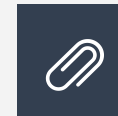
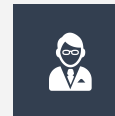
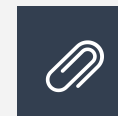
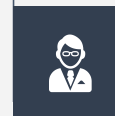


PT Manager



컴퓨터공학 201333686 이재호



컴퓨터공학 201333686 고수균

INDEX

1

시작 배경

2

시장 조사

3

어플리케이션 소개

4

사용 기술

5

시나리오

1. 시작 배경



무인 주차장의 등장으로 전국적으로 많은 주차장들이 생기게 되었다.

1. 시작 배경



대부분의 멀티 플렉스에서도 무인 주차장을 도입했다.

1. 시작 배경 - 현황(문제점)



입차 시간을 확인하기가 어렵다.

1. 시작 배경 - 현황(문제점)



주차 요금 할인에 대한 정보가 잘 제공되지 않는다.

1. 시작 배경 - 현황(문제점)



주차 요금 할인을 적용하기가 불편하다.

1. 시작 배경 - 현황(문제점)



정산기에서만 결제가 가능하기 때문에 결제가 불편하다.

1. 시작 배경



2. 시장 조사



1. iParking

출처 : <https://play.google.com/store/apps/details?id=kr.co.iparking.android&hl=ko>

