

[웹및모바일최신기술] 프로젝트 최종보고서

팀 명	Neo풀				
작 품 명	가이아(GAIA : GArdening In App)				
개발기간	2017년 5월 1일 ~ 2017년 10월 31일				
지도교수	컴퓨터공학과			오 병우	
구 분	학 년	학 번	성 명	휴대전화	E-mail
책임자(팀장)	4	20141280	진연주	6650-7951	aurora@naver.com
팀원1	4	20141238	조수정	9966-8334	qqq3444@gmail.com
팀원2	2	20140342	김태우	4904-7895	ealiza@naver.com
팀원3	2	20141084	임준혁	2678-0724	dbghkd0724@naver.com
<p style="text-align: center;">본인은 웹및모바일최신기술 프로젝트 최종보고서를 첨부와 같이 제출합니다.</p> <p style="text-align: right; margin-right: 100px;">2017년 6월 13일</p> <p style="text-align: right; margin-right: 50px;">팀 장 진 연 주 (서명)</p> <p style="text-align: center; margin-top: 50px;">컴퓨터공학심화프로그램</p>					

웹 및 모바일 최신기술 프로젝트 최종보고서

Gaia(Gardening In App)

1. 서 론

<개발 목표>

□ 아래 그림 1-1과 같이 반려식물 시장을 목표로 하여, 기상청의 날씨 데이터를 가지고 현실 환경을 기반으로 한 모바일에서의 식물 재배와 일정관리 & 기상알림 등의 기능을 제공하는 융합 Application을 제작하고자 하는 바입니다.

□2017년도 기상기후산업 청년창업 지원 사업에 지원하여 Application 제작을 통해 현실 기반 식물 키우기라는 콘텐츠로 특허를 얻는 것을 시작으로 하여, 직업 Application을 Google play 등의 시장에 배포하고, 최종적으로는 국내뿐만 아니라 전 세계적으로 여러 사람이 할 수 있는 게임 및 일정 관리 Application 개발을 목표로 하고 있습니다.

□해당 프로젝트의 목표는 기상청의 날씨 정보 DB 연동과 프로젝트에서 가장 중요한 부분이면서 가장 어려운 부분이라 할 수 있는 식물 별 관리 기능 구현을 제외한 Application 내에서 동작하는 모든 기능(이후 내부기능) 구현이 현재 목표입니다.



[그림 1-1. 프로젝트 목표]

<개발 동기>

□스마트폰 사용자가 기하급수적으로 늘어나는 요즘, 그에 따라 모바일 게임 시장 또한, 빠른 성장률을 보이는 것을 알 수 있습니다. 모바일 게임 시장이 확대됨에 따라 게임 시장에는 여러 다양한 게임들이 하루에도 수십 개씩 쏟아져 나오고 있으며, 특히 여러 게임들 중에서 ○○○키우기 게임 등이 인기에 힘입어 다양한 콘텐츠로 제작되고 있습니다.

□ 게임 시장 내에서 RPG 또는 MMORPG와 같은 인기 있고 유명한 게임분야뿐만 아니라 이러한 키우기 게임도 높은 순위를 유지하고 있는 있으며, 반려식물이라 하여, 마치 반려 동물과 같이 집에서 식물을 키우는 사람들을 위한 반려식물 시장이 반려동물 시장과 비례하여 점차 증가하는 추세[그림 1-2 참고]에 맞

추어 이 두 콘텐츠를 융합한 현실기반 식물 키우기 게임에 초점을 맞추어 프로젝트 개발을 추진하게 되었습니다.



[그림 1-2. 반려동물 시장과 비례하여 점차 증가하고 있는 반려식물 시장에 대한 기사]

<관련 기술 현황>

□ Appliepis Studio Inc 의 ‘헬리펫’이라는 Overlay Screen 기반의 애완동물 키우기 App과 여러 기업들과 연계 되어 사용자들이 재배한 식물 또는 여러 식료품을 제공하는 레알팜이라는 App 그리고 식물 키우기 게임인 plant nanny가 이미 존재하고 있으므로, 제도적 & 기술적으로 application을 구현하고 판매하여 수익을 창출하는 것은 가능하다고 확실할 수 있습니다.

□ 이런 여러 관련 App이 이미 존재하고 있기는 하지만, 현실 기반이라는 콘텐츠를 추가함으로써 사용자들이 좀 더 재미있고, 게임을 하고 있다는 것이 아닌 실제 식물을 키우고 있다는 느낌을 들게 하여 게임에서 더 나아간 융합 콘텐츠를 제공하는 것을 목표로 하고 있습니다.

2. 작품 개요

<설계구성요소 및 설계제한 요소>

설계 구성요소						설계 제한요소						
목표 설정	합성	분석	구현/ 제작	시험/ 평가	결과 도출	성능	규격/ 표준	경제성	미학	신뢰성	안정성/ 내구성	환경
○		○	○	○	○	○		○	○	○		○

-설계 구성요소 평가

1) 목표설정

i. 프로젝트의 목표는 android application에서는 UI 설계, Application의 수익 방식 및 내부 기능 설계 (내부 관리 기능 또는 화면 외부 관리 기능 등)를 완성하고, 설계 완료한 내부 기능들 중 Server에 있는 식물 DB와의 연동이 필요하지 않는 선에서 Application의 내부 기능 구현 완료를 목표로 하고 있고, Server part에서는 간단한 server 구현 및 DB를 구축하고 sever를 통해서 연동하는 기능까지 완료하는 것이 목표로 하고 있다.

ii. 프로젝트에서 가장 어려운 부분일 수 있는 식물에 따른 정보와 키우는 방식을 기반으로 구축된 DB와 이 정보들을 바탕으로 실시간 또는 외부 환경에 의해서 상태가 변하는 기능을 제외한, 상대적으로 쉬우면서 프로젝트에서 중요한 기능들을 1차 목표로 하고 있으며, 기존의 여러 android project에서 쌓은 경험들을 바탕으로 기간 내에 목표에 달성할 수 있을 것이라 확신할 수 있습니다.

2) 분석

i. 유사 Application ‘헬로펫’에서 사용되는 위젯으로 추정되는 기능을 목표로 하였으나 위젯으로는 해당 Application과 같이 프로젝트의 초반 설계처럼 구현하는 데에는 어려움이 있는 것을 발견했다. 추가 검색과 유사 기술 및 application들에 대해 조사한 결과를 통하여 관련 기능을 찾아보니 Overlay Screen이라는 기능이 사용된 것을 알게 되었다. 오픈소스를 분석하여 프로토타입을 완성하여, 실행해 보니 프로젝트가 목표로 하는 그런 기능임을 확인한 후, 위젯에서 overlay Screen으로 설계를 변경하여 개발을 시작함.

3) 구현/제작

i. application 개발을 위해서는 android studio 툴을 사용하였음. Server 구축을 위해서는 linux 환경에서 nodejs를 사용하여 구현함. DB 또한 linux 환경에서 mysql을 다운 받아 구축함.

ii. Client와 Server와의 통신을 위해 Android Studio가 지원하는 Volley Library사용

iii. 오픈소스로 제공되는 Overlay Screen 기술을 계획한 Application 개발 목표에 맞게 수정 및 필요한 코드를 추가하여 구현함.

4) 시험/평가

i. 삼성전자 갤럭시 노트5, 삼성전자 갤럭시 A7, 갤럭시 S8 플러스, LG전자 넥서스5 등 화면 크기, os 버전 등이 다양한 모델에서 잘 구동되는 것을 확인함으로써, application의 안전성을 확인하였으며, 여러 화면 크기에 맞게 UI를 RelativeLayout으로 제작하였다.

ii. 위에서 언급한 여러 기기에서 본 application이 정상적으로 실행되는 것을 확인하였고, UI도 각 기기에 적합하게 View들이 배치되는 것을 확인함.

5) 결과도출

i. 현 프로젝트의 목표((1)목표설정 참고)를 기준으로 했을 때, App에서는 UI 설계를 완료한 후, 전문 업체에 의뢰서를 제출한 상태이고, 여러 아이템 및 식물 구입 기능과 구입 한 후, 사용자에게 정보를 보여주고, 또한 사용자의 DB에서 해당 아이템 및 식물 구입 정보를 Update 하는 부분까지 구현 완료함.

ii. 프로젝트에서 가장 핵심부분이라 할 수 있는, App 메인화면에서 식물을 클릭할 시 overlay Screen 방식으로 App 외부에 식물을 배치할 수 있는 기능 구현 완료함. Server에 있는 식물(아직 식물별 정보와 키우는 방식에 따라서 어떻게 상태를 변경하고 update할지 좀 더 설계가 필요한 부분)에 관한 DB를 이용하지 않는 Application의 내부 기능 구현은 70~80% 완성도를 보인다고 말할 수 있음.

iii. Server part에서는 app이 필요로 하는 DB 정보를 사용자에게 보내주는 부분에 대한 구현과 DB를 구축하고 해당 DB를 Nodejs에서 연동하는 기능까지 완료하였음.

