

현대모비스SW아카데미 익스텐십 프로젝트

주차관리 시스템 제작 설계서

주차관리해조

이광진

고석환

송대근

2022.02.28

목차

1. 개요

가. 주제 및 선정 이유

나. 구조

(1) 개발 환경 구축

2. 수행 내용

3. 프로젝트 스케줄

1. 개요

현대모비스 SW 아카데미 프로젝트의 일환으로, C 언어의 활용과 일정 조율 및 분배로 팀워크를 향상할 수 있는 방법을 모색한다.

(가) 주제 및 선정 이유

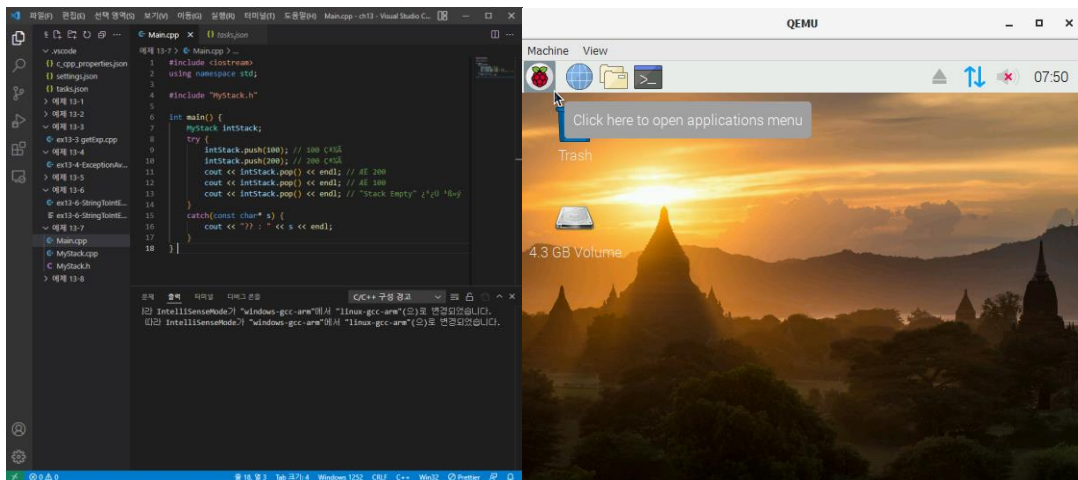
주제 : 주차관리 시스템

백화점이나 마트 등 흔히 마주칠 수 있는 주차장 관리 시스템을 손으로 직접 구현해보고, 더 나은 알고리즘을 생각하여 좀 더 향상할 수 있는 방안을 모색해봤다.