2. 시장 조사



2. 파킹박

출처 : <https://play.google.com/store/apps/details?id=kr.wisemobile.parking&hl=ko>

3. 어플리케이션 소개

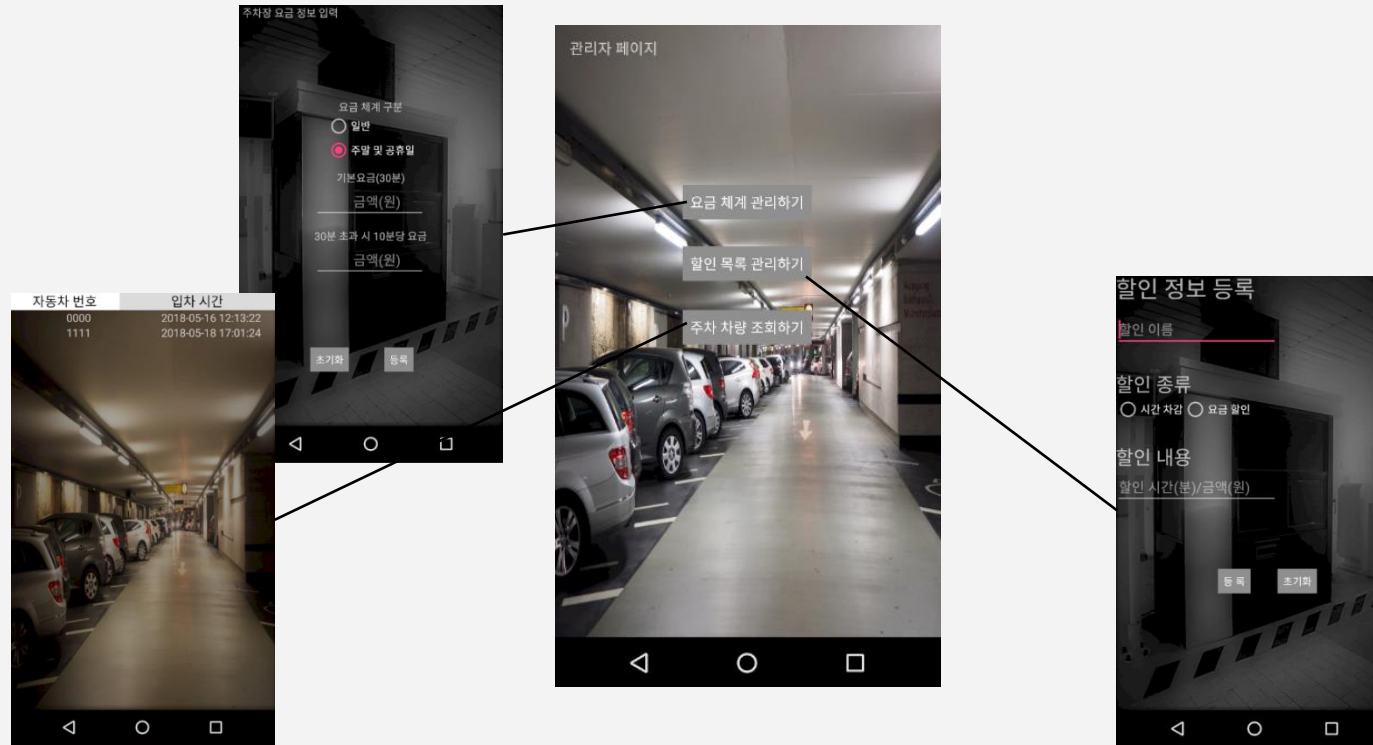
1. 로그인



Firebase 인증을 통해 Google 계정으로 회원 가입 및 등록

3. 어플리케이션 소개

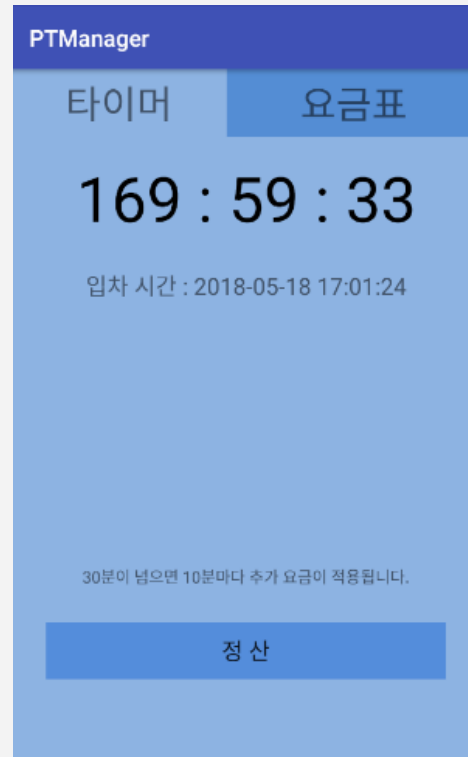
2. 관리자



요금 / 할인 정보 등록 및 현재 주차중인 차량 조회 가능

3. 어플리케이션 소개

3. 타이머



입차 시간을 기준으로 사용자의 현재 주차 시간을 시간과 공간에 관계 없이 확인 가능

3. 어플리케이션 소개

4. 요금 및 할인정보 확인

타이머		요금표
공휴일	1500원	300원
평일	1000원	200원

기본 요금은 30분까지입니다.
30분이 넘으면 10분마다 추가 요금이 적용됩니다.

이름	할인 종류	할인량
식사 할인	요금 할인	2000원
영화 할인	시간 차감	120분



등록되어 있는 요금 및 할인 정보를 조회 가능

3. 어플리케이션 소개

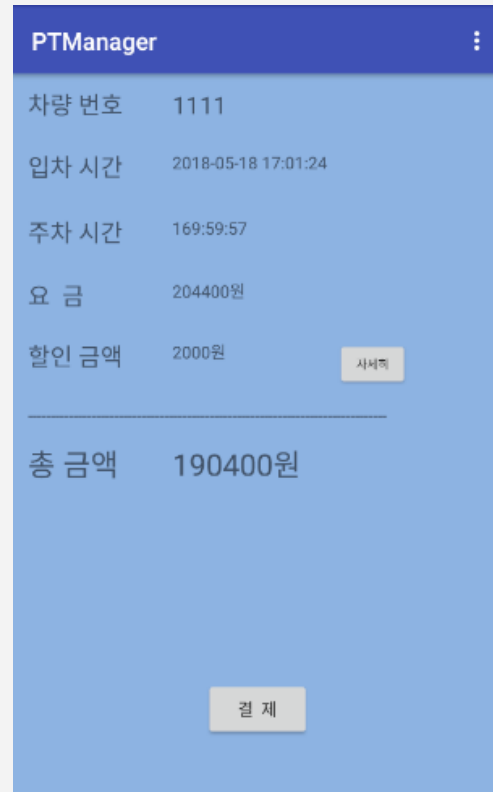
5. 마이 페이지



고객의 ID, 자동차 번호, 카드 번호를 확인이 가능하고 자동차 번호를 수정 가능

3. 어플리케이션 소개

6. 정산



The screenshot shows the PTManager app interface for calculating parking fees. It features a blue header with the title 'PTManager' and a menu icon. Below the header, there are several rows of data: '차량 번호' (Vehicle Number) 1111, '입차 시간' (Entry Time) 2018-05-18 17:01:24, '주차 시간' (Parking Time) 169:59:57, '요금' (Fee) 204400원, and '할인 금액' (Discount Amount) 2000원. A '사세액' (Discount Amount) button is next to the discount amount. A horizontal line separates the individual items from the '총 금액' (Total Amount) of 190400원. At the bottom, there is a '결제' (Payment) button.