-설계 제한요소 평가

1) 성능

i. Application에서 사용자가 아이템을 구입하였을 시, 사용자 DB에 사용자가 구입한 아이템에 대한 정보(구입한 아이템을 가지고 있는 개수 등)에 대해 정상적으로 update 되었으며, UI 상에서도 아이템의 개수를 표시해 주는 부분에서 개수를 +1하여 표시해 주는 것을 확인함.

ii. 사용자가 식물을 구입하였을 시, DB에 있는 각각의 사용자가 가지고 있는 식물에 대한 table인 plant table에 해당 사용자의 해당 식물을 저장하는 것을 최종적으로 확인, 또한 식물을 구입하였을 때, 바로바로 App 메인 화면에 해당 식물을 보여줌으로서, 사용자에게 식물 구매가 정상적으로 진행되었다는 것을 시각적으로 보여줌.

iii. App 메인화면에서 원하는 식물을 길게 클릭하면, overlay Screen에 해당 식물을 띄워주며, App 외부(App이 꺼진 후에도)에서도 식물을 관리할 수 있는 기능 구현을 완료함.

2) 경제성

i. 최종적으로 국내뿐만 아니라, 전 세계적으로 여러 사람이 이용할 수 있는 게임 & 일정 관리 App 개발을 목표로 두고 있습니다. 현재 전 세계적으로 게임 시장의 규모는 점점 증가하고 있으며, 특히 모바일 게임에 대한 대규모의 시장이 이루어진 점에서 다른 Application에 비해 좀 더 쉽고 간편하게 저희 게임을 상품화할 수 있고, 국내에서 뿐만 아니라 국제 시장에서도 높은 매출액을 얻을 수 있을 거라 보고 있습니다.

ii. Application 등록(판매) 초반에는 대부분 광고 업체와의 연계를 통한 광고로 수익을 얻고, 후반에는 점차 사용자들이 증가함에 따라, 식물을 키우는데 필요한 여러 아이템들과 현금을 통해서만 구입할 수 있는 희귀식물들을 따로 만들어 수익을 얻을 것입니다.

iii. 식물을 최종 단계까지 키우면, 응모하기 버튼이 화면에 나타나게 되는데 이 버튼을 누르면 응모를 할 수 있으며, 응모를 하면 계약을 맺은 원에 업체에서 해당 식물을 사용자의 집으로 배송 해줌으로서 Application 사용자에게 목표의식을 주어 사용자가 쉽게 질려하지 않으면서 사용 빈도수(하루 사용량)를 증가시켜 더 높은 수익을 창출하는 것이 목표입니다.

3) 미학

i. 사용자가 인터페이스를 보고 한눈에 기능을 이해하고 사용할 수 있도록 UI를 구상하였고, 시장 수요가 있는 UI와 일러스트를 제작하기 위해 “바나나몬 - 게임 그래픽 외주 회사(www.bananamon.com)”에 제작 의뢰서를 제출함.

ii. 팀원 각자가 원하는 식물과 화분을 선택하여, 각 식물의 성장 단계에 따른 식물의 이미지와 정보 등을 조사하여, 제작 의뢰서를 좀 더 전문적이면서, 저희가 원하는 식물의 모습으로 제작될 수 있도록 노력함.

4) 신뢰성

i. 목표로 했던 기능들은 삼성전자 갤럭시 노트5, 삼성전자 갤럭시 A7, 갤럭시 S8 플러스, LG전자 넥서스5 등 다양한 os 버전과 화면 크기를 갖는 여러 기기에서 완벽하게 실행되는 것을 확인하였음.

ii. 사용자가 로그인 한 후, 메인 화면에서 식물을 overlay Screen에 띄우기 위해 클릭하였을 시, App이 다시 로그인 화면으로 돌아가는 문제점이 발생함. 그 점을 제외하고는 목표로 했던 기능들은 성능을 검사해봤던 여러 기기에서 정상적으로 작동하는 것을 확인함.

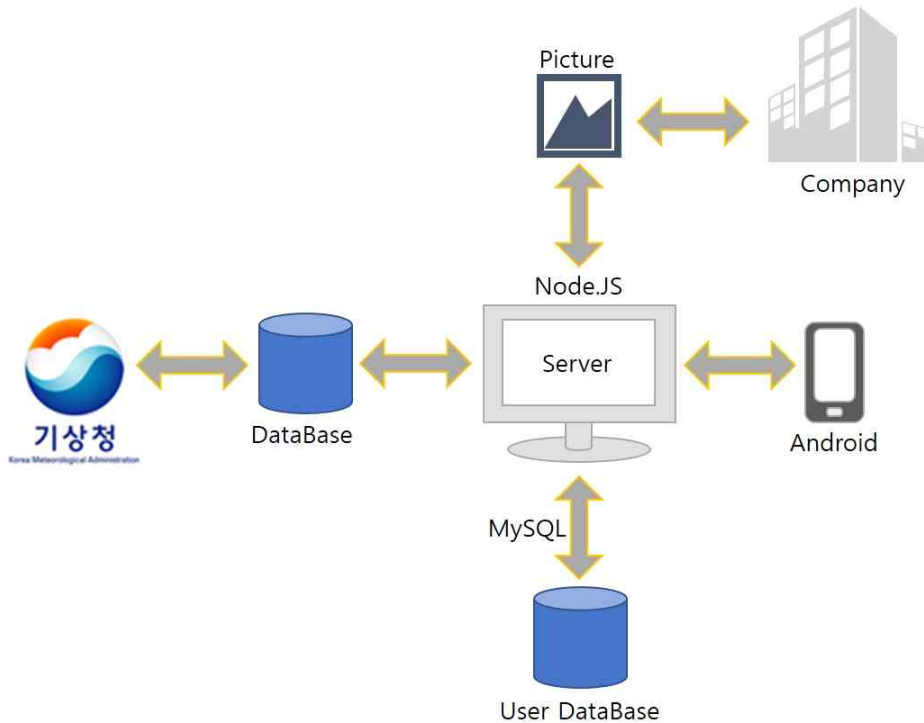
5) 환경

i. Client part : Android 6.0 (Marshmallow) 23API 이상 버전에서 실행이 가능함.

ii. Server part

- Node js로 구현되었으며, linux 환경에서 구축.
 - Client와의 주기적인 데이터 교환이 필요하므로 Server는 항상 켜져 있어야 함.
-

<시스템 구성도>



[그림 1-1. Application 전체 구성도]

□ 위에 그림과 같이, 저희 App은 크게 사용자 관리, 식물 DB, 사용자가 소유한 식물들의 관리 기능을 담당하는 자체 Server와 현실 기반의 실시간 기상 데이터를 제공하는 기상청 날씨 정보 DB, 직접적으로 사용자에게 제공되는 android 기반의 Application이 있습니다.

□ 게임 특성상 가장 중요한 UI는 자체적인 기술이 부족한 관계하고, 사용자에게 실질적으로 판매하는 app을 목표로 하고 있기 때문에, 자체 제작 보다는 전문 업체에 의뢰하는 것이 상업적인 면에서 더 높은 수익을 창출할 수 있을 것이라 판단하여, 직접 제작이 아닌, UI 전문 업체에게 제작 의뢰를 맡기는 형식으로 진행될 것입니다.

□ 사용자에게 직접적으로 제공되는 Application은 android 기반으로 제작된 App으로, 로그인/회원가입, 식물 추가, 식물 내부관리, 식물 외부관리 그리고 식물 구입 등의 기능이 제공됩니다. Server와의 통신에서는 Volley Library를 사용해 데이터를 주고받을 계획입니다. (Server로 송수신되는 data 형식에 대해서는 후에 Sequence Diagram 참고) 또한, 저희 App의 가장 핵심 기능인 App 외부에서의 식물 관리에서는 Overlay Screen 기술을 사용하여 초반에 계획했던 위젯 기반보다 더 쉬우면서 자유롭게 식물을 App 외부에서 관리할 수 있도록 사용자에게 지원할 계획입니다.

□ Server는 linux 환경에서 Node.js 기반으로 구축되었으며, DB는 MySQL로 User Table(사용자 관리), Plant Table(식물 Data 관리), Pot Table(화분 Data 관리)로 설계하였습니다. DB는 사용자가 직접적으로 접근하여 정보를 update하거나 추가/삭제하는 것이 아닌, Server로 DB update/추가/삭제 등을 요청하면, Server가 연동되어있는 DB를 관리하는 방식으로 진행하여, 더욱 안정적으로 관리하는 것을 목표로 하고 있습니다.

