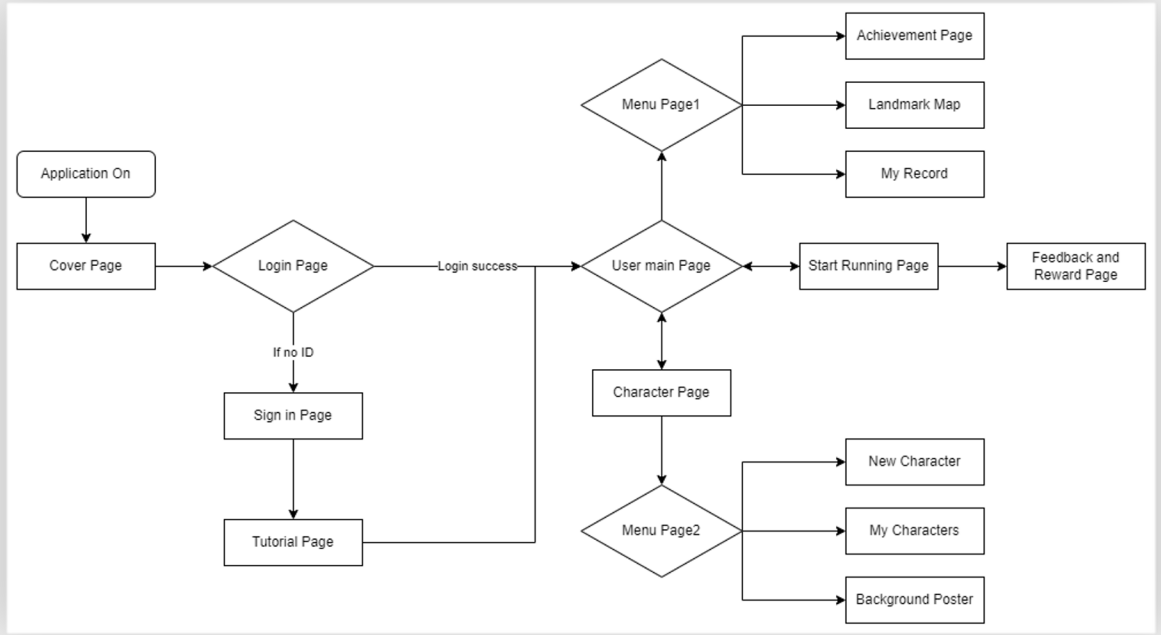
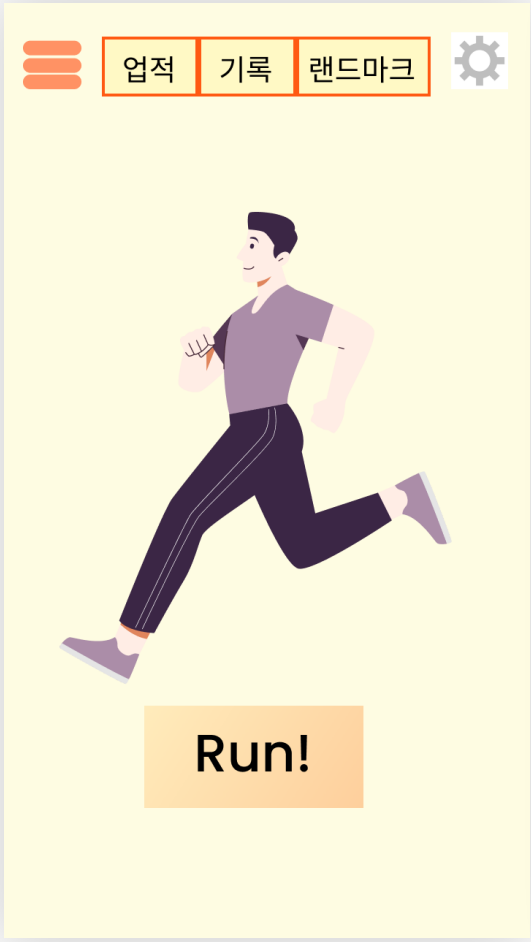
이상해 C