차량 번호	1111
입차 시간	2018-05-18 17:01:24
주차 시간	169:59:57
요금	204400원
할인 금액	2000원
<hr/>	
총 금액	190400원

결제

주차 시간에 따른 등록된 요금 체계에 따라 주차 요금을 계산
할인 적용이 가능할 시 자동적으로 할인 적용

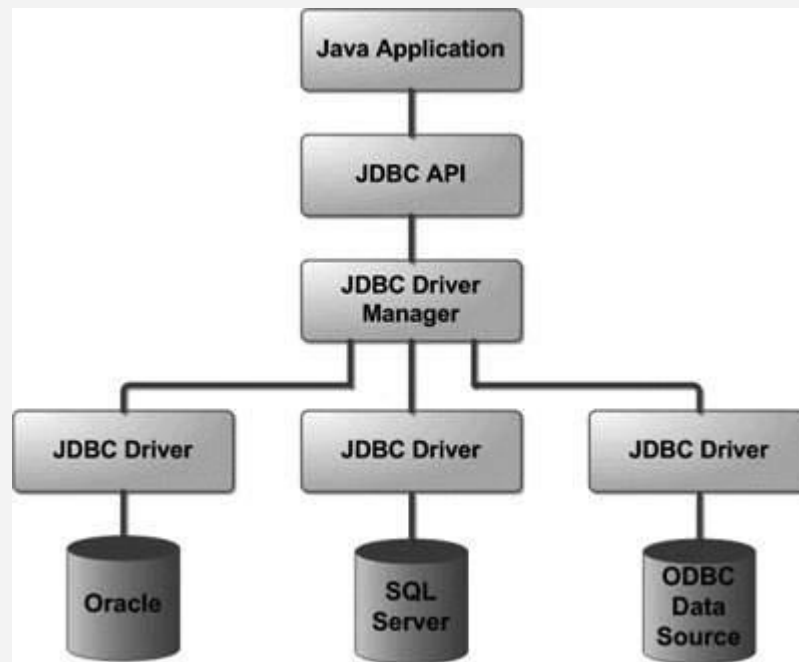
4. 사용 기술

1. JDBC

- Java DataBase Connectivity의 약자
- Java와 DB를 연결하기 위한 API
- JDBC를 이용하면 Java에서 SQL문을 사용해서 DB에 접근이 가능

4. 사용 기술

1. JDBC



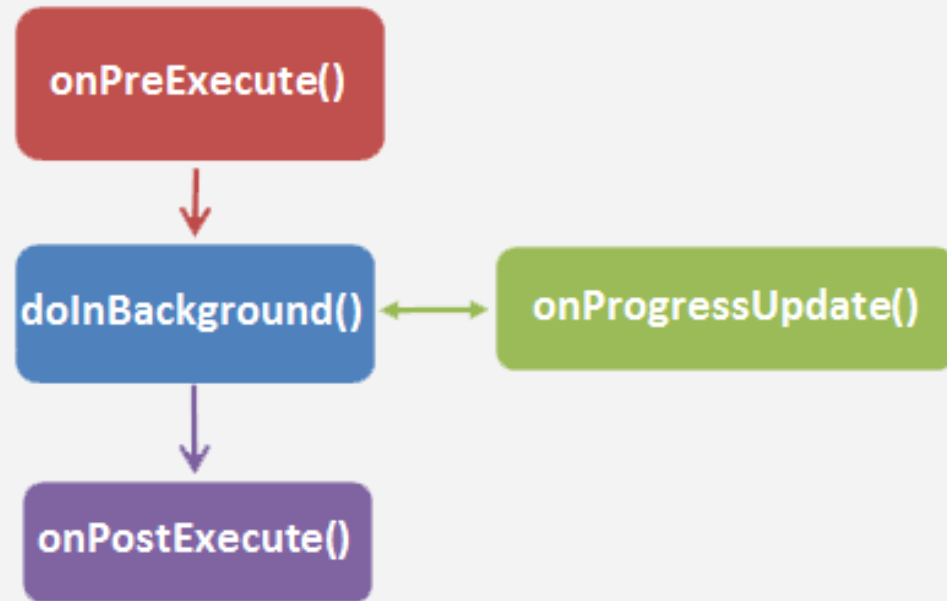
4. 사용 기술

2. AsyncTask

- Android 는 메인 스레드를 주축으로 한 싱글 스레드 모델이기 때문에 오래 걸리는 작업을 실행하면 앱의 성능이 떨어짐
- DB 작업처럼 오래 걸리는 작업들은 서브(백그라운드) 스레드에서 처리하는 게 유리
- 이렇게 백그라운드에서 작업을 처리하기 위해서는 핸들러나 AsyncTask 등을 이용해야 함

4. 사용 기술

2. AsyncTask



4. 사용 기술

3. Zxing

- Zebra Crossing의 약자
- 바코드와 QR코드를 인식하고 생성하는 API
- Zxing의 기술 중 바코드 관련 기술에서는 15가지 종류의 바코드를 생성 및 인식 가능