3. 개발 내용

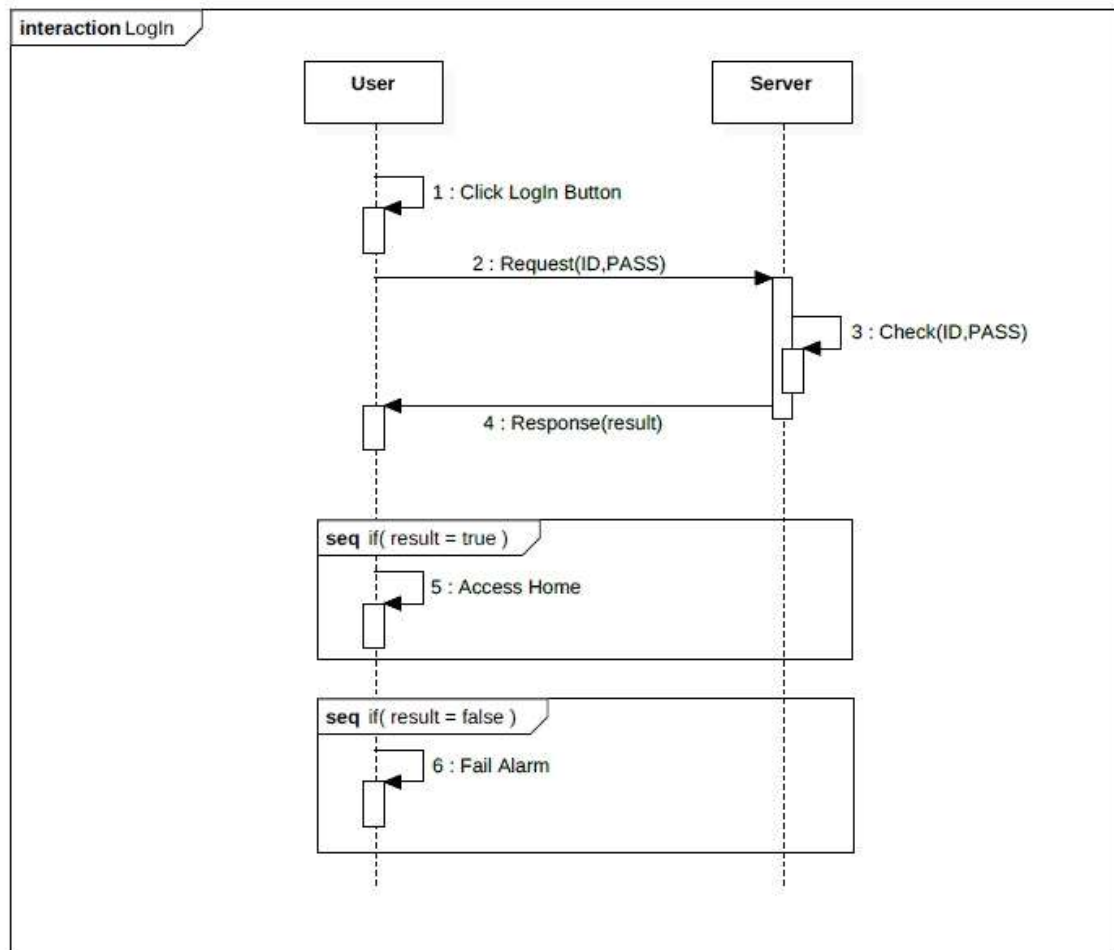
<상세 설계 내역>

-Sequence Diagram

1. 로그인/회원가입

□로그인

Collaboration1::Interaction1::Login



(1) 사용자는 자신의 ID와 Password를 입력하고 로그인 버튼을 누른다.

(2) ID, Pass를 서버에 전송한다.

->{ id : id, pass : password }

(3) Server는 사용자로부터 전송된 id와 pass가 존재하는지 또는 id와 password가 일치하는지 검사한다

-> User Table을 사용하여 검사

(4) Server는 검사 결과에 따라 result값을 Client에게 전송한다.

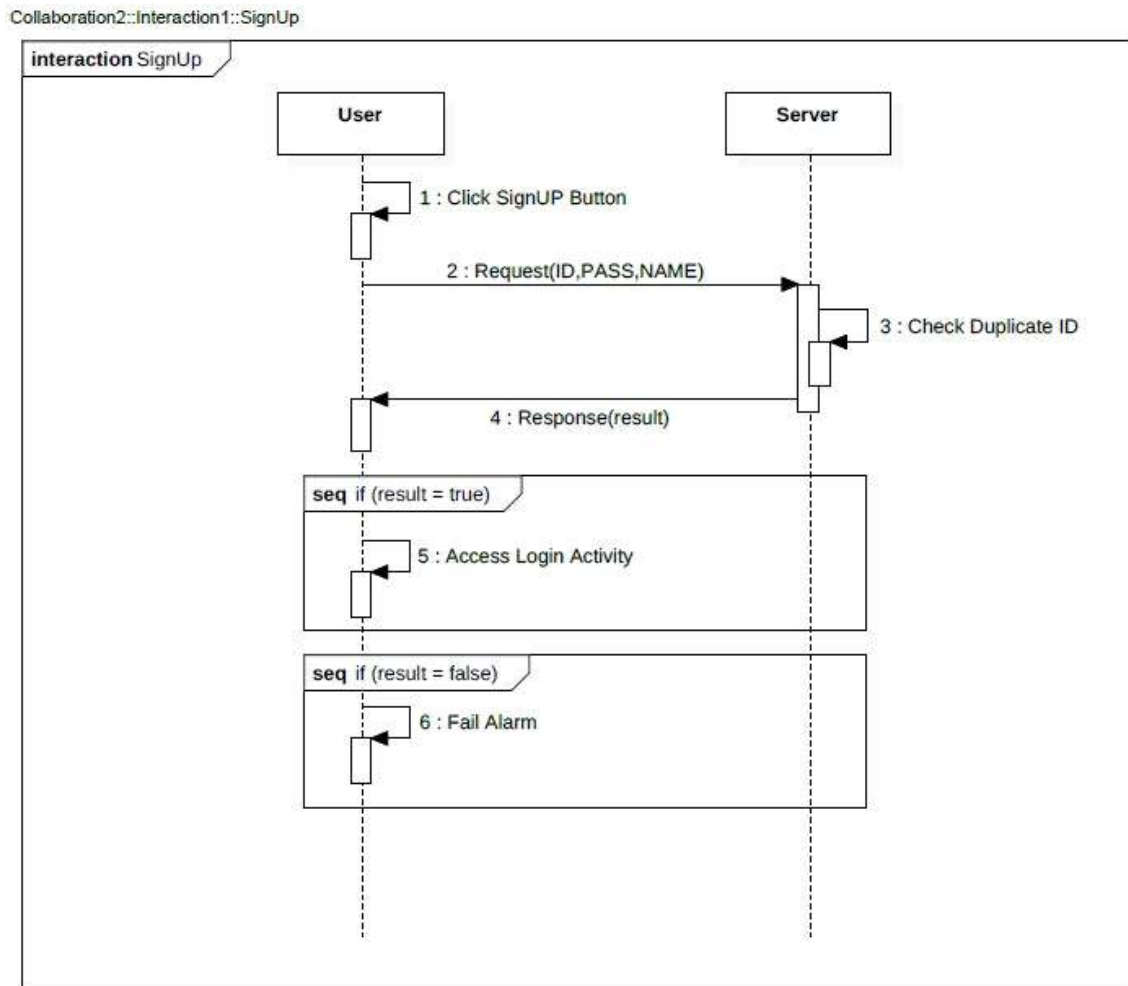
-> result : String ["true" or "false"]

-> client 부분에서도 String으로 받음.(StringRequest)

(5) result값이 true이면 Application의 홈 화면으로 이동한다.

(6) result값이 false이면 로그인에 실패했다는 알림을 화면에 띄어 준다.

□회원가입



(1) 사용자는 자신이 원하는 id, password, nickname을 작성하여 회원가입 버튼을 누른다.

(2) Server로 id, password, nickname을 서버에 전송한다.

->{id : id, pass : password, nickname : nickname}

(3) Server는 전송된 id, name이 이미 존재하는지 중복검사를 한다.

(4) Server는 검사 결과에 따라 result값을 Client에게 전송한다.

-> result : String ["true" or "false"]

-> client 부분에서도 String으로 받음(StringRequest)

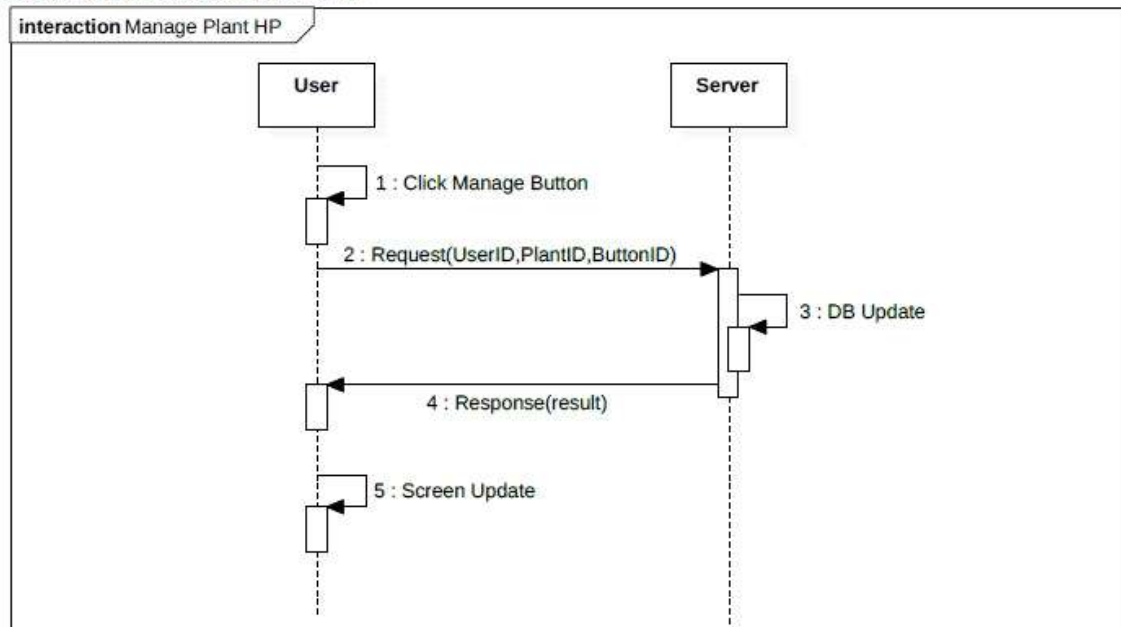
(5) result 값이 true면 회원가입 성공 -> 로그인 화면으로 이동한다.

