

자율주행 도서관 서비스 로봇

20183090 위지훈 (팀장)
20183043 하재민
20183051 오진서



Literature presentation

Contents

/ 01

프로젝트를 선택한
배경

/ 02

프로젝트 목표

/ 03

프로젝트 수행과정

/ 04

프로젝트 성과물

/ 05

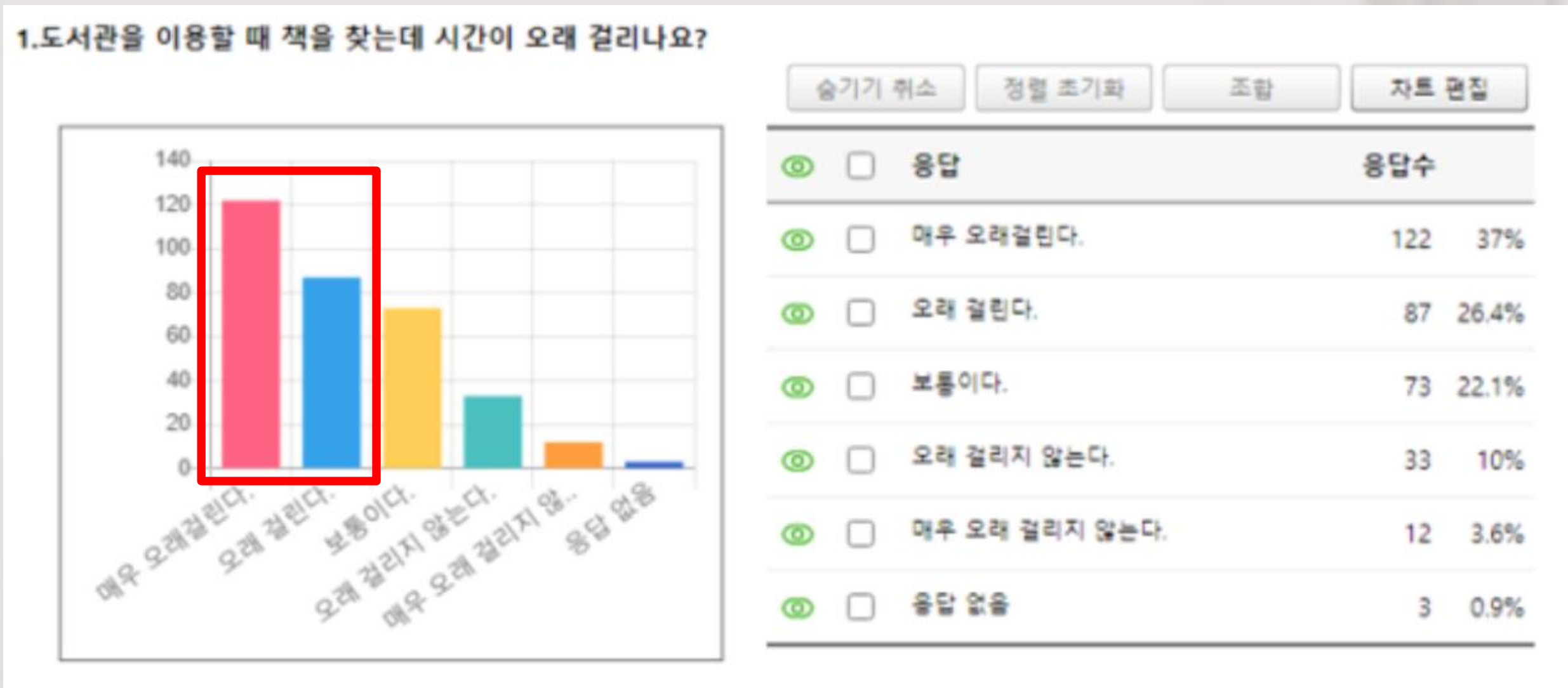
프로젝트 결론

프로젝트를 선택한 배경

도서관을 사용하면서 느낀 불편함

- 책을 찾는데 오래걸림
- 대출 /반납이 불편함
- 도서 검색을 할 수 있는 기기가 한정되어 있어 불편함

프로젝트를 선택한 배경



책을 찾는데 오랜 시간이 걸린다는 응답 총 63.4%

프로젝트를 선택한 배경

2. 저희는 자율 주행 도서관 서비스 로봇을 개발하여 대여/반납의 자율화, 원하는 도서의 위치 탐색, 도서 검색 등과같은 기능을 추가할 계획입니다. 도서관에 본 로봇이 비치된다면 자주 사용할 것 같은가요?



<input type="button" value="슬기기 취소"/>	<input type="button" value="정렬 초기화"/>	<input type="button" value="조합"/>	<input type="button" value="차트 편집"/>
<input checked="" type="radio"/>	<input type="checkbox"/>	응답	응답수
<input checked="" type="radio"/>	<input type="checkbox"/>	매우 그렇다.	155 47%
<input checked="" type="radio"/>	<input type="checkbox"/>	그렇다.	105 31.8%
<input checked="" type="radio"/>	<input type="checkbox"/>	보통이다.	62 18.8%
<input checked="" type="radio"/>	<input type="checkbox"/>	그렇지 않다.	3 0.9%
<input checked="" type="radio"/>	<input type="checkbox"/>	매우 그렇지 않다.	2 0.6%
<input checked="" type="radio"/>	<input type="checkbox"/>	응답 없음	3 0.9%

대여/반납의 자율화, 도서 위치 탐색, 도서 검색과 같은 기능들이 추가된 로봇이 비치되면 사용할것 같은가에 대한 긍정적인 응답 총 78.8%

프로젝트를 선택한 배경

3. 마지막으로 저희 프로젝트에 추가 되었으면 하는 기능이나 의견을 자유롭게 작성해 주세요.

대여한 도서를 로봇이 들고 이동할 수 있으면 좋을거 같다.

도서관의 규모가 클 수록 책을 찾기 힘들고 다시 대여하러 사서가 있는곳 까지 가기 힘들어 유용하게 쓰일거 같다.