4. 사용 기술

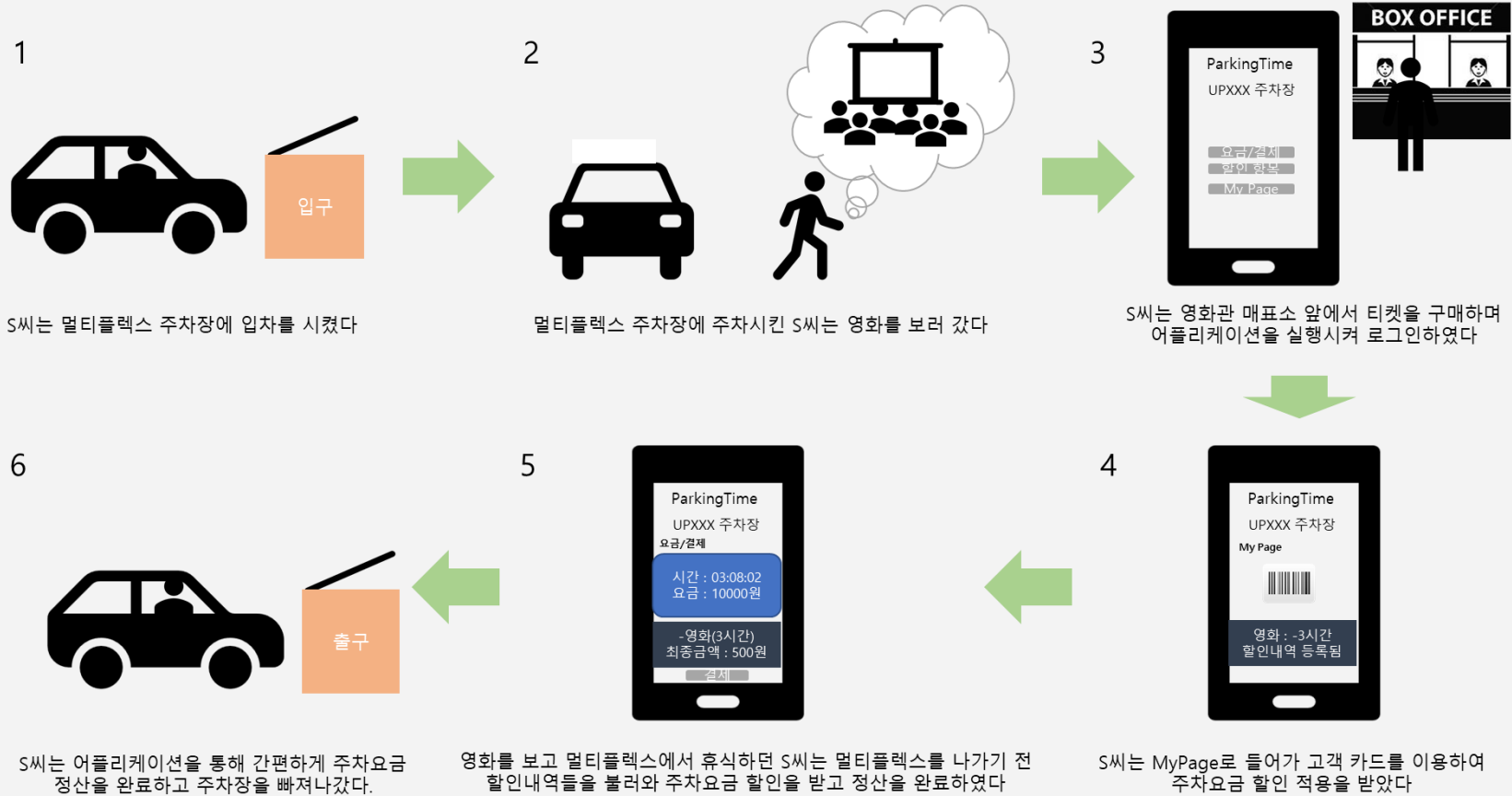
3. Zxing

```
public Bitmap createBarcode(String code) {
    Bitmap bitmap = null;
    MultiFormatWriter gen = new MultiFormatWriter();
    try {
        final int WIDTH = 1280;
        final int HEIGHT = 320;
        BitMatrix bytemap = gen.encode(code, BarcodeFormat.CODE_128, WIDTH, HEIGHT);
        bitmap = Bitmap.createBitmap(WIDTH, HEIGHT, Bitmap.Config.ARGB_8888);
        for (int i = 0; i < WIDTH; ++i)
            for (int j = 0; j < HEIGHT; ++j) {
                bitmap.setPixel(i, j, bytemap.get(i, j) ? Color.BLACK : Color.WHITE);
            }
    } catch (Exception e) {
        e.printStackTrace();
    }

    return bitmap;
}
```

5. 시나리오

1. 고객의 경우



5. 시나리오

2. 관리자의 경우