(6) result 값이 false면 회원가입에 실패했다는 알림을 화면에 띄어 준다.

2. 식물 관리하기

□식물 HP 관리

Collaboration5::Interaction1::Manage Plant HP



(1) 사용자는 식물 내부 관리 창에서 적용하고 싶은 관리 버튼을 클릭한다

(2) user와 관리를 받을 식물, 관리 버튼의 ID를 서버로 전송한다.

- { userID : userID, manageID : mID , plantID : plant }

(3) Server는 User DB에서 해당 user가 누른 관리 버튼의 효과에 따라 사용자가 관리하기 원하는 식물 (plantID)의 이 HP를 증가시킨다.

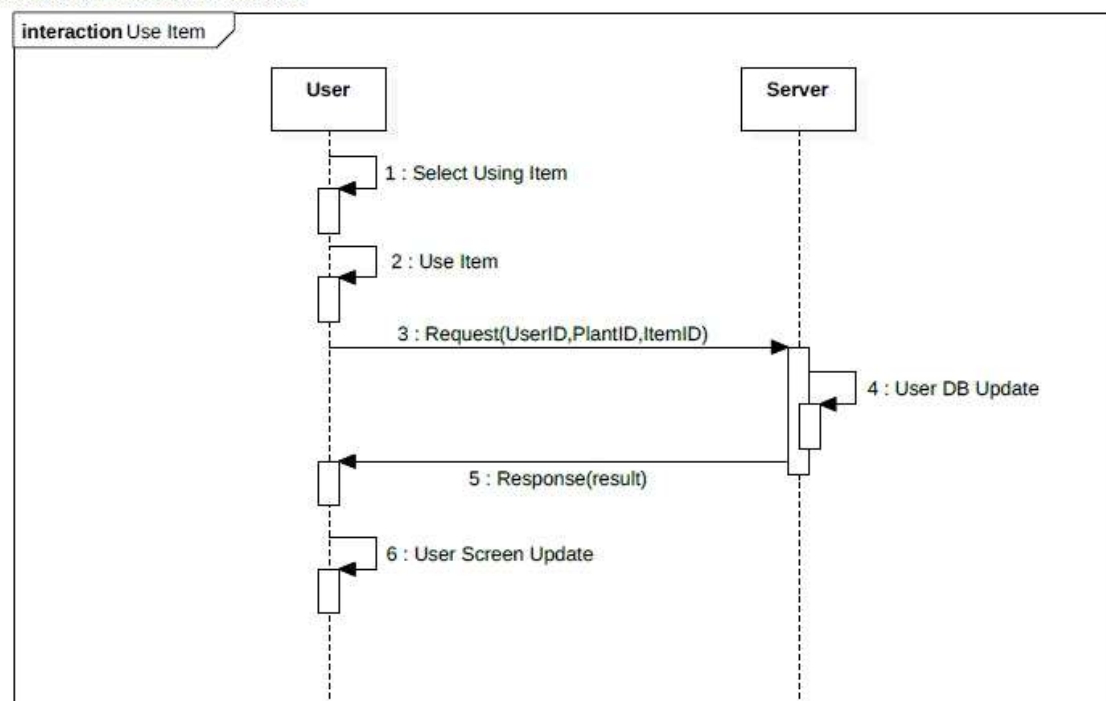
- 관리기능과 관련된 DB 사용

(4) Server에서 관리 버튼의 따른 식물의 HP 정보 Update 작업이 완료 되면 result 값을 Client에게 전송한다.

(5) result 값이 true면 Application 내부관리 기능에서 HP Bar의 상태를 갱신한다.

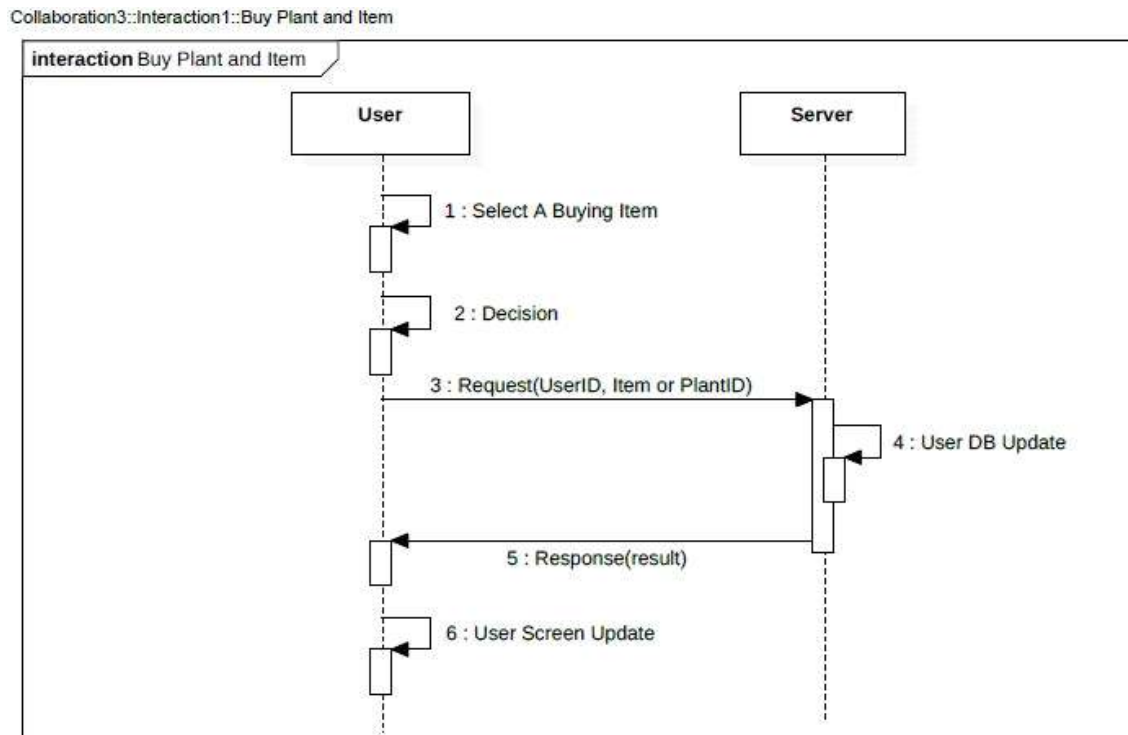
□아이템을 사용하여 식물 EXP 증가

Collaboration4::Interaction1::Use Item



- (1) 사용자는 App의 식물 내부 관리 창에서 사용할 아이템을 선택한다.
- (2) App의 화면에 아이템을 사용하는 효과를 보여준다.
- (3) 사용된 아이템과 아이템이 사용된 식물, user의 ID를 Server로 전송한다
 - {userID : userID , itemID : item, plantID : plant }
- (4) Server는 User DB에서 해당 user가 사용한 아이템만큼 해당하는 아이템의 개수를 차감한다.
- (5) Server는 해당 아이템이 사용된 식물의 EXP를 아이템의 효과에 따라 증가시킨다.
 - item DB에 존재하는 정보를 이용
- (6) Server에서 아이템에 따른 식물 정보 Update 작업이 완료 되면 Client로 result 값을 전송한다.
 - result : String [“true” or “false”]
- (7) result 값이 true이면 Application 내부 관리 화면에서 해당 식물의 EXP Bar를 갱신해준다.

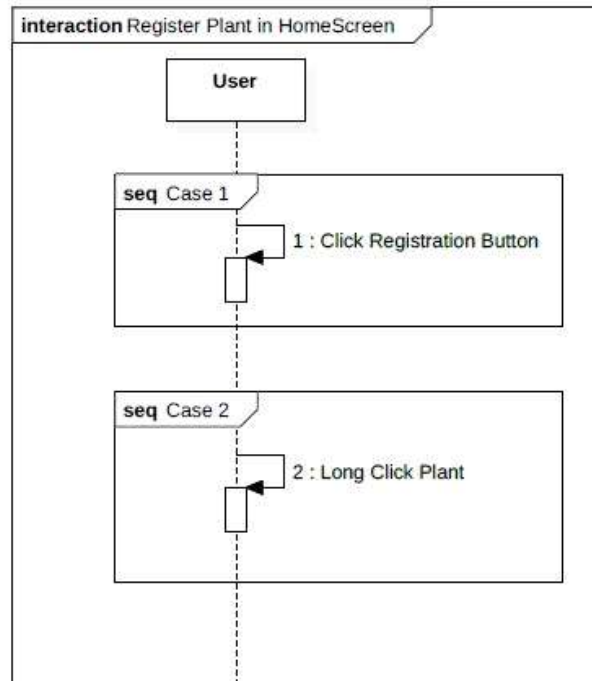
3. 아이템/식물 구매



- (1) 사용자는 구매할 아이템 혹은 식물을 선택한다.
- (2) 사용자는 선택이 완료되었다면 구매 결정하고 구매 버튼을 클릭한다.
- (3) 구매를 하는 userID와 구매하는 식물 혹은 아이템의 ID를 Server로 전송한다
 - > {userID : userID, item/plantID : item/plant}.
- (4) Server는 UserDB에
 - 구입한 물품이 식물이면 User가 가진 식물 정보에 해당 식물을 추가한다.
 - 구입한 물품이 아이템이면 user가 가진 아이템 정보에 해당 아이템을 추가한다.
- (5) Server는 DB에 정보를 Update하는 작업이 완료되면 Client에게 결과값(result)를 전송한다.
 - result : String [“true” or “false”]
- (6) result값이 true면 Application의 인벤토리 화면을 갱신한다.