자유롭게 돌아다니다가 원하는책을 바로바로 대여할 수 있을거 같아 좋아 보인다

설문자들이 제시한 로봇의 수납기능을 채택

프로젝트 목표

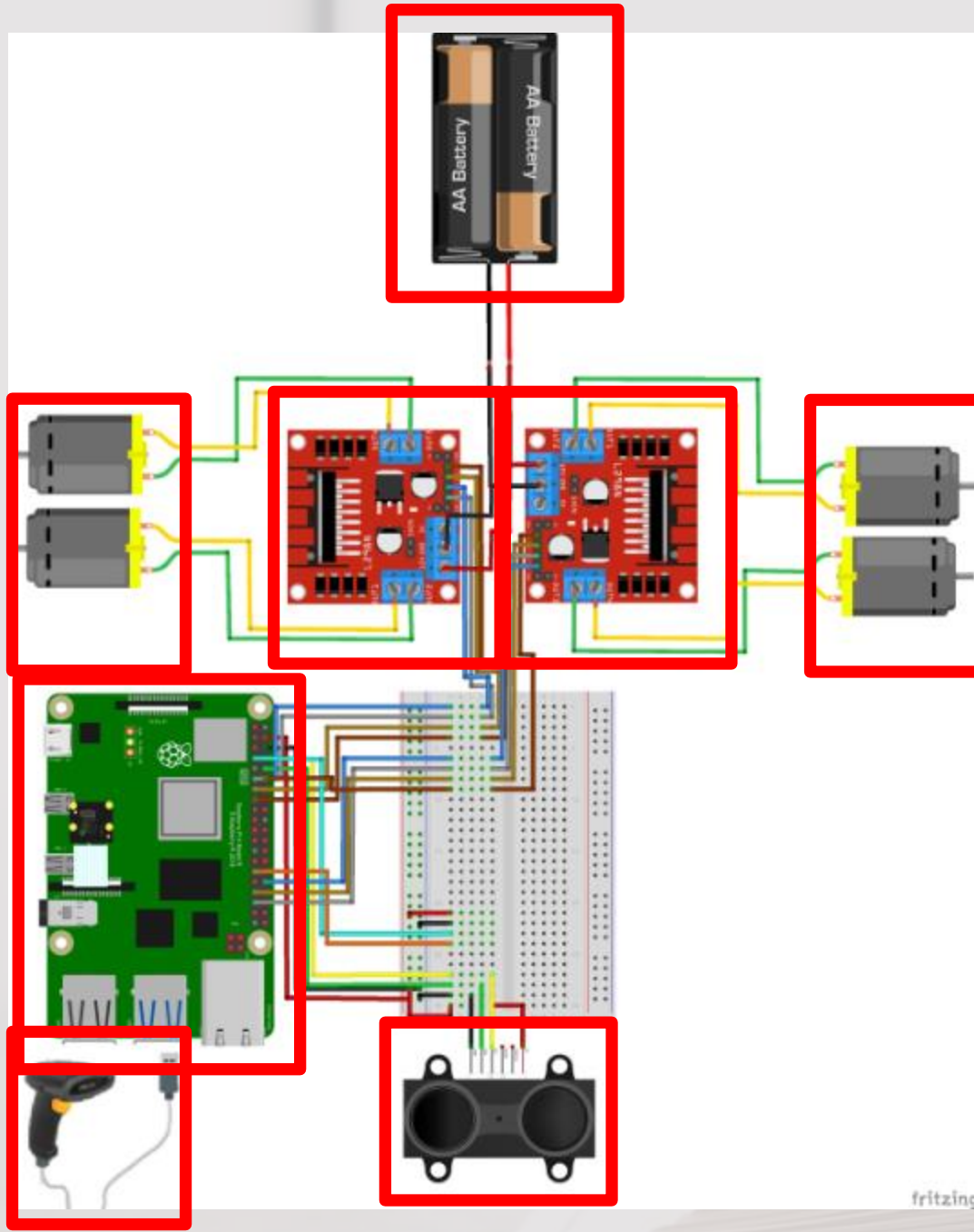
앱

- 회원가입
- 로그인
- 도서 대출/ 반납
- 회원정보
- 도서 검색
- 도서 위치 안내

하드웨어

- 자율주행
- 도서 수납

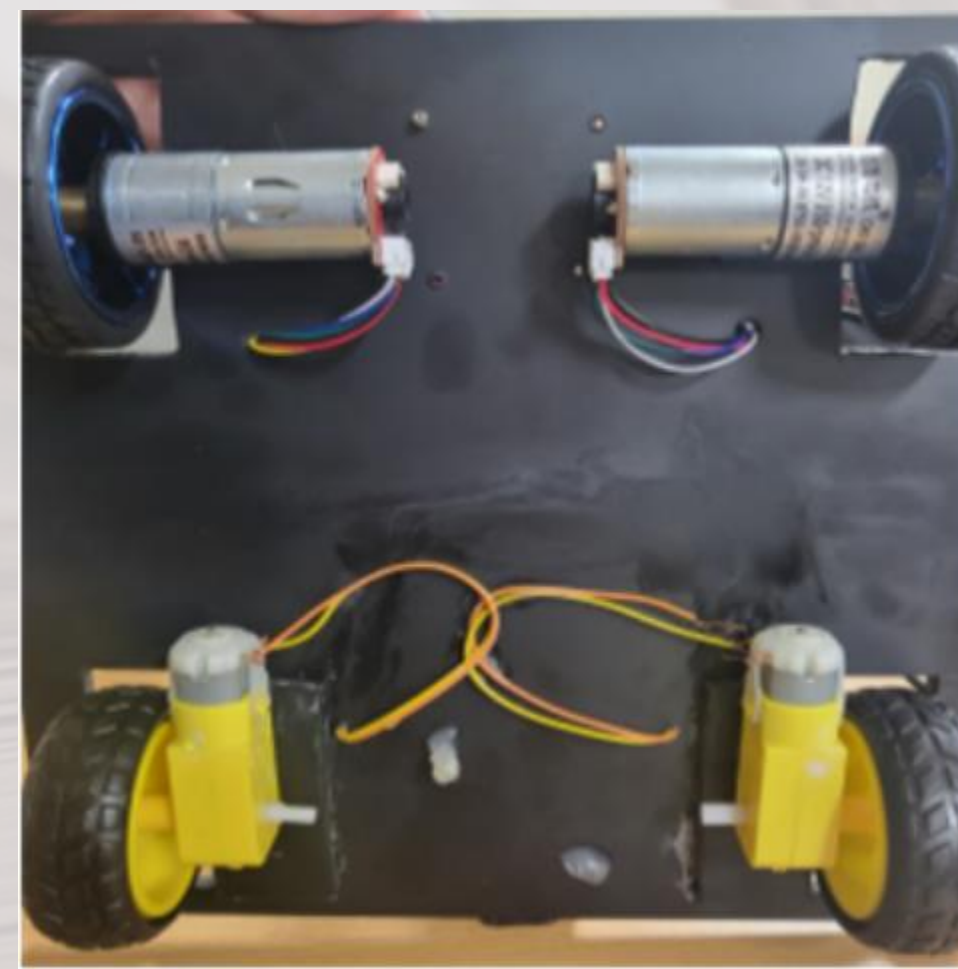
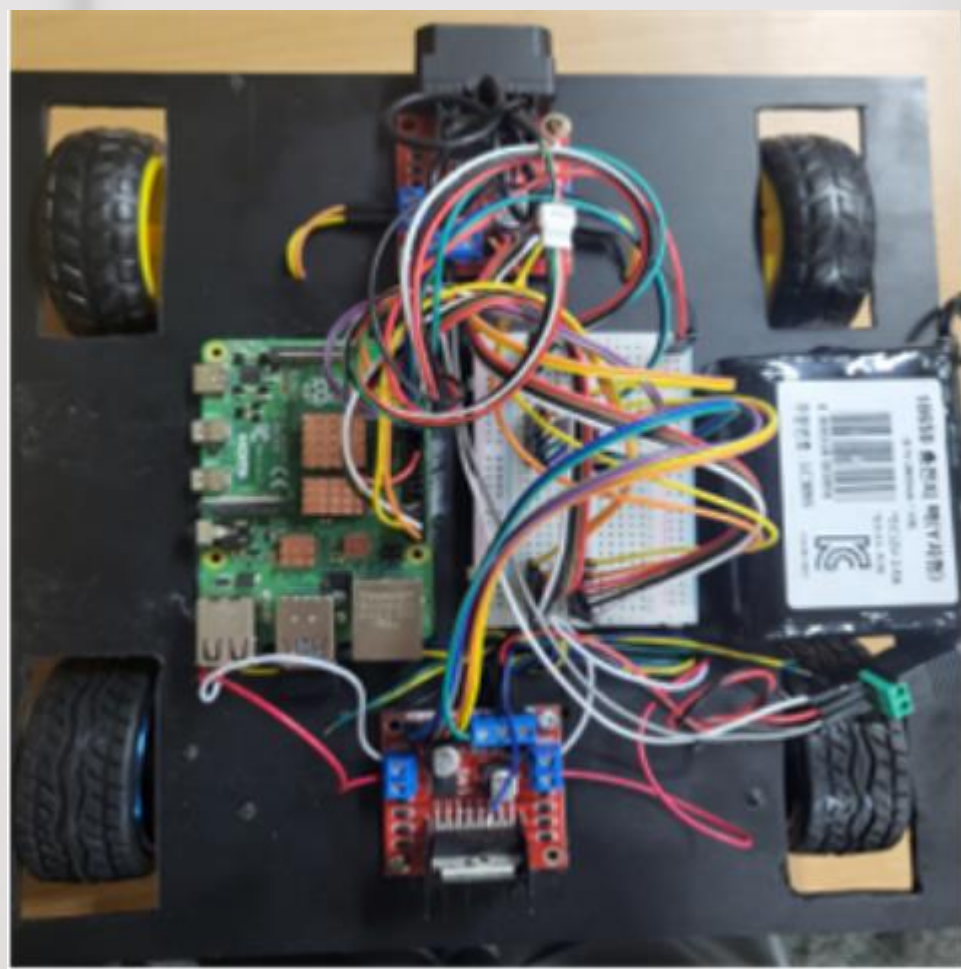
프로젝트 수행과정



- 기어드DC모터 타이어세트 [SZH-GNP194] 2개 5,000₩
- 기어박스장착모터 (NP01D-288) 2개 3,600₩
- 66파이 WHEEL [SZH-GNP105-1] 4개 7,400₩
- A L298 모터드라이버 모듈 (아두이노 호환) [SZH-EK001] 2개 4,000₩
- TFmini Plus LiDAR 거리 측정 센서 IP65 (10cm-12M) 1개 69,000₩
- [LC3095] Coms 18650 충전지 팩(Y자형), DC12V/2.6Ah
3.7V/2600mA*3ea 1개 27,100₩
- 18650 건전지 충전전용 DC 아답터 (DC12.6V/2A) [IB703] 1개 18,100₩
- 터미널블럭 to DC 커넥터 (DC2.1 FEMALE) [SZH-LC002] 1개 500₩
- 테스트[CH254] 소켓 점퍼 케이블 40P (칼라) (M/M) 20cm 850₩
- SYMCODE 바코드 스캐너 New S950 + 거치대 38,500₩
- 라즈베리파이 4B 2GB/32GB 스타터키트 176,000₩
- 포맥스 25,500₩