강승목, 김준석, 박재윤, 이혜원

Front-end


Application On
Cover Page
Login Page
Login success
If no ID
Sign in Page
Tutorial Page
Menu Page1
Achievement Page
Landmark Map
My Record
User main Page
Start Running Page
Feedback and Reward Page
Character Page
Menu Page2
New Character
My Characters
Background Poster

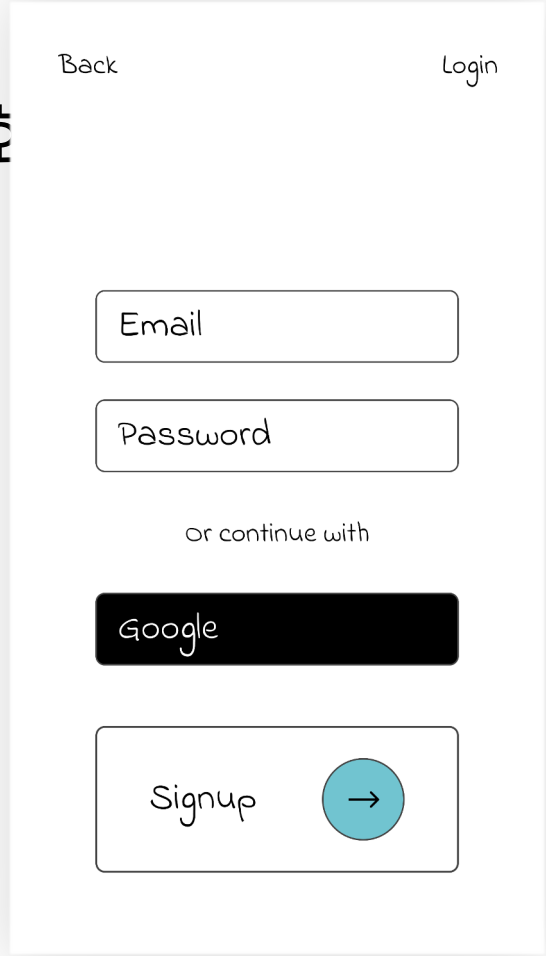

업적
기록
랜드마크
Run!


