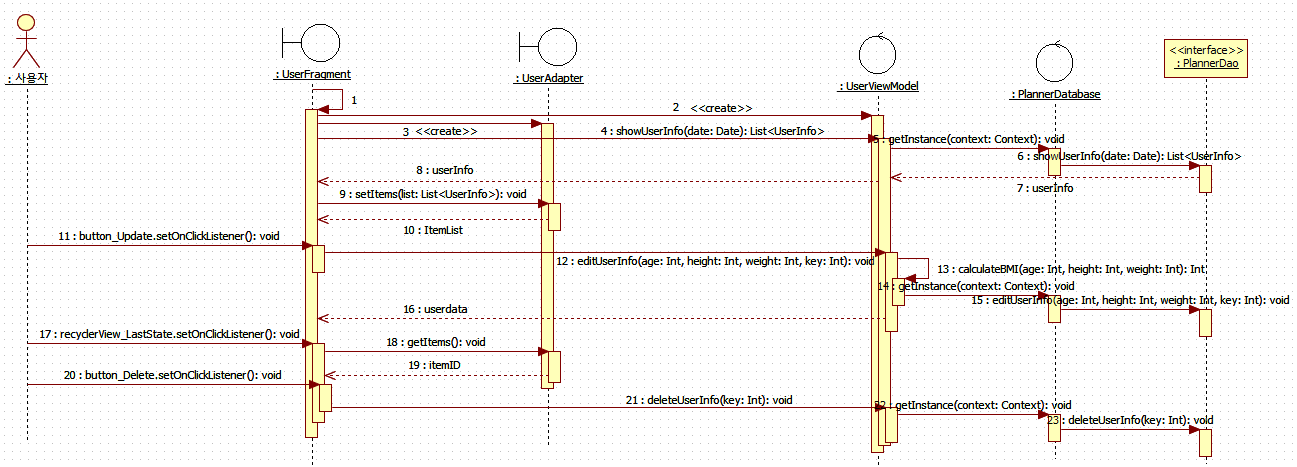
4.1 식단기능(입력,수정,조회,영양성분계산,칼로리계산) 시퀀스 다이어그램



4.3 물정보기능(추가,감소,저장,조회,계산) 시퀀스 다이어그램



4.4 사용자정보기능(수정,삭제,조회,계산) 시퀀스 다이어그램



5. DB설계

UserInfo

-----------------------

user\_date

user\_age

user\_height

user\_weight

user\_id

FoodInfo

-----------------------

food\_name

food\_serving

food\_kcal

food\_carbo

food\_protein

food\_fat

food\_sweet

food\_sodium

food\_distingiush

food\_id

food\_date

food\_userid

WaterInfo

-----------------------

water\_NowDrink

water\_GoalDrink

water\_date

water\_id

water\_userid