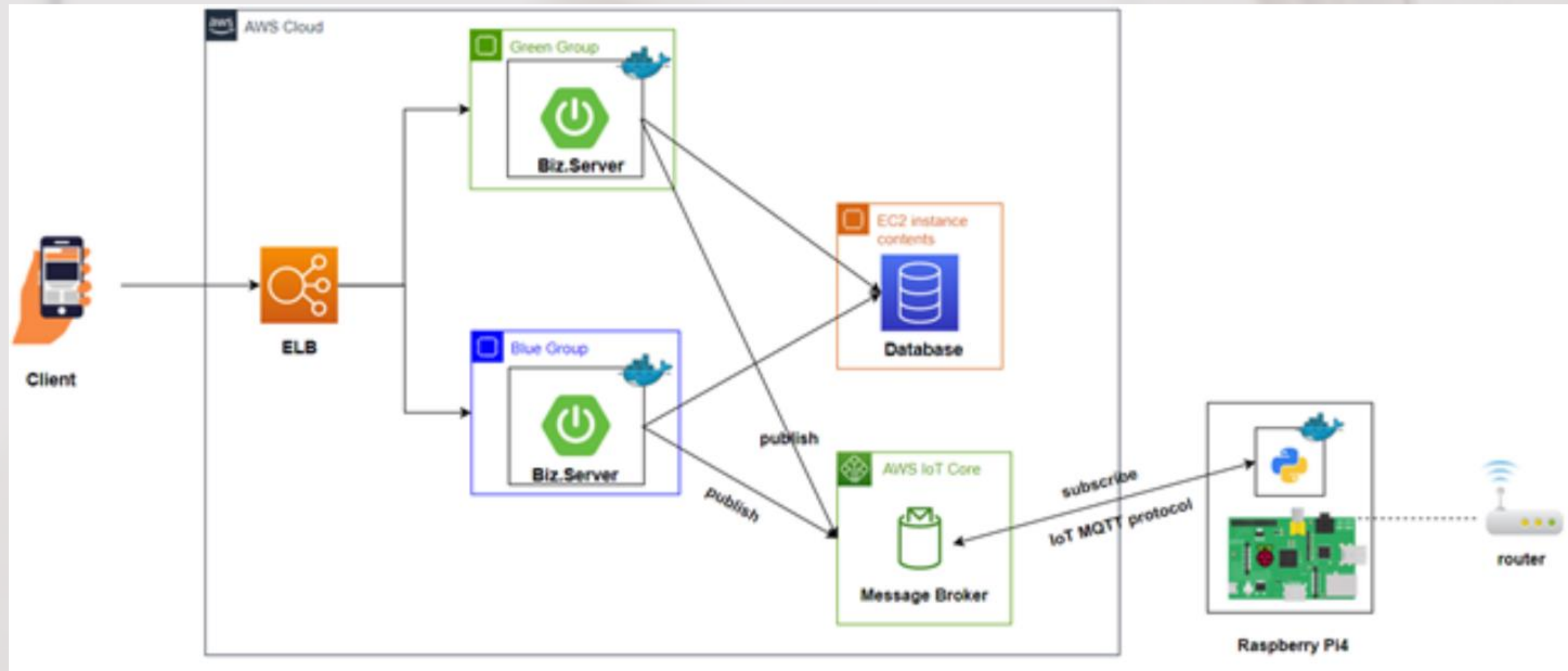
총 금액: 375,550₩

프로젝트 수행 과정



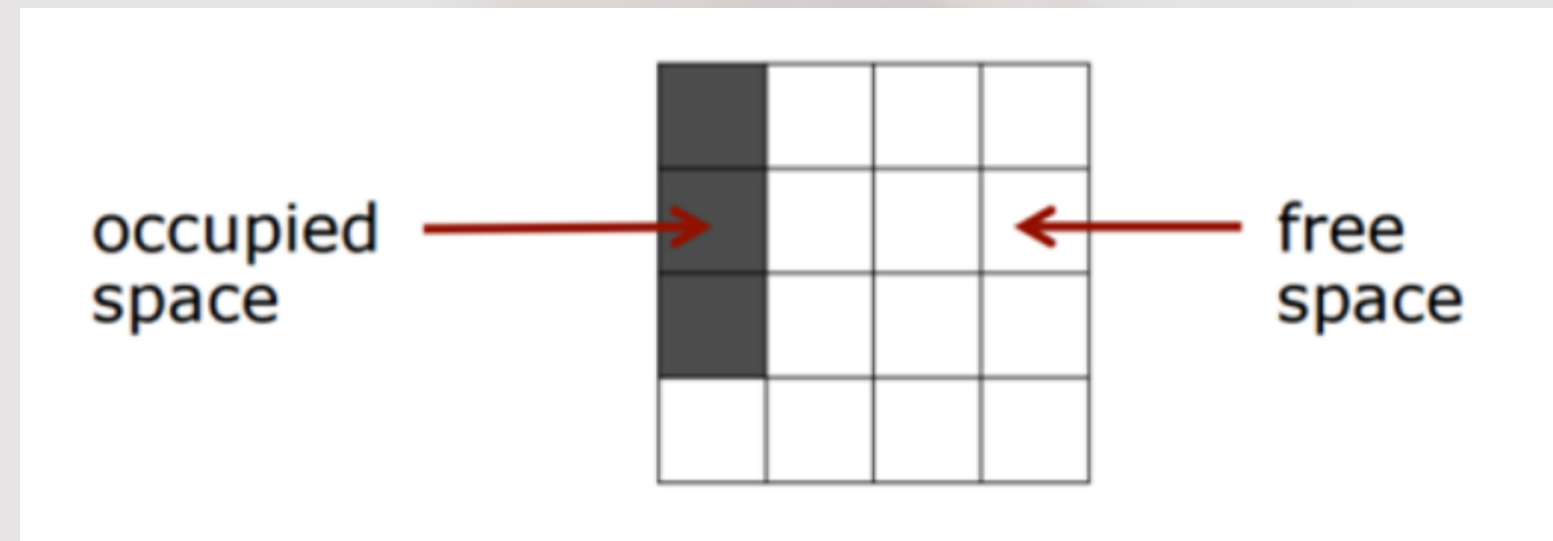
프로젝트 수행 과정

시스템 아키텍처



프로젝트 수행 과정

자율주행 알고리즘 - Occupancy Grid Mapping



Occupancy Grid Mapping은 로봇의 주변 환경을 그리드로 분할하고, 각 셀을 장애물 여부에 따라 1(occupied space) 또는 0(free space)으로 표시하여 맵을 구축하는 알고리즘입니다.