뽑기
보관함
포스터
경험치
애정도
...
먹이
놀아주기

Front-end

디자인 및 레이아웃 수정

1. 상단에 메뉴 또는 버튼을 최소화
2. Flow의 Depth를 최소화
3. 직관적이고 심플하게



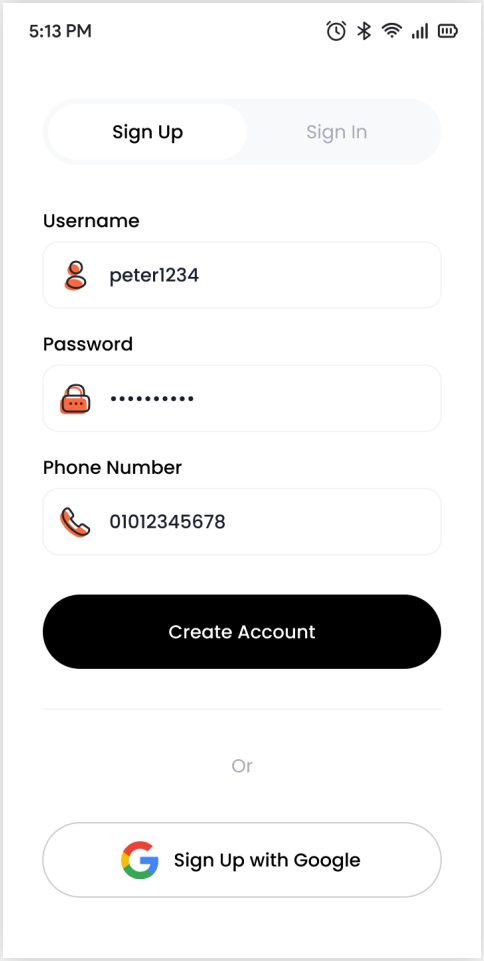
테마

폰트: Poppins

1. Sans-serif (고딕체)의 기본 틀
2. 부드럽고 생동감 부여
3. 오픈 라이센스

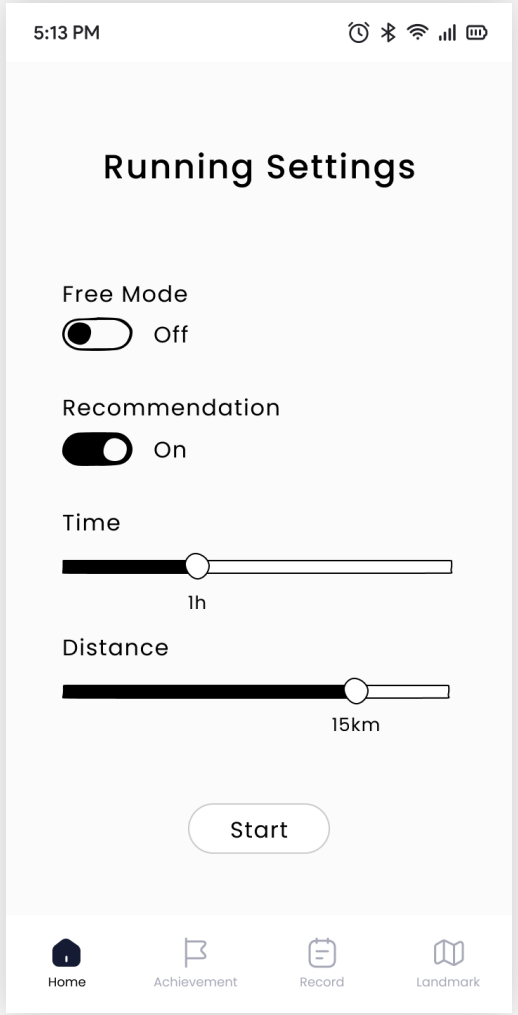
컬러:

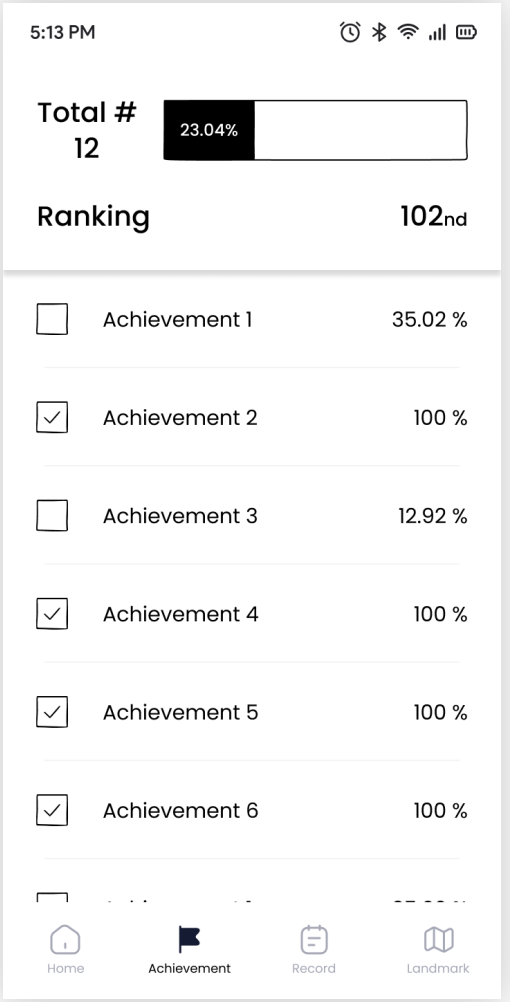
- Gray-scale 베이스
- 포인트 색상으로 단조롭지 않게



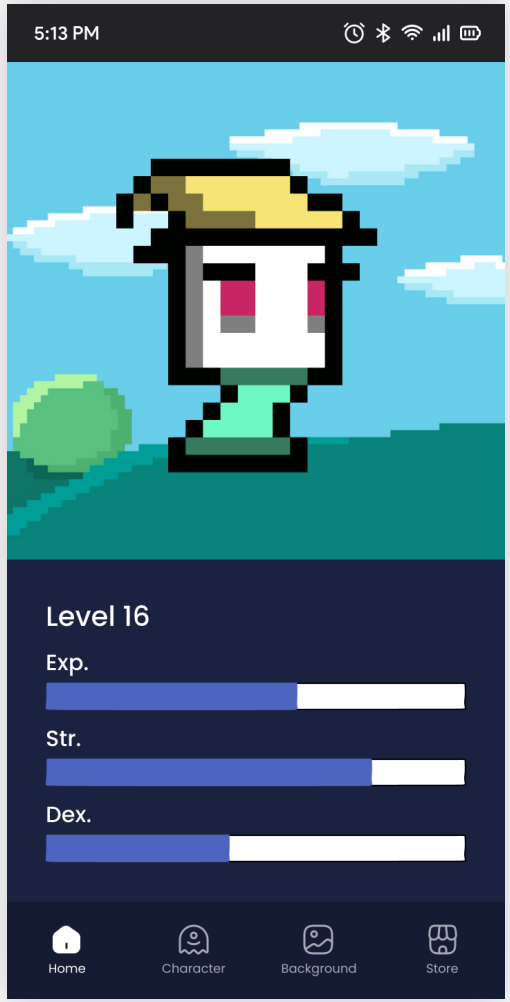
테마 - Running

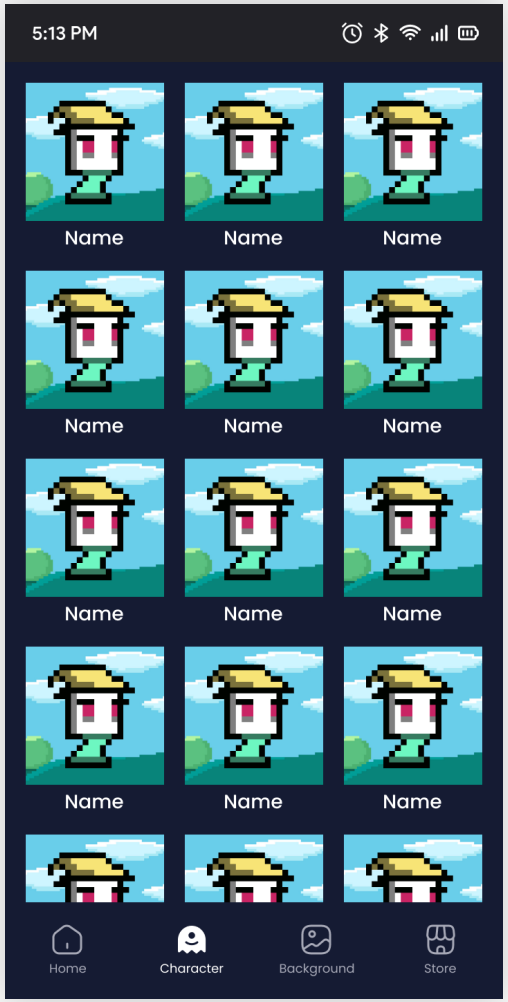

5:13 PM
00:00:00
Time
0.00
km
0.00
%
00:00
minute/km
제1공학관
보통신대학
화산
어린이공원
124
공학실습동
성균관대학교
제2종합연구동
108
125
주차장3
제1종합연구동
126
성균관대학교
약학관
반도체관
약학대학
화학관
DRC관
127
119
RUN
128
웅지어린이집
삼나빌라
명문가
129
122
Home
Achievement
Record
Landmark