(나) 구조

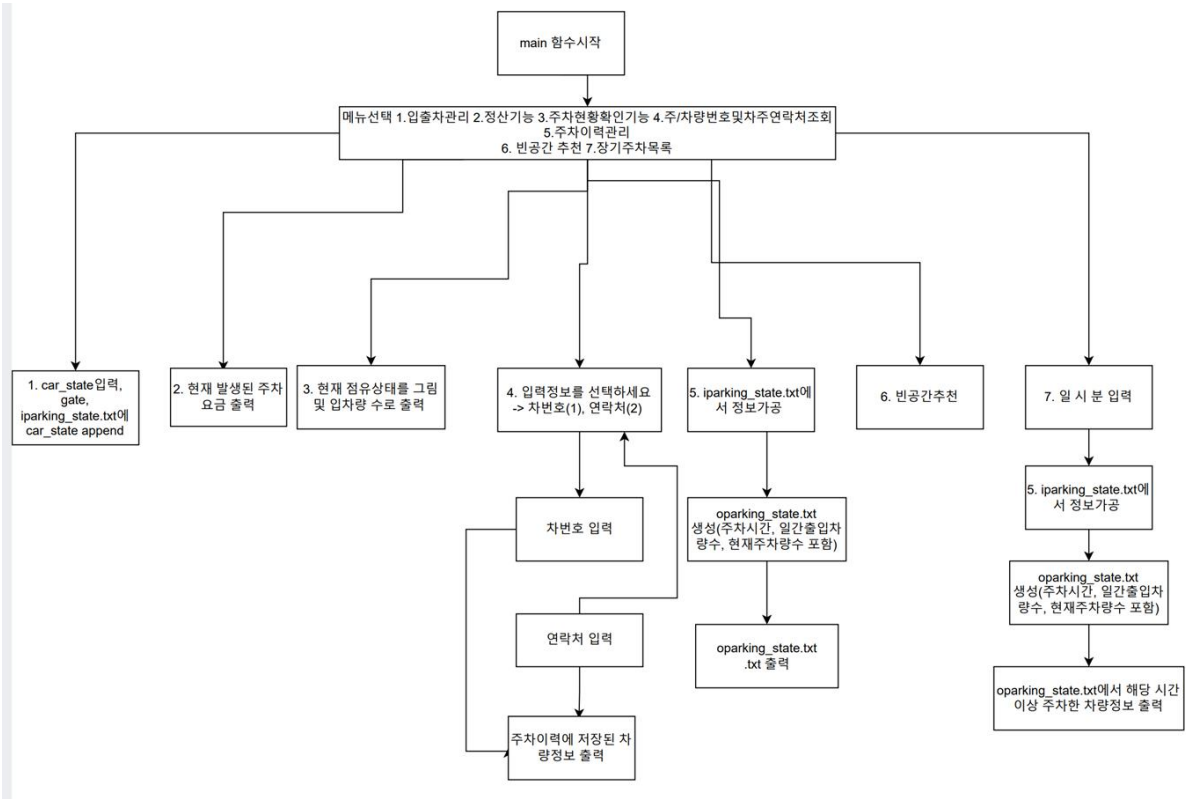
나-1) 개발 환경 구축

라즈베리파이 가상 OS 기반의 CLI 를 기반으로 한다. 윈도우 상으로 VSC 로 크로스 컴파일하여 디버깅이 용이하고 코드 수정이 더 원활하게 수행하고자 한다.