프로젝트 수행 과정

자율주행 알고리즘 - Occupancy Grid Mapping

- Occupancy Grid Mapping 소스코드

```
import numpy as np

def occupancy_grid_mapping(laser_ranges, robot_pose, grid_map):

    map_width, map_height = grid_map.shape
    resolution = 0.1

    # 로봇 위치 및 방향
    robot_x, robot_y, robot_theta = robot_pose

    max_range = 30.0

    # 레이저 스캔 각도 범위
    angle_min = -np.pi / 2
    angle_max = np.pi / 2

    # 셀 단위 변화량 계산
    dx = resolution * np.cos(robot_theta)
    dy = resolution * np.sin(robot_theta)

    # 로봇의 현재 위치를 맵 좌표로 변환
    robot_map_x = int(robot_x / resolution)
    robot_map_y = int(robot_y / resolution)

    # 맵을 순회하며 레이저 스캔 결과 반영
    for i in range(len(laser_ranges)):
        range_i = laser_ranges[i]
        angle_i = angle_min + i * (angle_max - angle_min) / len(laser_ranges)

        obstacle_x = robot_x + range_i * np.cos(robot_theta + angle_i)
        obstacle_y = robot_y + range_i * np.sin(robot_theta + angle_i)

        obstacle_map_x = int(obstacle_x / resolution)
```

```
        obstacle_map_y = int(obstacle_y / resolution)

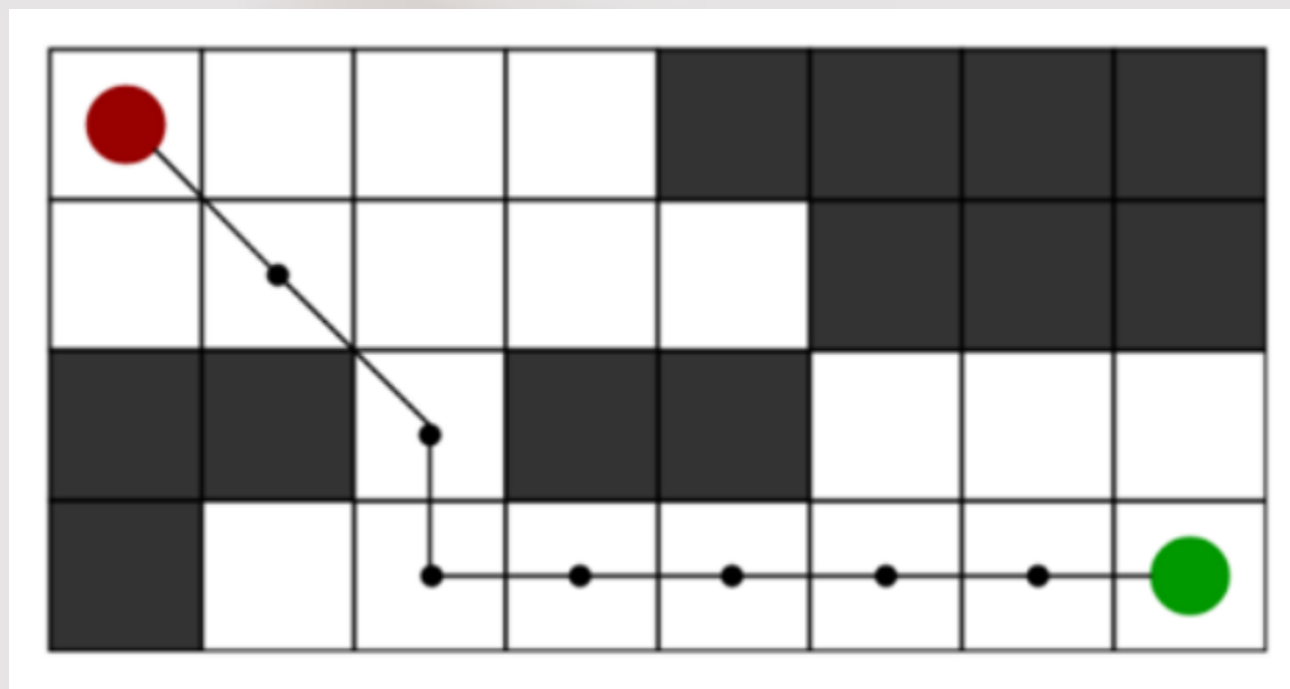
        if obstacle_map_x < 0 or obstacle_map_x >= map_width or \
            obstacle_map_y < 0 or obstacle_map_y >= map_height:
            continue

        if range_i < max_range:
            grid_map[robot_map_x:obstacle_map_x, robot_map_y:obstacle_map_y] += 1
        else:
            grid_map[obstacle_map_x, obstacle_map_y] += 1

    return grid_map
```


프로젝트 수행 과정

자율주행 알고리즘 - A*



A* 알고리즘은 시작 노드로부터 목표 노드까지 가는 경로를 찾는데 사용되는 알고리즘으로 탐색 공간을 효율적으로 탐색하면서 최적 경로를 보장해주는 알고리즘입니다.

프로젝트 수행 과정

자율주행 알고리즘 - A*

- A* 소스코드

```
import numpy as np
import heapq

def heuristic_cost_estimate(start, goal):
    return np.linalg.norm(goal - start)

def reconstruct_path(came_from, current):
    total_path = [current]
    while current in came_from:
        current = came_from[current]
        total_path.append(current)
    return total_path[::-1]

def a_star(start, goal, grid_map):
    map_width, map_height = grid_map.shape

    open_list = []
    closed_list = set()
    heapq.heappush(open_list, (0, start))

    g_score = {start: 0}
    h_score = {start: heuristic_cost_estimate(start, goal)}

    f_score = {start: h_score[start]}

    came_from = {}

    while open_list:
        current = heapq.heappop(open_list)[1]

        if current == goal:
```

```
            return reconstruct_path(came_from, current)

        closed_list.add(current)

        neighbors = []
        x, y = current
        if x > 0:
            neighbors.append((x - 1, y))
        if x < map_width - 1:
            neighbors.append((x + 1, y))
        if y > 0:
            neighbors.append((x, y - 1))
        if y < map_height - 1:
            neighbors.append((x, y + 1))

        for neighbor in neighbors:
            tentative_g_score = g_score[current] + 1

            if grid_map[neighbor] > 0:
                continue

            if neighbor in closed_list:
                continue

            if neighbor not in g_score or tentative_g_score < g_score[neighbor]:
                came_from[neighbor] = current
                g_score[neighbor] = tentative_g_score
                h_score[neighbor] = heuristic_cost_estimate(neighbor, goal)
                f_score[neighbor] = g_score[neighbor] + h_score[neighbor]
                heapq.heappush(open_list, (f_score[neighbor], neighbor))

    return None
```


프로젝트 성과물

회원가입



A mobile app interface for library membership registration. At the top, the status bar shows 'SKT 1:54' and '60%' battery. The title '도서관 회원가입' (Library Membership Registration) is centered. Below it are three input fields: 'ID 입력' (ID Input), 'pw 입력' (Password Input), and '이름 입력' (Name Input). At the bottom is a blue button labeled '회원가입' (Sign Up).

구성

- ID 입력 텍스트 상자
- PW 입력 텍스트 상자
- 이름 입력 텍스트 상자
- 회원가입 버튼

유효성 검사

공백이 포함되거나 길이의 범위가 4~12를 벗어나는 경우 회원가입을 할 수 없습니다. 회원가입 성공 실패 여부를 화면 상단에 알림으로 띄워 사용자에게 알려주고 성공시 로그인 창으로 넘어갑니다.

프로젝트 성과물

회원가입 (유효성 검사)

SKT 2:05 57%

도서관 회원가입

ID 입력

공백을 채우세요

pw 입력

공백을 채우세요

이름 입력

공백을 채우세요

회원가입

SKT 2:05 57%

도서관 회원가입

097년

영문과 숫자로 입력하세요

..

PW 길이가 너무 짧습니다

영문숫자길이

회원가입

프로젝트 성과물

회원가입 성공/실패

SKT 1:55 60%

회원가입 성공
로그인 해주세요

도서관 로그인

ID 입력

pw 입력

로그인

회원가입

SKT 11:35 83%

회원가입 실패
이미 가입된 ID입니다

도서관 회원가입

potato

감자자

회원가입

프로젝트 성과물

로그인

A mobile app login screen for a library. At the top, the status bar shows 'SKT 1:58' and '59%'. The main title '도서관 로그인' (Library Login) is centered. Below it are two text input fields: the first contains 'potato' and the second contains masked characters '.....'. A blue '로그인' (Login) button is positioned below the password field, and a smaller blue '회원가입' (Sign Up) link is at the bottom.

SKT 1:58 59%

도서관 로그인

potato

.....

로그인

회원가입

구성

- ID 입력 텍스트 상자
- PW 입력 텍스트 상자
- 로그인 버튼
- 회원가입 버튼

유효성 검사

공백이 포함되거나 길이의 범위가 4~12를 벗어나는 경우 회원가입을 할 수 없습니다. 회원가입 성공 실패 여부를 화면 상단에 알림으로 띄워 사용자에게 알려주고 성공시 로그인 창으로 넘어갑니다.