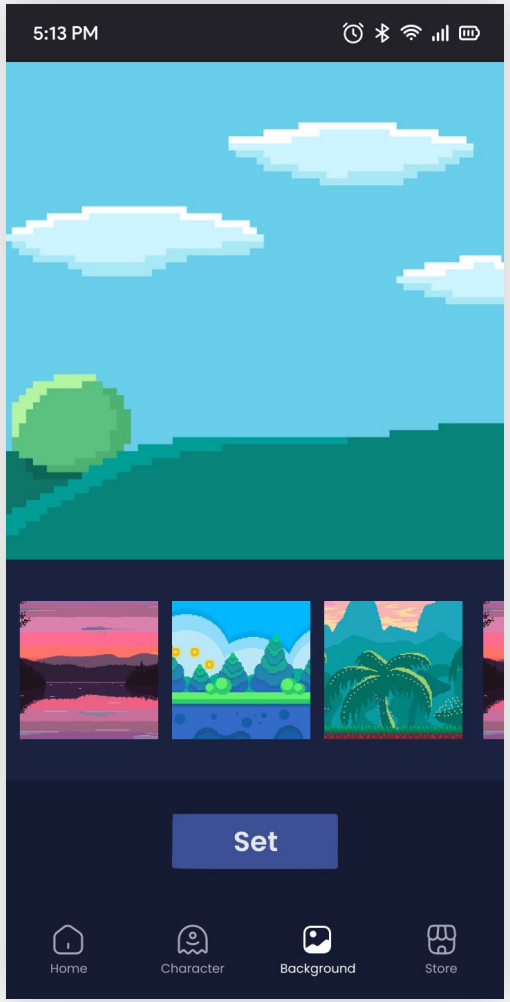

5:13 PM
Running Settings
Free Mode
Off
Recommendation
On
Time
1h
Distance
15km
Start
Home
Achievement
Record
Landmark


5:13 PM
Total #
12
23.04%
Ranking
102nd
Achievement 1
35.02 %
Achievement 2
100 %
Achievement 3
12.92 %
Achievement 4
100 %
Achievement 5
100 %
Achievement 6
100 %
Home
Achievement
Record
Landmark