4. 스마트폰 홈 화면에 식물 등록

Collaboration6::Interaction1::Register Plant in HomeScreen



1. 사용자는 식물 내부 관리 창에 있는 ON / OFF 버튼을 클릭한다.

- OFF 상태로 변경할 경우 : 식물을 App 외부에서 내부로 가지고 온다.
- ON 상태로 변경할 경우 : 식물을 App 내부에서 외부로 가지고 간다.
- > Overlay Screen에 해당하는 view를 추가한다.

2. 사용자는 App의 메인 화면(식물이 나열되어 있는 화면)에서 외부로 가지고 가길 원하는 식물을 길게 클릭한다.

- 해당 식물을 외부 관리 창에서 원하는 위치에 이동시킨다.

-자료구조 설계

□User table

Field	Type	Null	Key	Default	Extra
userNo	int(11)	NO	PRI	NULL	auto_increment
userEmail	varchar(20)	NO		NULL	
userPassword	varchar(20)	NO		NULL	
nickname	varchar(20)	NO		NULL	
seed	int(11)	NO		0	
fruit	int(11)	NO		0	
water	int(11)	NO		0	
fertilizer	int(11)	NO		0	
pesticideNum	int(11)	NO		0	

- userNo int NOT NULL AUTO_INCREMENT PRIMARY_KEY auto_increment
 - 유저에게 부여 된 고유 번호(유저 관리 등에 사용)
- userEmail varchar(20) NOT NULL
 - 사용자가 지정한 비밀번호
- userPassword varchar(50) NOT NULL
 - 사용자가 지정한 비밀번호(후에 암호화 수행)

- nickname varchar(20) NOT NULL
 - 사용자가 지정한 닉네임
- seed int NOT NULL DEFAULT 0
 - 사용자가 가지고 있는 게임 내부의 재화
- fruit int NOT NULL DEFAULT 0
 - 사용자가 가지고 있는 현금 재화
- water int NOT NULL DEFAULT 0
 - 사용자가 가지고 있는 물(아이템)의 개수
- fertilizer int NOT NULL DEFAULT 0
 - 사용자가 가지고 있는 비료(아이템)의 갯수
- persticideNum int NOT NULL DEFAULT 0
 - 사용자가 가지고 있는 물약(아이템)의 개수

□ Flower table

Field	Type	Null	Key	Default	Extra
flowerNo	int(11)	NO	PRI	NULL	auto_increment
flowerName	int(11)	NO		NULL	
flowerImagePath	varchar(30)	NO		NULL	
flowerExplain	varchar(200)	NO		NULL	
seedPrice	int(11)	NO		NULL	
fruitPrice	int(11)	NO		NULL	
maxHP	int(11)	NO		100	
maxEXP	int(11)	NO		100	
maxLevel	int(11)	NO		NULL	
water	time	NO		NULL	
specialInfo	varchar(20)	YES		NULL	
flowerEffect	int(11)	YES		NULL	

- flowerNo int NOT NULL AUTO_INCREMENT PRIMARY_KEY auto_increment
 - 꽃이 가지는 unique 번호(primary key)
- flowerName varchar(20) NOT NULL
 - 꽃의 이름
- flowerImagePath varchar(50) NOT NULL
 - 꽃 이미지가 저장된 경로
- flowerExplain varchar(20) NOT NULL
 - 꽃에 대한 설명
- seedPrice int NOT NULL
 - 꽃의 게임 내 재화 가격
- fruitPrice int NOT NULL
 - 꽃의 현금 재화 가격
- maxHP int NOT NULL DEFAULT 100
 - 꽃의 최대 체력
- maxEXP int NOT NULL DEFAULT 100
 - 꽃의 최대 경험치
- water time NOT NULL
 - 꽃에 물을 줘야하는 시간 주기
- specialInfo varchar(20)
 - Effect와 같이 쓰일 꽃의 특수 정보
- flowerEffect int
 - 해바라기 등 해 방향에 따라 꽃이 따라가는 등의 특수한 행동

□Pot table

Field	Type	Null	Key	Default	Extra
PotNo	int(11)	NO	PRI	NULL	auto_increment
PotImagePath	varchar(30)	NO		NULL	
PotName	varchar(20)	NO		NULL	
seedPrice	int(11)	NO		NULL	
fruitPrice	int(11)	NO		NULL	
HPEffect	int(11)	NO		NULL	
EXPEffect	int(11)	NO		NULL	

- PotNo int NOT NULL AUTO_INCREMENT PRIMARY_KEY auto_increment
 - 화분이 가지는 unique ID(primary key)
- PotImagePath varchar(30) NOT NULL
 - 화분 이미지가 저장된 경로
- PotName varchar(20) NOT NULL
 - 화분 이름
- seedPrice int NOT NULL
 - 화분의 게임 내 재화 가격
- fruitPrice int NOT NULL
 - 화분의 현금 재화 가격
- HPEffect int NOT NULL
 - 화분에 있는 특수 효과(HP 감소량을 줄여준다 등)
- EXPEffect int NOT NULL
 - 화분에 있는 특수 효과(EXP 얻는 양을 높여준다 등)

<기능별 설명 또는 화면별 설명>



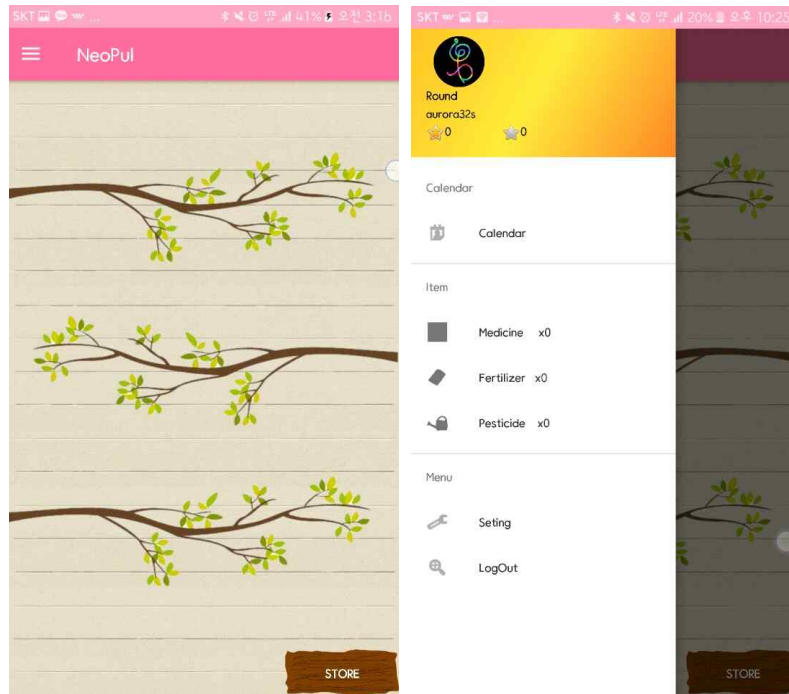
[그림 2-1. 로그인 화면]

회원정보(Email, PassWord)를 통해 로그인을 하는 화면이다.

Edit Box에 각 정보를 넣고 로그인 버튼을 누르면 메인 화면으로 이동한다.

회원가입 버튼을 누르면 회원정보를 등록할 수 있다.

Check Box를 통해 다시 Application을 실행 했을 때, 로그인 정보를 기억하여 자동 로그인 할 수 있게 한다.

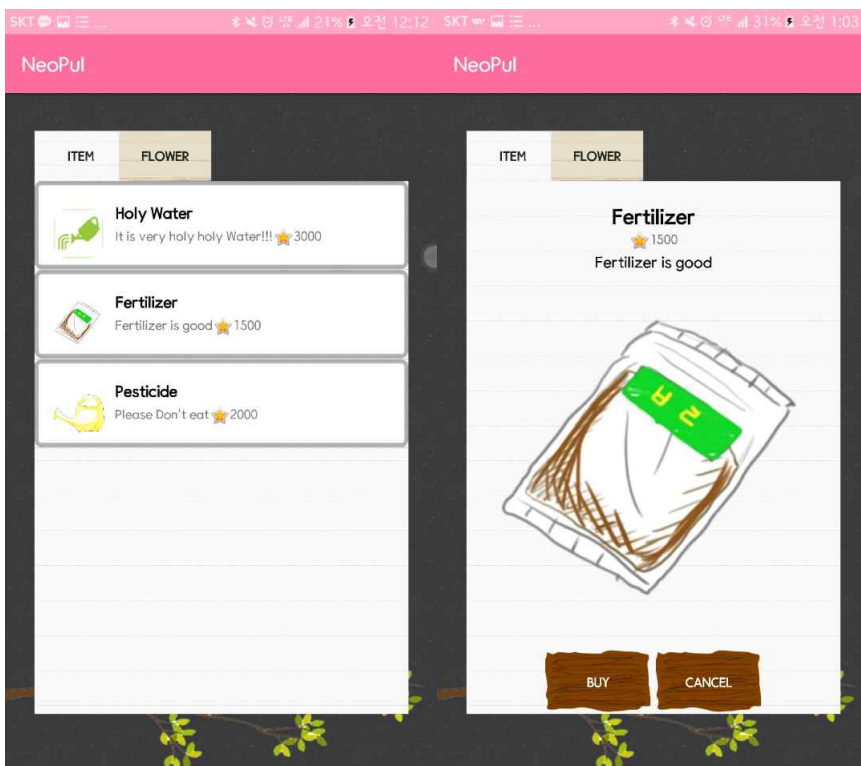


[그림 2-2(1). 메인 화면] [그림 2-2(2). Navigation Drawer 화면]

키우는 식물을 한 눈에 볼 수 있는 화면이다.