프로젝트 성과물

로그인(유효성 검사)

SKT 2:04 58%

도서관 로그인

h wiwo

영문과 숫자로 입력하세요

로그인

회원가입

SKT 3:20 37%

도서관 로그인

ID 입력

공백을 채우세요

pw 입력

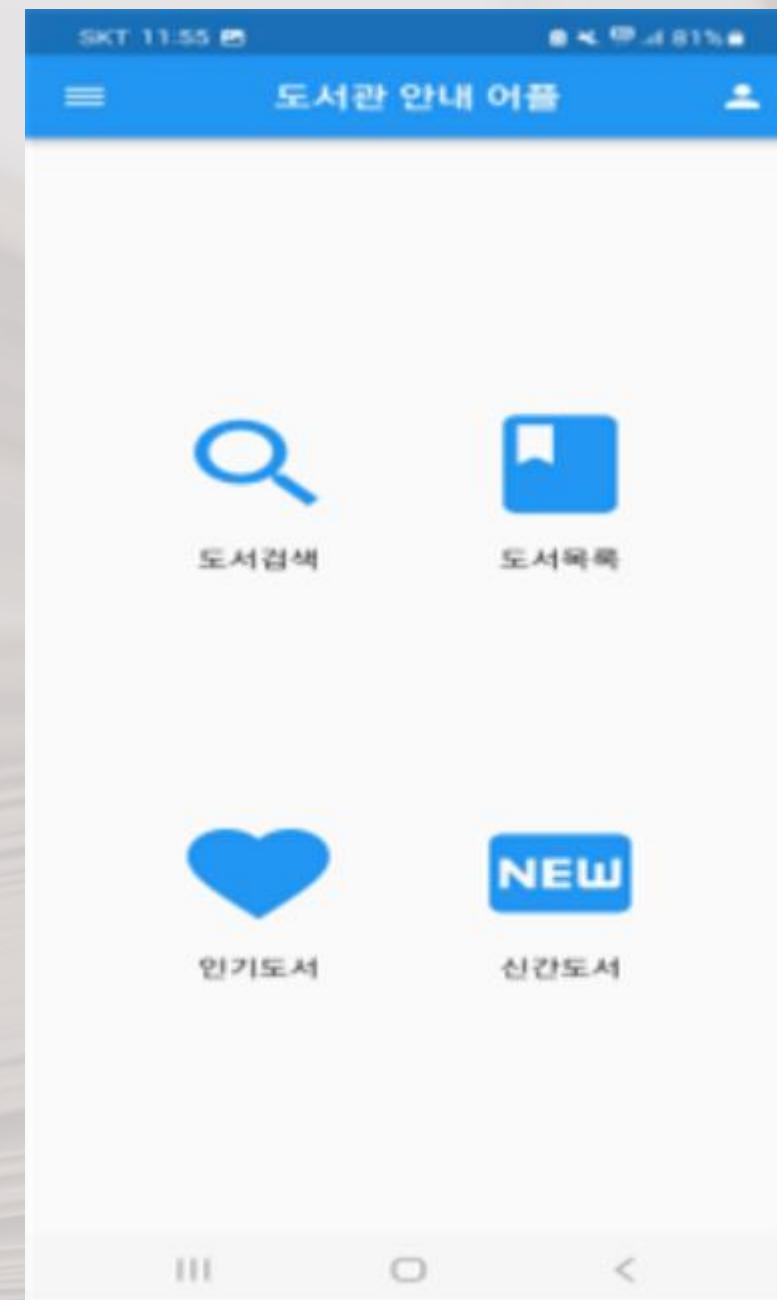
공백을 채우세요

로그인

회원가입

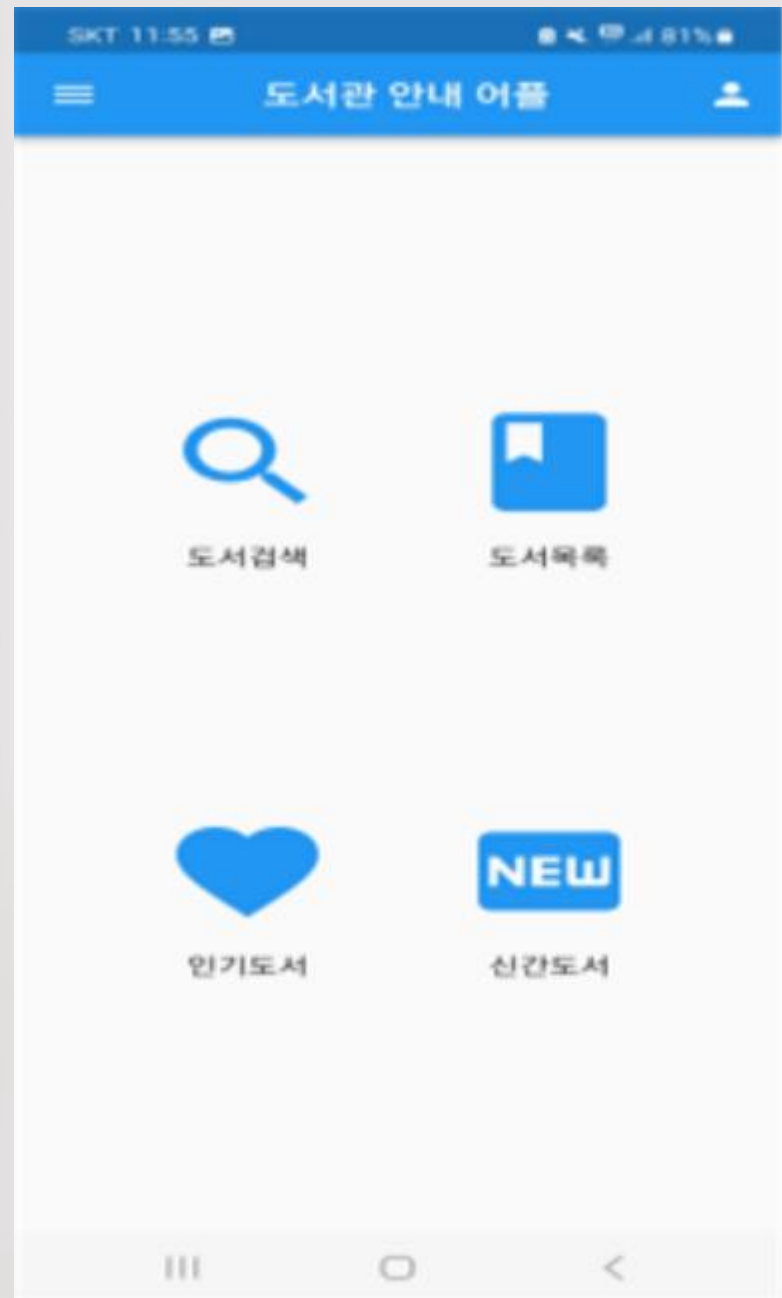
프로젝트 성과물

로그인 성공/실패



프로젝트 성과물

메인 화면



구성

- 메뉴버튼
- 사용자 정보 버튼
- 도서 검색 버튼
- 도서 목록 버튼
- 인기 도서 버튼 (미구현)
- 신간 도서 버튼 (미구현)

프로젝트 성과물

도서 목록



구성

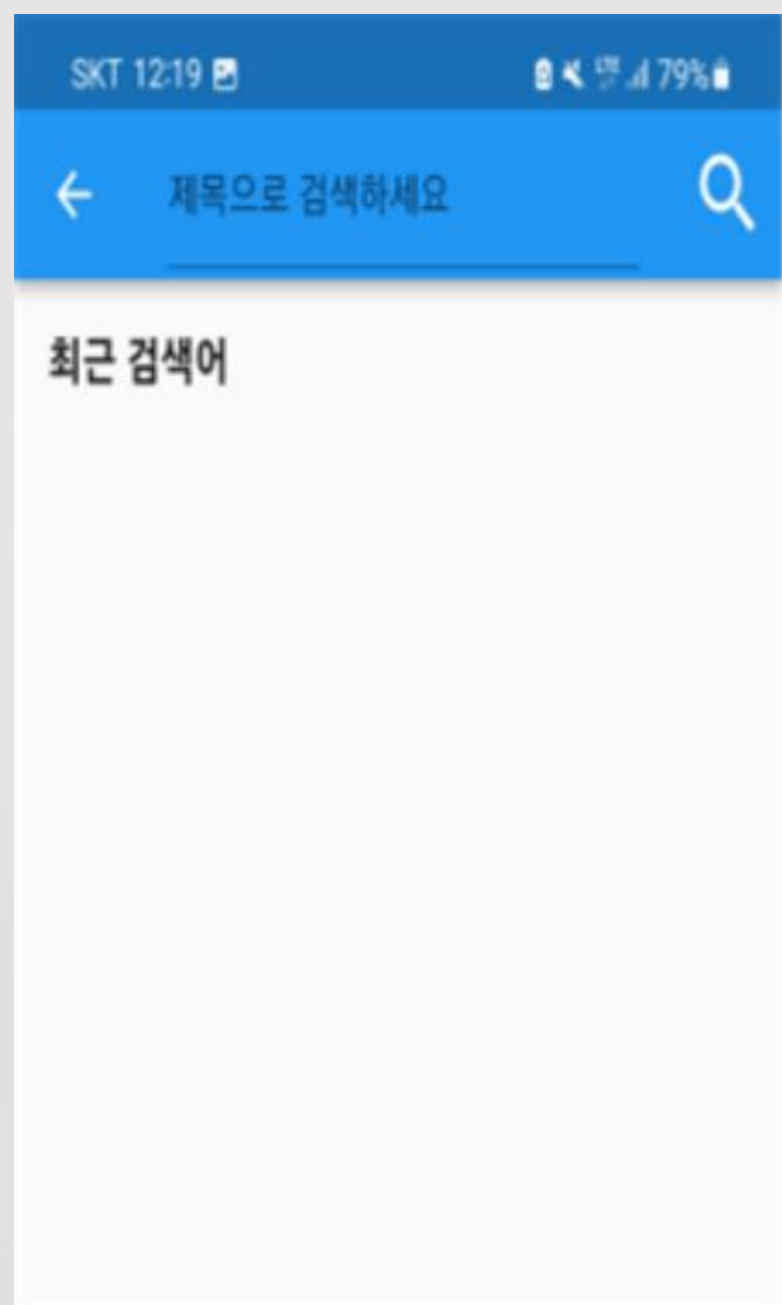
- 현재 보유 하고 있는 도서 리스트
- 도서 이미지/도서 제목/ 글쓴이

기능

- 도서관에서 보유한 책을 상하 스크롤 형태로 띄웁니다.
- 전체 도서를 한번에 불러오지않고 스크롤을 할 때마다 10개씩 도서를 불러옵니다.
- 책을 클릭하면 도서 정보를 띄웁니다.