테마 - Gaming


5:13 PM
Level 16
Exp.
Str.
Dex.
Home
Character
Background
Store


5:13 PM
Name
Home
Character
Background
Store


5:13 PM
Set
Home
Character
Background
Store

Prototype

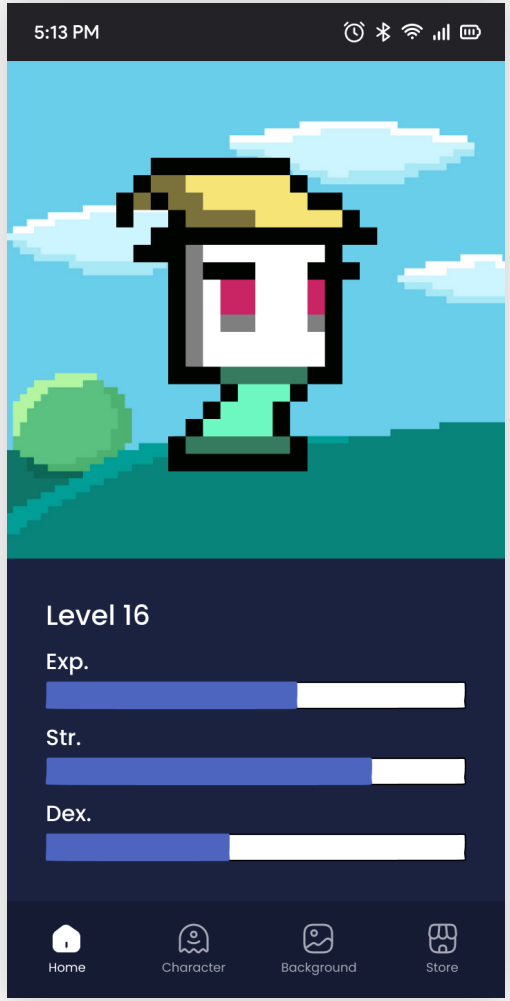
- Figma Prototype

Consideration

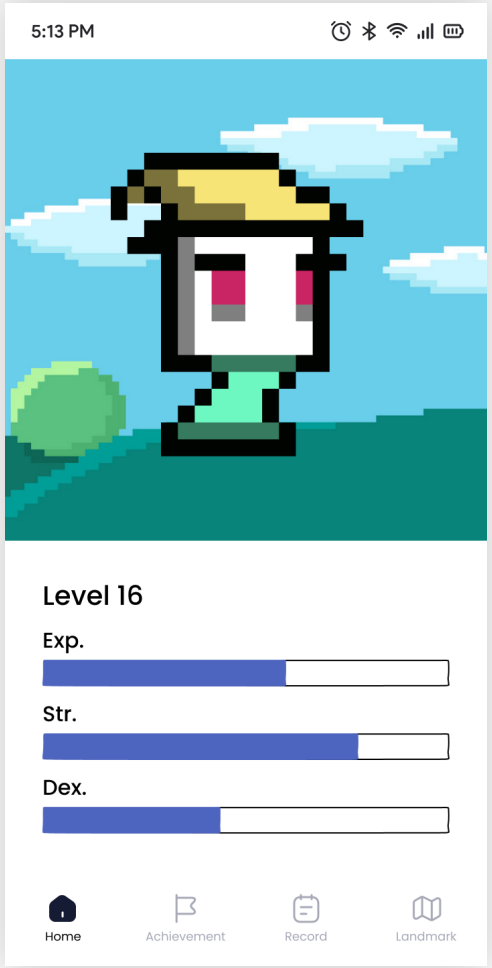
각 부분의 테마를 구분
→ 다른 2개의 앱인 느낌

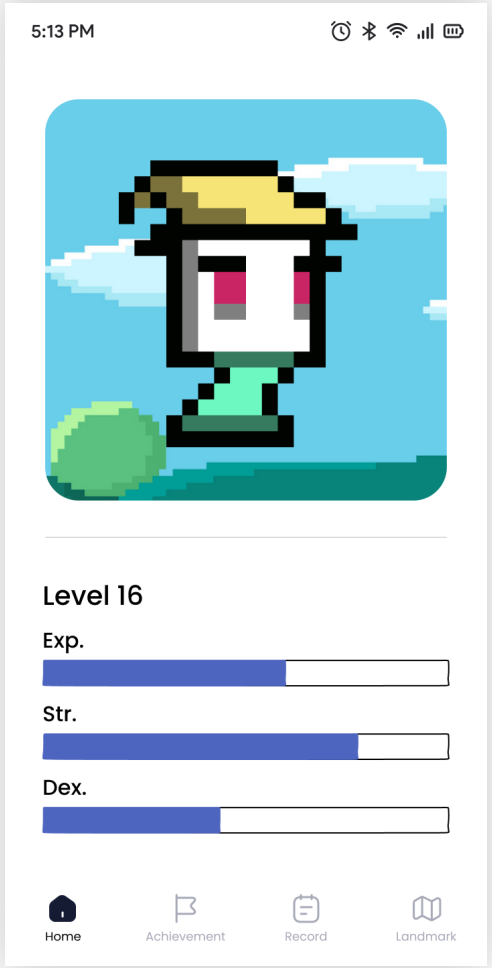
다른 시도
1. 배경을 둘 다 화이트로 가져간다
→ 픽셀 아트가 튀어 조화롭지 않다