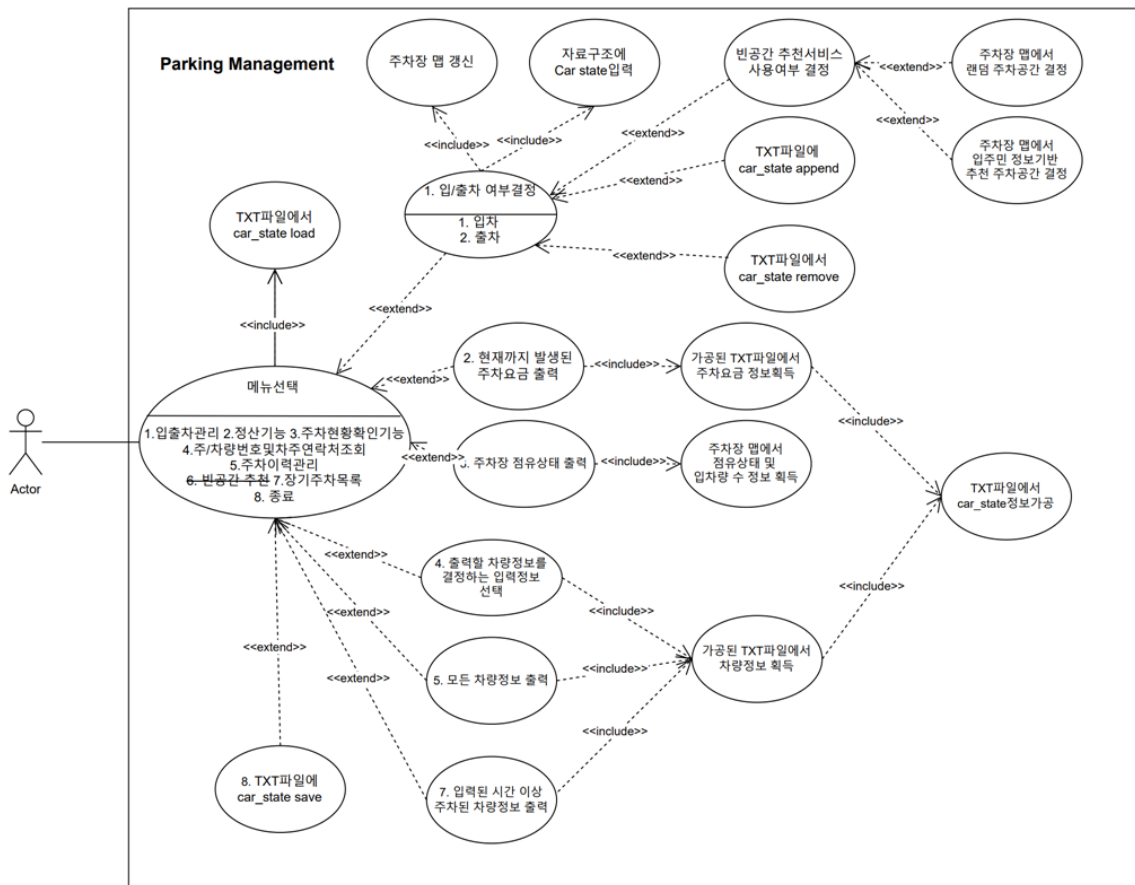


(좌측, 윈도우 상에서의 VSC / 우측, 라즈베리기반의 가상 OS)

2. 수행내용



(초기 프로젝트 구조 설계)



(구현하고자하는 시나리오의 간단한 모델 설계도)

시나리오 구성

입차 혹은 출차 이벤트가 발생할 때 사용하는 프로그램을 설계하고자 한다.

현재 주차장에서 흔히 볼 수 있는 주차장 프로그램은 별도의 센서를 통해 입, 출차가 자동화되었지만, 본 시나리오에서 사용자가 임의로 입력하여 시나리오를 진행한다고 가정하겠다.

유저, 관리자 모두 해당 프로그램에 접속할 수 있다.

하지만 연락처 조회나 장기 주차 목록 등 개인정보에 관련되고, 혹여나 범죄에 이용될 수 있다는 점을 생각하여 관리자와 유저가 볼 수 있는 목록을 제한했다.

초기 메뉴 접근 권한은 크게 2 가지로 나뉜다.

1. 관리자 모드(Admin) <- 추가기능 구현
2. 일반 유저 모드(User)

초기 프로그램에 진입 시 유저 모드와 관리자 모드 중 선택하라는 선택지가 등장한다. 이때 접근 권한에 따라 보이는 메뉴가 달라진다.

1. 입, 출차 관리 (User, Admin)
2. 정산 기능(User)
3. 주차현황 확인 기능(User)
4. 주/차량 번호 및 차주 연락처 조회(Admin)
5. 주차 이력 관리 (Admin)
6. 장기주차목록 (Admin)
7. 종료 (User, Admin)

1.입, 출차 관리(int car_state_append, int car_state_remove, int enter_exit_time)

입차 내역은 iparking_state.txt 에 저장된다. 해당 데이터는 입차 시각을 기준으로 인덱스가 순차적으로 증가하며, 어레이에 저장된다.

어레이는 순차 자료 구조이므로, 입차 기준으로 정렬을 할 때 필요한 데이터가 인덱스이다. 시각적으로 보기 쉽고 정렬 시 가장 알맞은 데이터가 인덱스이기 때문에 자료구조 중 하나인 어레이를 선택했다.

차가 출차 시 데이터가 추가, 가공되어 oparking_state.txt 에 저장된다. 방식은 출차 시각 기준으로 정렬되며, 해당 파일에는 차종 색깔, 차 번호, 차량번호, 입차 시각, 출차 시각, 정산금액 등이 적힌다.