프로젝트 성과물

도서 검색



구성

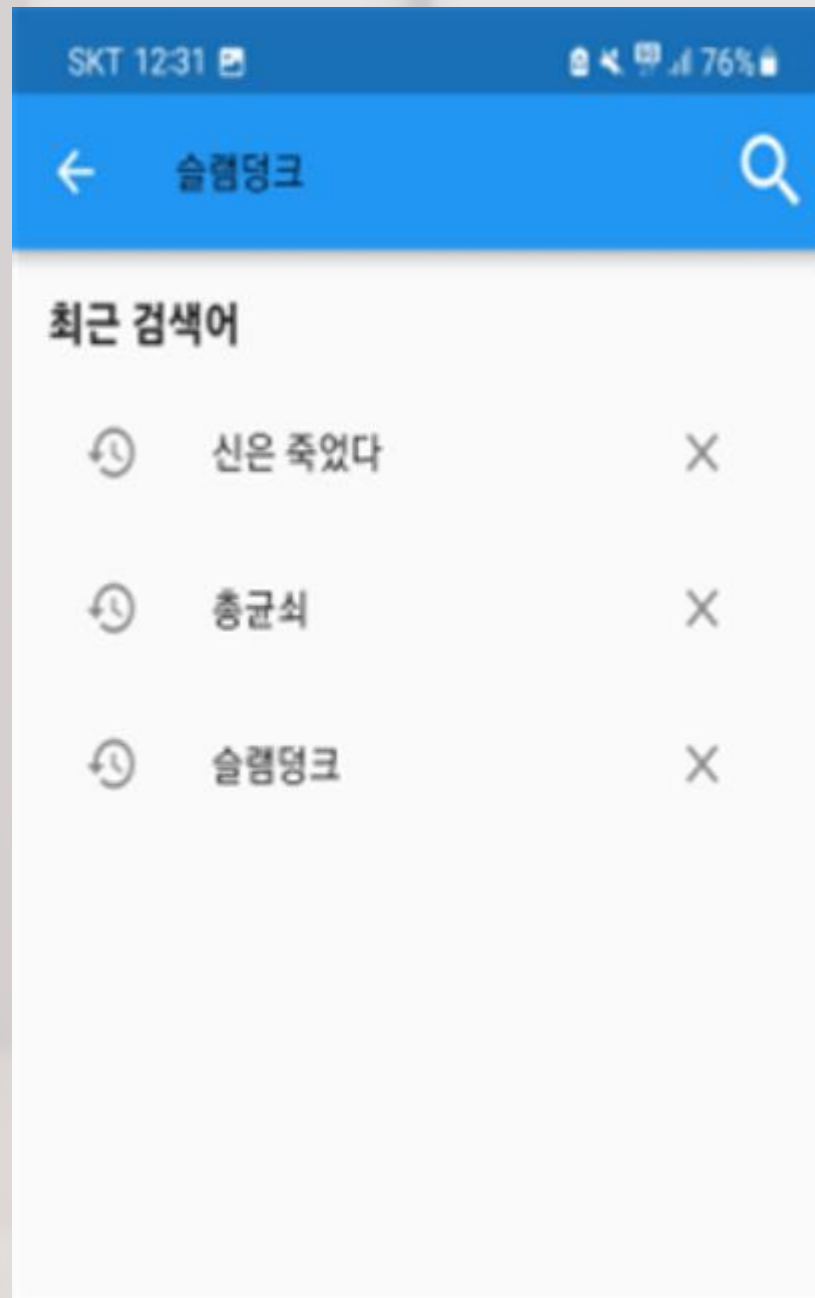
- 뒤로가기 버튼
- 검색 상자
- 검색 버튼
- 최근 검색어

기능

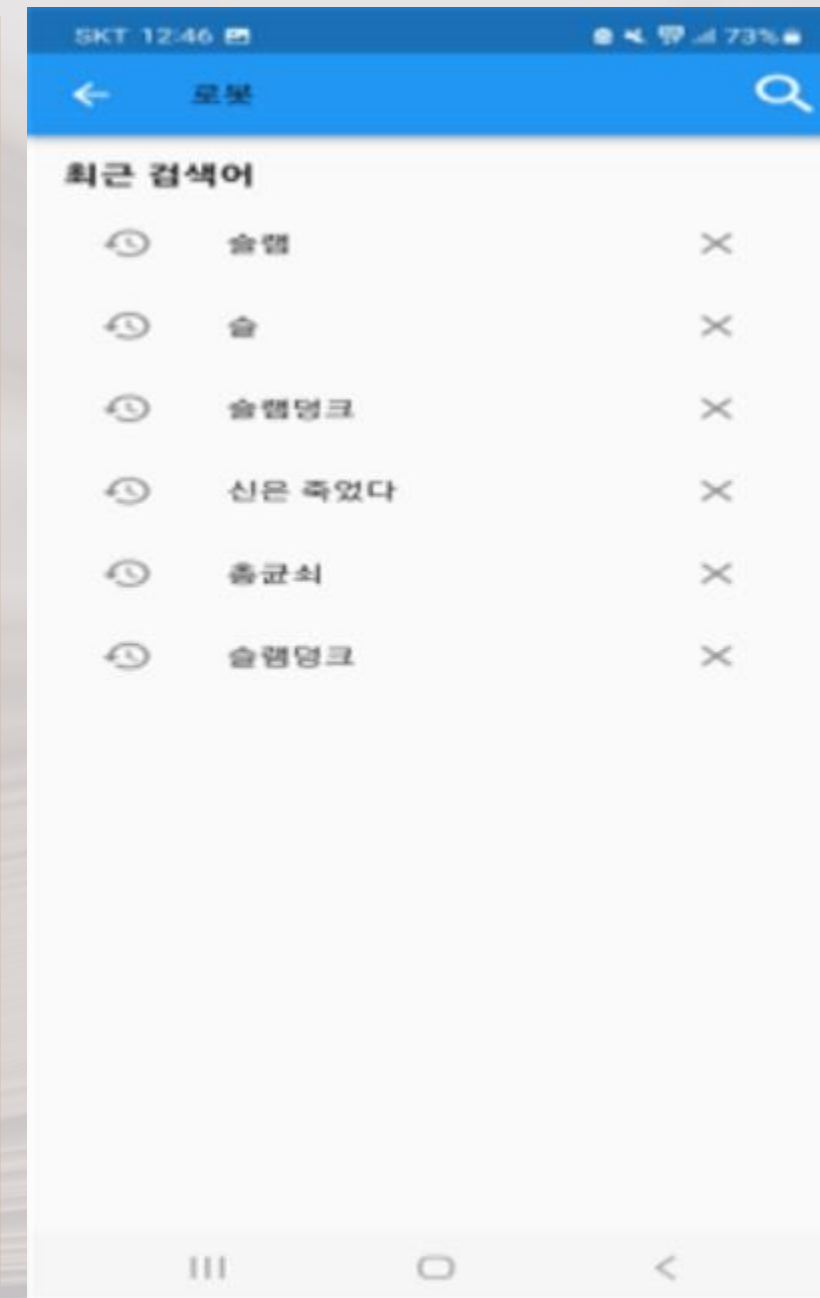
- 최근 검색한 도서 기록이 남습니다.
- 제목으로 도서를 검색할 수 있고 검색버튼을 누르면 제목, 저자, 출판사, 과학기술, 문화예술, 인문사회를 선택해서 상세한 검색을 할 수 있습니다.