2. 픽셀 아트의 배경을 전체 바탕으로 사용한다
→ 배경이 보이지 않음

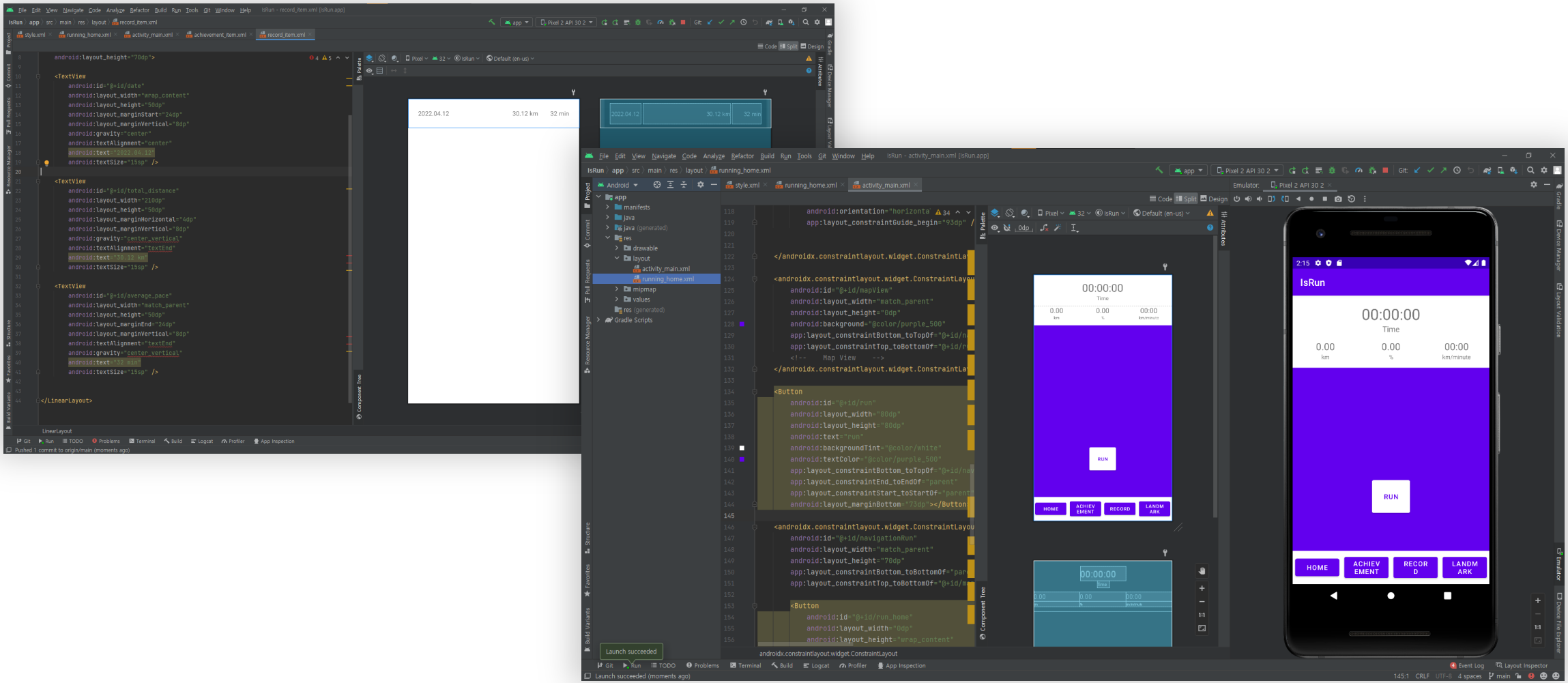


Consideration





구현


2022.04.12
30.12 km
32 min
00:00:00
Time
0.00
km
0.00
%
00:00
km/minute
RUN
HOME
ACHIEV EMENT
RECORD
LANDM ARK
IsRun
Launch succeeded

Back-end

- AWS SSH 연결을 통한 개발환경 구축
- 의존성 라이브러리 설치
- PAHO MQTT 라이브러리 기반 데이터베이스 스켈레톤 구축

```python
def mainClient_connect(client, userdata, flags, rc):
    if rc == 0:
        print("connected OK")
    else:
        print("Bad connection Returned code=", rc)

def mainClient_disconnect(client, userdata, flags, rc=0):
    print(str(rc))

def mainClient_subscribe(client, userdata, mid, granted_qos):
    print("subscribed: " + str(mid) + " " + str(granted_qos))

def mainClient_message(client, userdata, msg):
    print(str(msg.payload.decode("utf-8")))
    # if문으로 메시지 분리후
    # thread로 분리해서 처리

def Create_Game_Client(MQTT_BROKER, PORT):
    # 메인 쓰레드 클라이언트 선언
    client = mqtt.Client()
    # callback 함수 덮어쓰기
    client.on_connect = mainClient_connect
    client.on_disconnect = mainClient_disconnect
    client.on_subscribe = mainClient_subscribe
    client.on_message = mainClient_message
    # MQTT브로커에 연결
    client.connect(MQTT_BROKER, PORT)
    client.subscribe("IsRun/Data", 2)
    # 서버 loop 시작
    client.loop_forever()

if __name__ == "__main__":
    MQTT_BROKER = 'broker.mqttdashboard.com'
    PORT = 1883

    file_copy_thread = threading.Thread(MQTT_BROKER=file_copy, PORT=(src_file_name, des_file_name))
    file_copy_thread.start()
    print("Is Run Server Started")

    file_copy_thread.
```

Next Progress

Front-end

- App의 모든 페이지 UI 구현
- Kakao GPS API 데이터 테스트 및 Rendering

Back-end

- DB Schema에 따른 모델 구축 및 세팅
- API Call Design (Front 와의 통신 테스트)

Q & A

감사합니다