Navigation Drawer 기능으로 로그인 한 회원의 정보와 게임 내 재화 정보, 가지고 있는 아이템 정보를 확인할 수 있으며 Application 내부 설정과 로그아웃, 캘린더로 이동하는 기능을 가진다.

또한, 키우려는 식물을 클릭 했을 때, 해당 식물의 관리 화면으로 이동하며, 길게 클릭 했을 때, Overlay Screen 기능으로 해당 식물을 스마트폰 홈 화면에 놓을 수 있으며, 우측 하단에 있는 Store 버튼을 통해 상점 화면으로 이동 할 수 있다.

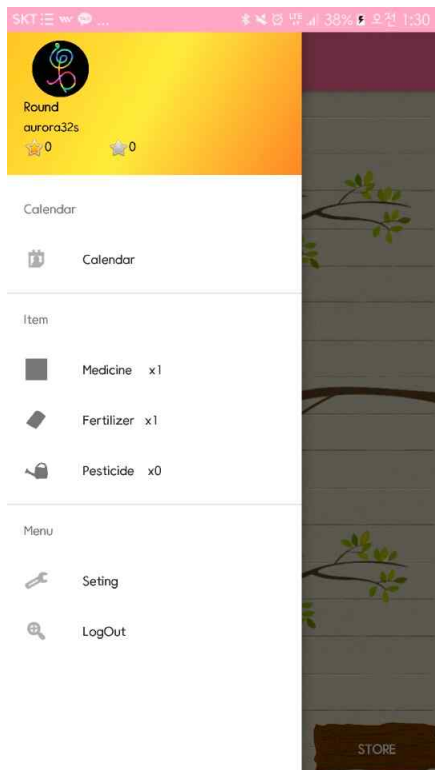


[그림 2-3(1). 상점(아이템) 화면(좌측)]

[그림 2-3(2). 아이템 상세 구매 화면(우측)]

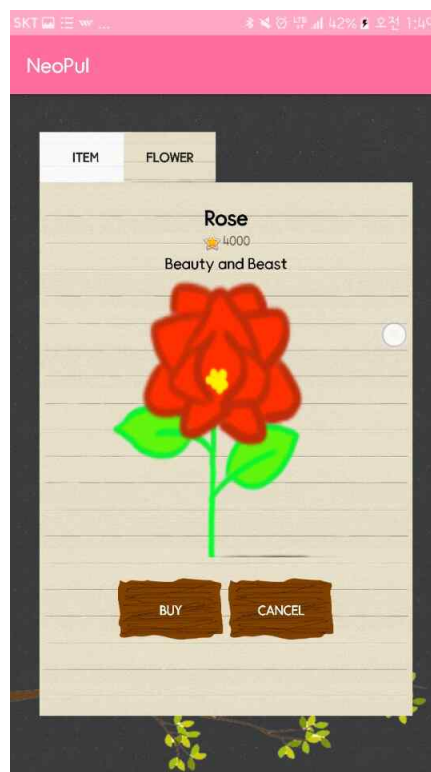
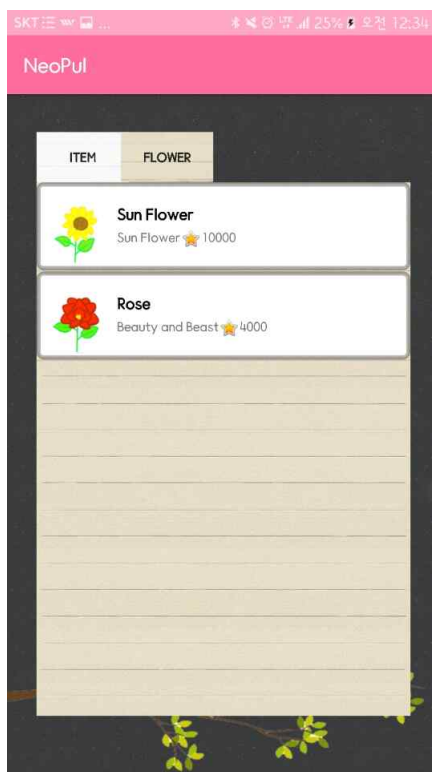
상점 화면에 들어오면 보이는 Fragment 구조의 아이템/꽃 상점에서 아이템에 해당한다(그림 2-3(1) 참고).

리스트뷰 형태로 아이템을 보여주고, 구매하고자 하는 아이템을 클릭하면, 해당 아이템의 상세 정보와 정말 살 것인지를 보여준다(그림 2-3(2) 참고)



[그림 2-4. 아이템 구매 후 Navigation Drawer 화면]

좌측 이미지와 같이 가진 아이템 수가 0개지만(그림 2-2(2) 참고), 상점에서 아이템을 구매 시 화면이 갱신되어 1개가 된 것을 볼 수 있다(그림 2-4 참고).

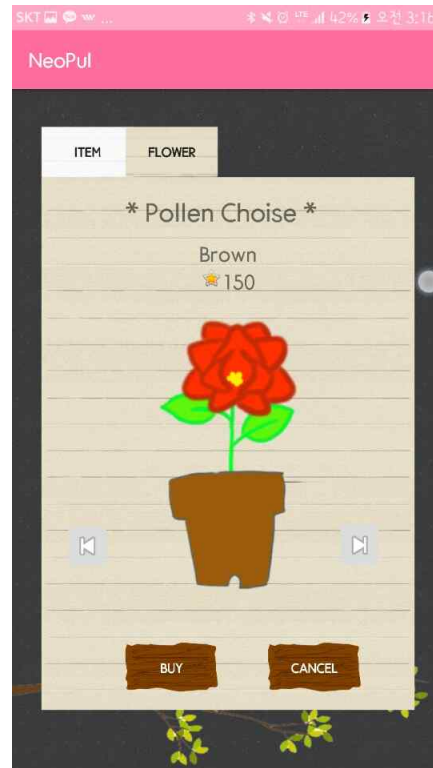
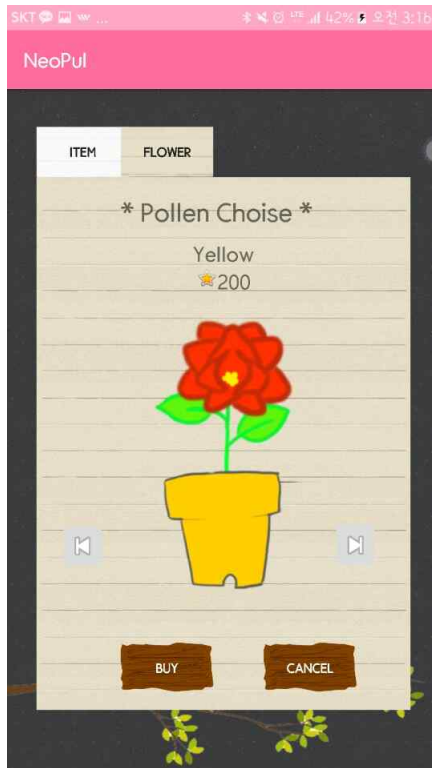


[그림 2-5(1). 상점(식물) 화면(좌측)]

[그림 2-5(2). 식물 상세 구매 화면(우측)]

상점 화면에 들어오면 보이는 Fragment 구조의 아이템/꽃 상점에서 꽃에 해당한다(그림 2-5(1) 참고).

리스트뷰 형태로 꽃을 보여주고, 구매하고자 하는 꽃을 클릭하면, 해당 꽃의 상세 정보와 정말 살 것인지를 보여준다 (그림 2-5(2) 참고)



[그림 2-6(1). 화분 선택 화면1(노랑)]

[그림 2-6(2). 화분 선택 화면2(갈색)]

그림 2-5(2)에서 Buy 버튼을 눌렀을 때, 해당 꽃의 화분을 선택하는 화면으로 넘어오는데, 그림 2-6(1)과 2-6(2)처럼 화분을 미리보기로 보고 Buy 버튼을 누르면 사려는 꽃에 고른 화분이 적용되어 구매된다.