프로젝트 성과물

도서 검색

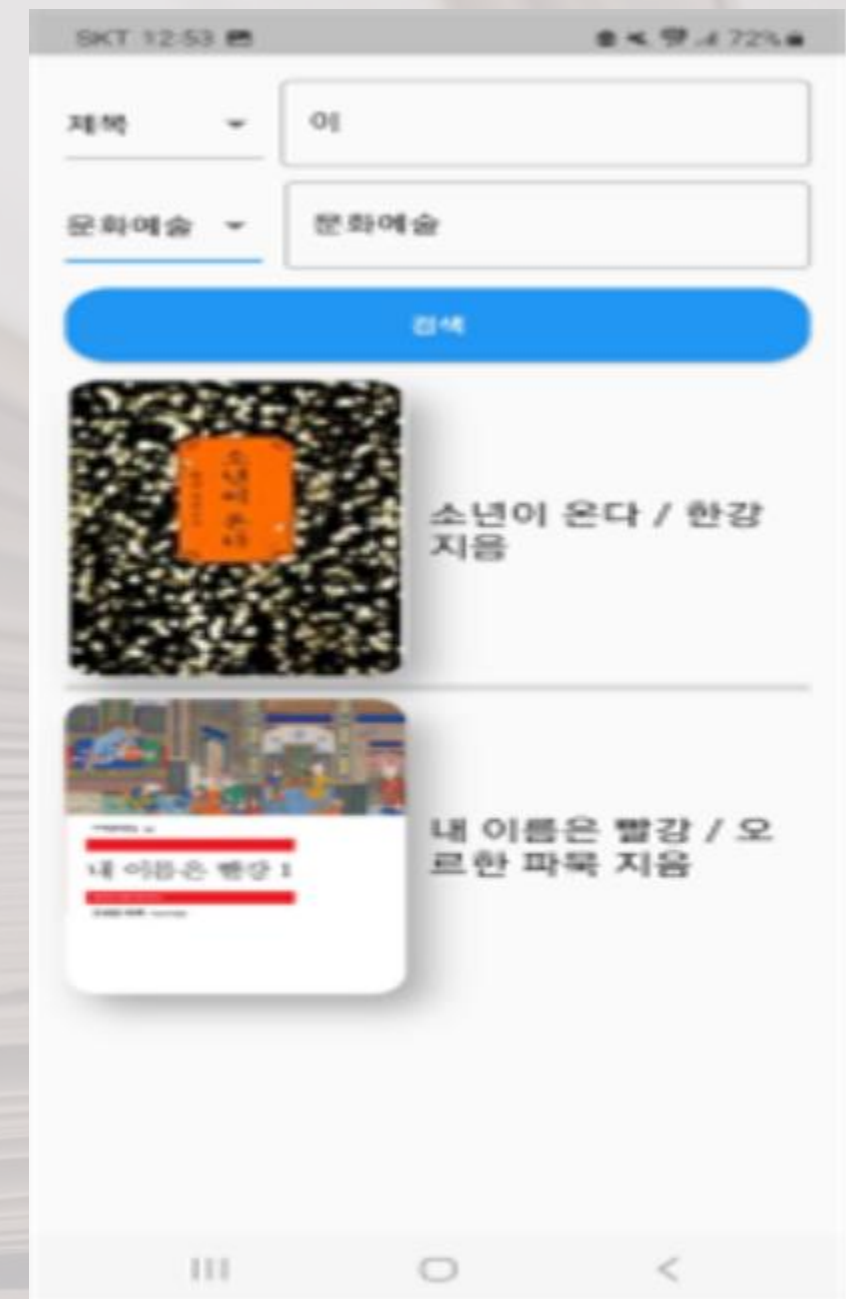


단어로 도서 검색



프로젝트 성과물

도서 조합 검색



프로젝트 성과물

도서 정보



구성

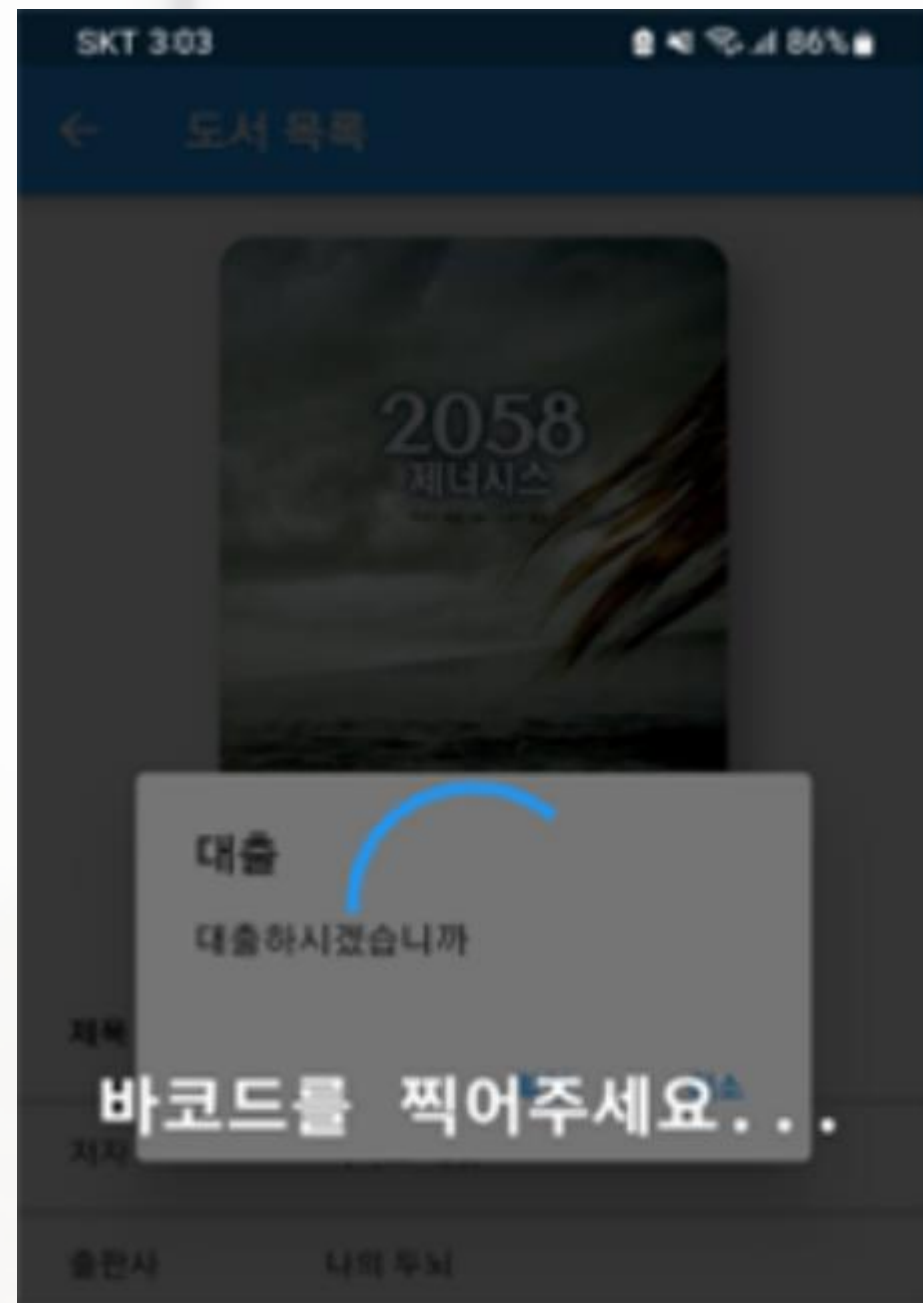
- 제목
- 저자
- 출판사
- 발행연도
- 카테고리
- ISBN
- 대출 버튼
- 안내 버튼

기능

- 대출 버튼을 통해 도서를 대출 할 수 있습니다.
- 안내 버튼을 통해 도서의 위치까지 안내 받을 수 있습니다.