입차 시 현재 시각을 업데이트하는 함수는 **int enter_exit_time** 로, 변수에 현재시각을 업데이트해준다.

int car_state_append 함수로 iparking_state.txt 에 현재 차량에 대한 정보를 업데이트한다. 추후 차량이 출차될 시, **int car_state_remove** 로 iparking_state.txt 에 있는 해당 차량의 정보를 삭제한다.

2. 정산기능(int calculate_fee(car_state state))

해당 차량이 출차할 때, 혹은 현재 정산금액을 알고 싶을 때 iparking_state.txt 에 있는 입차 시각을 불러와 현재 시각과 비교하여 해당 금액을 출력한다.

정산 시 계산되는 알고리즘은 시간에 따른 차등 방식으로 계산한다.

해당 함수는 출차 시 oparking_state.txt 에 적히는 정산금액에도 사용될 수 있다.

3. 주차현황 확인 기능(int print_my_state)

현재 주차장 점유 상태를 그림과 차량 수로 출력하는 함수이다.

root >

2층

-1입력시 메뉴 나가기

		26바 3434				
				34하 1234		

1층으로 이동(down)

3층으로 이동(up)

>

해당과 같이 UI를 구성했으며 보이는 것과 같이 주차공간 및 주차한 차량의 정보를 알 수 있다.

명령어 창에 down, up을 입력하면 위층과 아래층을 자유롭게 이동하며 주차장을 확인할 수 있다.

입차 시 입력되는 iparking_state.txt에 있는 차량 정보와 주차할 위치를 입력 받아 함수로 보내주고, 함수에서 주차할 위치에 차량을 주차한다.

3 X 3 X 10의 배열로 구성한다.

4. 주/차량 번호 및 차주 연락처 조회(int select_input_info, int vehicle_info_out(car_state state))

해당 함수는 관리자가 이용할 수 있는 메뉴로, 합리적인 상황에 의해 경찰과 같은 공무집행이나 주차장 내 특정 사건이 발생하여 차량 운전자에 대한 정보가 필요할 경우 요청으로 차량에 대한 정보를 검색할 수 있다.

lparking_state.txt 에 있는 모든 정보를 불러와 해당 내용을 출력한다.

5. 주차 이력 관리(int parkHistory)

20.01.01 00:00 ~ 20.09.31 20:00						
순번	차량번호	입차일시 /	비고	출차일시	주차시간	주차
3	3006	2020.01.22 12:56	시간	2020.09.18 21:24	240 08 28	7,307
4	5008	2020.01.24 15:58	시간	2020.03.03 17:16	039 01 18	2,811
5	5009	2020.01.24 15:59	시간	2020.09.18 21:24	238 05 25	7,154
6	01머6200125	2020.01.24 15:59	시간	2020.09.18 21:24	238 05 25	7,154
7	03다1234125	2020.01.24 15:59	시간	2020.09.18 21:24	238 05 25	7,154
8	45다12341n49	2020.01.24 15:59	시간	2020.09.27 23:23	247 07 24	7,793
9	34너12321n4	2020.01.30 11:43	시간	2020.04.09 09:51	069 22 08	5,034
10	9846	2020.04.09 09:52	시간	2020.09.27 23:24	171 13 32	2,354
11	25가15840	2020.05.26 09:32	시간	2020.10.16 18:28	143 08 56	1,440
12	9999	2020.09.14 20:04	시간	2020.09.27 23:15	013 03 11	947
13	5544	2020.09.15 22:30	시간	2020.09.15 22:30	000 00 00	2

oparking_state.txt 에 있는 내용과 lparking_state.txt 를 가공 처리 후 인덱스를 기준으로 정렬하여 화면에 출력한다.

6. 장기 주차 목록

lparking_state.txt 에서 지정한 기간보다 초과한 주차한 차량을 보여준다. lparking_state.txt 에 있는 내용을 불러오고 입차 시간과 현재 시간을 비교하여 지정한 기간 보다 넘으면 해당 차에 대한 정보를 모두 출력한다.

3. 프로젝트 스케줄



4. 참고

Notion link : <https://www.notion.so/ccbcbccf6f9041ca94c001b7dea7474e>