<실험 항목 및 실험 결과>

- 실행화면은 위 그림 2-1 ~ 2-6 참고

□실험 항목

i. 로그인(성공)

- 사용자가 자신의 아이디와 비밀번호를 입력
- 로그인에 성공할 시 메인화면으로 이동, 실패할 시 toast 메시지를 띄어줌,

ii. 회원가입(성공)

- 사용자가 자신이 원하는 아이디, 닉네임, 비밀번호를 입력
- 회원가입에 성공할 시 로그인 화면으로 이동, 실패할 시 toast 메시지를 띄어줌.
- 성공 시, user table에 추가

iii. 아이템 / 식물 구입(성공)

- 구입하기 원하는 아이템 / 식물 선택 후, 구입 버튼 클릭
- 구입 성공 시, user table / plant table에 추가
- 실패 시 toast 메시지를 띄어줌.

iv. 식물을 overlay Screen 에 추가(추가는 성공)

- 원하는 식물을 길게 클릭할 시, 해당 식물을 overlay Screen에 추가
- 로그인 후, 식물을 overlay screen에 추가 시, app이 다시 로그인 화면으로 돌아가는 오류 발생

4. 결 론

□개발 내용 및 실험 결과 요약

i . Client part(App)

- 로그인과 회원가입 기능
- 메인 화면에서 최대 9개(현금을 통해 키울 수 있는 식물의 수를 늘릴 수 있다.)의 식물을 키울 수 있으며, 식물을 키우는 데 필요한 아이템과 키우고 싶은 식물과 화분을 상점에서 구입할 수 있다.
- 식물을 외부로 꺼내고 싶으면, 식물을 길게 클릭해 Overlay Screen 방식으로 App 외부로 꺼낼 수 있다.
- 원하는 식물을 짧게 클릭하면, 식물 내부 관리 기능 화면으로 이동할 수 있다.
- Volley library를 사용하여 Server와 통신

ii . Server part(Server + DB)

- Nodejs를 통해 구현
- MySQL로 DB 구축
- Server에서 DB 관리
- 사용자에게 요청에 따라 DB 정보 전송

□향후 개선 과제

i . 기상청 DB로부터 실시간으로 사용자가 위치한 지역의 날씨 정보를 받아와, 식물에게 영향을 주는 기능을 추가

ii . 로그인 후, 식물을 Overlay Screen에 추가할 시, 다시 로그인 화면으로 이동하는 오류를 추가할 필요가 있음.

iii. 내부관리 기능, 외부 관리 기능을 통해서 식물 상태를 변경하여, 식물의 경험치 및 HP를 조절할 필요가 있음. 이를 위해, 먼저 각 식물의 정보 및 키우는 방식(자라는 환경 등)을 조사하여 식물 DB를 좀 더 세부적으로 update시키고 사용자의 관리 및 외부 환경(날씨 등)에 의한 영향으로 각 식물을 어떻게 컨트롤 할지 더 구체화할 필요가 있음.

iv. 현재 발생하는 여러 문제들을 개선하여 오류 없이 실행 되도록 하고 또한, 임시로 넣어놓은 식물의 종류, 이미지 등을 이미지 외주를 통하여 더 질 높은 이미지로 대체한다.

□기술적, 사회적, 경제적 파급 효과 및 기대 효과

i . 모바일 게임 시장이 확대됨에 따라 게임 시장에는 여러 다양한 게임들이 하루에도 수십 개씩 쏟아져 나오고 있으며, 특히 여러 게임들 중에서 ○○○키우기 게임 등이 많이 등장하고 있으나, 이런 키우기 게임들은 스마트폰 application 내부의 소프트웨어적으로 설정한 환경(배경)에 국한되어 있고, 일부를 제외하고는 대부분의 게임이 게임의 기능에만 충실하기 때문에 사용자들이 쉽게 질려한다는 단점이 존재합니다. 그래서 이 프로젝트를 통해 기상청의 날씨 데이터를 가지고 현실 환경을 기반으로 한 모바일에서의 식물 재배와 일정관리 & 기상알림 등의 기능을 제공하는 융합 Application을 제작하여, 사용자에게 실제 식물을 키우고 있다는 현실성을 제공하면서, 단지 식물 키우기에 그치지 않고 일정관리 기능까지 제공하면서, 거대한 게임 시장에서 큰 경제적 수익을 얻을 수 있을 것이라 예상합니다.

ii. 서울 양재화훼공판장과 대전 둔산동 꽃시장 등, 원예 시장과 계약을 통해, Application 사용자가 키운 식물을 최종 단계까지 키우면, 응모하기 버튼이 화면에 나타나게 되는데 이 버튼을 누르면 응모를 할 수 있으며, 응모를 하면 계약을 맺은 원예 업체에서 해당 식물을 사용자의 집으로 배송 해줌으로서 Application 사용자에게 목표의식을 주어 사용자가 쉽게 질려하지 않으면서 사용 빈도수(하루 사용량)를 증가시켜 더 높은 수익을 창출하는 것이 목표입니다. 사용자에게는 좀 더 높은 현실성과 만족도를 제공하면서, 현재 점차 증가하고 있는 원예 산업에 활성화에 기여할 것이라 예상된다.

5. 자체 평가서

5.1 자체 분석과 평가

□ 진 연주

처음 해보니 큰 프로젝트이기 때문에 많은 어려움이 있을 것이라 생각했지만, 매주 회의를 하면서 진행进度와 앞으로 진행해야할 List를 작성하면서, 자칫 늘어질 수 있는 프로젝트가 잘 진행이 되었으며, 팀원들도 한 자신이 맡은 부분에 대해서 열심히 구현을 하고 매주 있는 회의에서 발표를 함으로서, 4월에는 프로젝트의 제안서 작성 및 발표 준비 때문에 구현을 잘 진행하지 못하고, 5월 초반부터 구현을 시작한 것에 비해 약 1달 간 짧은 기간 안에 충분히 1차 목표를 거의 구현하였다고 생각합니다.

□ 조 수정

팀원들과 자주 만나지 못 하더라도 온라인을 통해 회의를 하고, 각자 개발할 부분을 Github에 올려 공유하면서 원활하게 소통되었다고 생각한다. 5월에 프로젝트 진행이 확정되면서 약 한 달간 설계와 구현을 했지만 기간에 비해 완성도 있게 만들어진 것 같다. 개발하다 발생하는 오류가 있으면 팀원들과 소통하여 해결하였고, 자신이 잘 못하는 부분을 팀원이 해결 해 주거나, 팀원이 잘 못하는 부분을 알려주면서 한 달 간에 여기까지 완성한 것 같다.

□ 김 태우

첫 회의 때 개인별 자료조사를 바탕으로 한 구체적인 프로젝트 일정 계획을 잡고 그것에 맞추어 수행하다 보니 큰 문제점은 없었으나, 안드로이드를 처음 다뤄보면서 시작할 때 많은 어려움이 있었고, 안드로이드 스튜디오에서 발생한 여러 오류와 충돌로 인해 고생했다. 팀원 모두 맡은 역할을 착실히 수행하고 일주일마다 회의를 진행하며 의사소통한 결과 예상보다 완성도가 높은 작품이 나온 것 같다.

□ 임 준혁

팀 프로젝트를 진행하면서 팀원들과 개발을 함에 있어 오류가 나는 부분은 서로 피드백을 해주었고 DB설계나 게임 시스템 조정 등에 대해서도 팀원들끼리 의견을 모아서 더 좋은 쪽으로 계속 수정을 하여 프로젝트가 진행되는 데 큰 문제는 없었던 것 같다.

5.2 팀원 역할 및 기여도

□진 연주(30%) : 팀장

- Overlay Screen으로 설계변경 전, Widget 기능 자료 조사 및 유사 application 정보 조사
- Overlay Screen 부분 구현
- 전체적인 코드 정리

□조 수정(30%)

- 설계서에 맞게 Server와 DB 구축 및 Server와 Client와의 통신을 위한 Spec(주요받을 정보의 형식) 설계
- Android App에서 Volley library를 사용하여 Server와의 통신을 위한 코드 구현
- UI 설계 및 프로토타입을 위한 UI 구현
- App 내부 기능 구현(아이템 및 식물 구입, navigation view 등)

□김 태우(30%)

- UI 설계 및 프로토타입을 위한 UI 구현
- App 내부 기능 구현(아이템 및 식물 구입, navigation view 등)
- UI 제작 의뢰서 총괄

□임 준혁(10%)

- DB 설계 및 식물에 관한 자세한 정보 조사, 정리
-

6. 부록

□시스템 설계서

[부록 1] github에 있는 문서/GAIA 설계서 1.5 참고

□소스코드 핵심 부분

- Overlay Screen 부분

(1) Overlay Screen에 해당 식물 추가

```
public void onLongClick(final ImageView imageButton){
    Log.i(TAG,"*****1 "+imageButton.toString()+"/"+mWindowManager.toString());
    int [] location = new int[2];
    imageButton.getLocationOnScreen(location);
    final ImageView overlayButton = new ImageView(this);
    overlayButton.setImageResource((Integer)imageButton.getTag());
    final WindowManager.LayoutParams params = new WindowManager.LayoutParams(
        imageButton.getWidth(), imageButton.getHeight(),
        WindowManager.LayoutParams.TYPE_TOAST,
        WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE
WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL,
        PixelFormat.TRANSLUCENT);
    params.gravity = Gravity.LEFT|Gravity.TOP;
    params.x = location[0];
    params.y = location[1];

    Log.i(TAG,"*****1 "+overlayButton.toString()+"/"+mWindowManager.toString());

    mWindowManager.addView(overlayButton,params);

    overlayButton.setOnTouchListener(new View.OnTouchListener() {
        @Override
        public boolean onTouch(View view, MotionEvent motionEvent) {
            if (motionEvent.getAction() == MotionEvent.ACTION_DOWN) {
                Log.i(TAG,"*****"+motionEvent.toString());
                float x = motionEvent.getRawX();
                float y = motionEvent.getRawY();

                moving = false;

                int[] location = new int[2];
                //overlayedButton.getLocationOnScreen(location);
                view.getLocationOnScreen(location);

                originalXPos = location[0];
                originalYPos = location[1];

                offsetX = originalXPos - x;
                offsetY = originalYPos - y;

            } else if (motionEvent.getAction() == MotionEvent.ACTION_MOVE) {
                int[] topLeftLocationOnScreen = new int[2];
                topLeftView.getLocationOnScreen(topLeftLocationOnScreen);
```

```

        System.out.println("topLeftY="+topLeftLocationOnScreen[1]);
        System.out.println("originalY="+originalYPos);

        float x = motionEvent.getRawX();
        float y = motionEvent.getRawY();

        WindowManager.LayoutParams params = (WindowManager.LayoutParams)
view.getLayoutParams();

        int newX = (int) (offsetX + x);
        int newY = (int) (offsetY + y);

        if (Math.abs(newX - originalXPos) < 1 && Math.abs(newY - originalYPos) <
1 && !moving) {
            return false;
        }

        params.x = newX - (topLeftLocationOnScreen[0]);
        params.y = newY - (topLeftLocationOnScreen[1]);

        mWindowManager.updateViewLayout(view, params);
        moving = true;
    } else if (motionEvent.getAction() == MotionEvent.ACTION_UP) {
        if (moving) {
            mArray.add(new PlantInfo(params,(ImageView)view,imageButton.getId()));
            Intent intent = new Intent(OverlayService.this,StartActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
        }
    }

    return false;
}
});
}