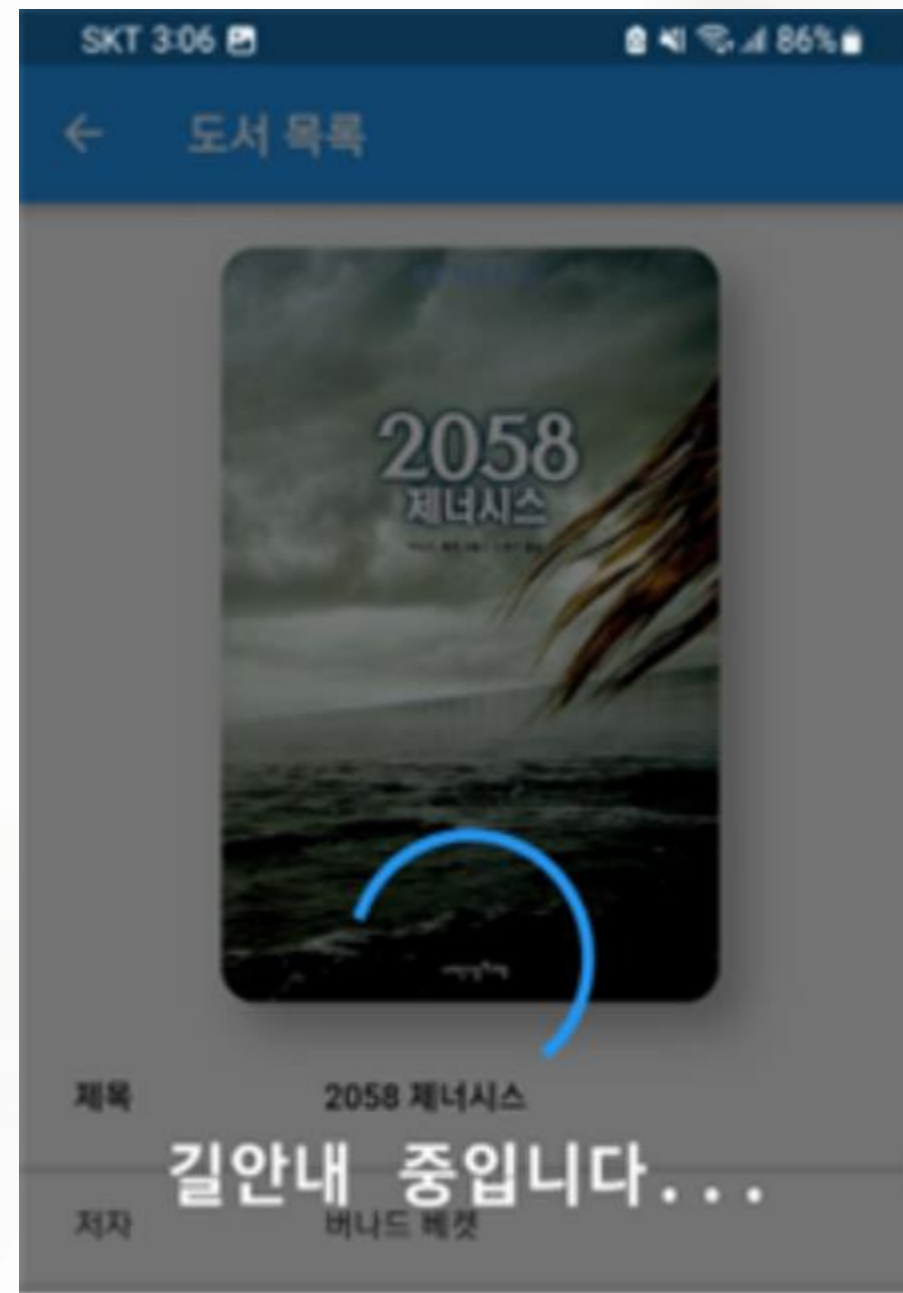
프로젝트 성과물

도서 대출



프로젝트 성과물

안내



프로젝트 성과물

회원 정보



구성

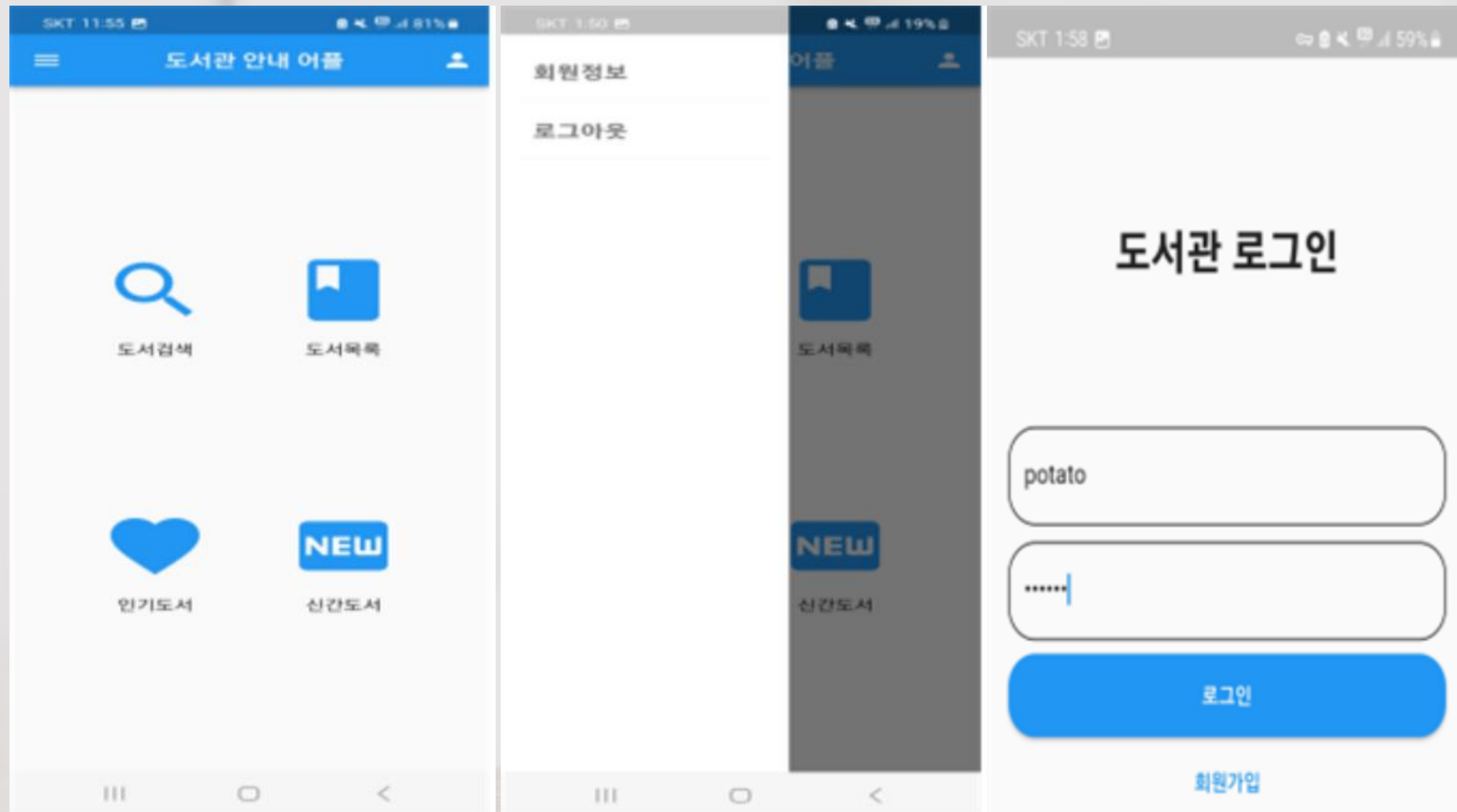
- 뒤로 가기 버튼
- 회원명
- 반납 버튼
- 도서 대출 목록
 - 도서 제목/대출일/반납일

기능

반납 버튼을 통해 도서를 반납할 수 있습니다.

프로젝트 성과물

로그아웃



구성

- 로그아웃 버튼

프로젝트 결론

프로젝트 달성도

목표	중요도(%)	달성도(%)	수행내용
API 설계 및 구현	20	100%	도서 관련 API 작업
앱 UI/UX 구현	20	100%	도서 서비스 UI/UX 작업
하드웨어 구현	30	80%	자율주행 서비스를 위한 HW 구현
자율주행 알고리즘 구현	30	60%	최단 경로, 맵 생성 알고리즘 구현
합계	100	82%	

프로젝트 결론

프로젝트 결과 논의

- Spring Boot를 이용한 API 설계 및 구현, Flutter를 이용한 앱 UI/UX 구현 등 소프트웨어 측면에서의 목표를 대부분 달성
- 회전 모터가 따로 존재하지 않아 방향전환 각도가 항상 달라지는 문제 개선 필요
- 바닥 재질에따라 로봇이 제대로 작동하지 않는 문제가 발생해 개선 필요
- 배터리양에 따라 모터의 속도가 변하는 문제 개선 필요
- 자율주행 알고리즘 중 맵 생성하는 부분이 잘 안되 현재는 수동으로 맵 생성하는 문제 개선 필요
- NAT 환경의 라즈베리파이와 AWS EC2 WAS와 통신이 지속적으로 끊기는 문제 개선 필요

프로젝트 시연



Autonomous Library Service Robot

16조 발표를 마치겠습니다!