```

□개발 일지(회의록 포함)

[부록2] 참고

참고문헌

- Overlay Screen 오픈 소스

<https://gist.github.com/bjoernQ/6975256>

[부록2. 회의록]

회 의 록

회 의 명	■ 네오플 프로젝트 1차 회의		
일 시	17. 05. 03	장 소	임준혁 학생 자취방
팀 명	NeoPul	대 표	진연주
참 석 자	■ 총 4명: 금오공과대학교 진연주, 임준혁, 조수정, 김태우		
회의내용	<ol style="list-style-type: none">프로젝트 일정 & 지원금, 서류 확인<ul style="list-style-type: none">7월 창업캠프(중간발표), 10월 최종발표운영지침에 맞게 서류정리 잘할것!기자재 구입을 위해, 기자재 가격 알아볼 것프로젝트 준비 회의<ul style="list-style-type: none">어플에 필요한 기능 정리어플에 들어갈 화면과 각 화면에서 할 수 있는 일 정리총 개발단계를 8개로 나누웠고, 우선적으로 다같이 어플 화면을 구현한 후 역할을 나누기로 함회의관련<ul style="list-style-type: none">개발 기간동안 가능한 1주~2주에 한번 회의 진행예정시험기간 동안은 잠시 개발중단거리가 먼 학생이 있어서 모이기 힘들다면 온라인회의로 진행 (스카이프 사용)		

Neo플 팀 회의 참석자 명단 (17. 5. 3)

구분	성명	소 속	서 명
1	진연주	금오공과대학교	
2	김태우	금오공과대학교	
3	조수정	금오공과대학교	
4	임준혁	금오공과대학교	
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

회의록

회의명	■ 네오플 프로젝트 2차 회의		
일시	17. 05. 10	장소	온라인(스카이프)
팀명	NeoPul	대표	진연주
참석자	■ 총 4명: 금오공과대학교 진연주, 임준혁, 조수정, 김태우		
회의내용	1. 어플리케이션 기본구성, 기초코드 작성		
	■ 정원화면, 관리화면, 상점화면 구성		
	■ 기초코드 공유 및 해당부분 설명		
	■ 코드 수정,추가를 위해 토의		
	2. 2팀으로 나뉘어 메인앱, 추가기능개발을 진행하기로 함		
	김태우, 조수정	UI 구현, Application 내부 기능	6월 15일
	임준혁, 진연주	DB 구현, 움직이는 Widget	6월 15일
	임준혁, 조수정	메인화면	중간 발표(7월)
	김태우, 진연주	Calendar	중간 발표(7월)
	진연주, 조수정	User 관리	8월 내로
	김태우, 임준혁	날씨/일정 알림 기능	8월 내로
	전원	추가기능, 및 에러처리, 보수작업	8월 말
	전원	테스팅 및 어플수정	결과발표(10월)

Neo풀 팀 회의 참석자 명단 (17. 5. 10)

구분	성명	소 속	서 명
1	진연주	금오공과대학교	
2	김태우	금오공과대학교	
3	조수정	금오공과대학교	
4	임준혁	금오공과대학교	
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

회 의 록

회 의 명	■ 네오플 프로젝트 3차 회의		
일 시	17. 05. 18	장 소	임준혁 학우 자취방
팀 명	NeoPul	대 표	진연주
참 석 자	■ 총 4명: 금오공과대학교 진연주, 임준혁, 조수정, 김태우		
회의내용	<ol style="list-style-type: none"> 각 팀 진행상황 발표 <ul style="list-style-type: none"> ■ 임준혁,진연주 조 DB구성 완료 ■ 조수정,김태우 조 어플리케이션 뼈대 완성, 내부기능 구상 UI제작 계획서 - 김태우,조수정 <ul style="list-style-type: none"> ■ 이미지 제작이 필요한 부분 체크 ■ 꽃 성장 관련 이미지 추가논의 필요 예산변경 신청서 변경내용 토의 <ul style="list-style-type: none"> ■ 금액 분배변경에 대해 논의 시스템 변경 사항 토의 <ul style="list-style-type: none"> ■ 식물의 성장 단계를 기본적으로 절대값, 후에 수정하면서 식물별로 성장단계를 다르게 하기로 함 ■ 아이템사용으로 경험치를 증가시킬까? <ul style="list-style-type: none"> > 자동돌봄기능이나, 일정시간동안 경험치량 증가로 수정 ■ 다 키운 식물에서 일정시간마다 열매 수급, 응모기능 부족하거나 급한 기능 정리 <ul style="list-style-type: none"> ■ 내부기능구현, 위젯개발이 촉박함 좀 더 투자해야함 ■ 빠른 DB구현 예정 		

Neo풀 팀 회의 참석자 명단 [17. 5. 18]

구분	성명	소 속	서 명
1	진연주	금오공과대학교	
2	김태우	금오공과대학교	
3	조수정	금오공과대학교	
4	임준혁	금오공과대학교	
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

회 의 록

회 의 명	■ 네오플 프로젝트 4차 회의		
일 시	17. 05. 26	장 소	온라인(스카이프)
팀 명	NeoPul	대 표	진연주
참 석 자	■ 총 4명: 금오공과대학교 진연주, 임준혁, 조수정, 김태우		
회의내용	<ol style="list-style-type: none">현재진행사항<ul style="list-style-type: none">로그인,회원가입 페이지 완성이미지, 디자인 일부 수정,보수부족한 기능 구현이 시급해보임<ul style="list-style-type: none">데이터연동(DB), 각기능들과 연결내부관리기능 추가로 먼저 구현할 것UI제작비용에 맞춰 UI계획서 수정할 필요있음예산변경신청서, 계획변경신청서 제출했음위젯개발<ul style="list-style-type: none">위젯기능 대신 Overlay screen 사용하기로 결정임준혁 학우가 Overlay screen 조사7월 창업캠프 일정이 나오면 바로 공지해줄 것		

Neo풀 팀 회의 참석자 명단 (17. 5. 26)

구분	성명	소 속	서 명
1	진연주	금오공과대학교	
2	김태우	금오공과대학교	
3	조수정	금오공과대학교	
4	임준혁	금오공과대학교	
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

회 의 록

회 의 명	■ 네오플 프로젝트 5차 회의		
일 시	17. 06. 02	장 소	온라인(스카이프)
팀 명	NeoPul	대 표	진연주
참 석 자	■ 총 4명: 금오공과대학교 진연주, 임준혁, 조수정, 김태우		
회의내용	<p>1. 5월 진행 사항 정리</p> <ul style="list-style-type: none"> ■ 로그인, 회원가입 화면 완성 ■ 식물 자동 배치 기능 구현 ■ 식물/화분/아이템 구입 기능 구현 ■ 식물 내부 관리 기능 완료 ■ OverlayScreen을 이용한 View 이동 기능 구현 ■ DB 설계 완료 <p>2. ToDoList 정리</p> <p>Server</p> <ul style="list-style-type: none"> ■ Server, DB 구축, 식물의 특성을 정해서 DB에 추가 및 연동 <p>App</p> <ul style="list-style-type: none"> ■ Overlay 기능과 내부 기능 합병, 기상정보를 앱에 연동, 기본 식물에 대한 관리 기능 구현 <p>3. UI 제작 계획</p> <ul style="list-style-type: none"> ■ 전문 업체를 통해 식물 성장 단계에 따른 정보와 식물 이미지 외주를 계획(예정 6/30, 6/23까지 조사:임준혁) ■ 식물, 화분 종류 결정(튜토리얼 식물 : 민들레) + 각자 식물, 화분 하나씩 조사 ■ 외주를 하지 않는 UI 찾아보기(로그인, 로딩, 회원가입 화면) 		

4. 5월 창업지원금 지출 내역 정리

일자	구분	지출	잔액
5월 2일	창업지원금 입금	-	8,000,000원
5월 2일	활동비(회의 후 식비)	48,000원	7,952,000원
5월 3일	활동비(회의 후 식비)	100,800원	7,851,200원
5월 18일	활동비(회의 후 식비)	32,000원	7,819,200원
5월 29일	활동비(회의 후 식비)	92,000원	7,727,200원

5. GitHub에 초대

Neo플 팀 회의 참석자 명단 (17. 6. 2)

구분	성명	소 속	서 명
1	진연주	금오공과대학교	
2	김태우	금오공과대학교	
3	조수정	금오공과대학교	
4	임준혁	금오공과대학교	
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			