**주차장 관리 시스템**

**소프트웨어 공학 개발 과제**

**프로젝트 보고서**



**한국외국어대학교**

201303411 천수홍

201403152 조병건

201400948 김효성

201402412 이승준

201400582 김영민

**문서 개정이력**

|  |  |  |  |
| --- | --- | --- | --- |
| **순번** | **개정내용** | **작성자** | **제.개정일자** |
| 1 | 신규작성 | 김효성 외 4명 | 2018-10-30 |
| 2 | 사용자 친화적인 문서로 개정 | 조병건, 김영민 | 2018-11-01 |
| 3 | 유스케이스 내용 추가 | 김효성 외 4명 | 2018-11-04 |
| 4 | 사용자 친화적인 문서로 개정 | 김효성 | 2018-11-07 |
| 5 | DFD 내용 추가 및 유스케이스 변경 | 조병건, 김영민 | 2018-11-27 |
| 6 | 유스케이스 다이어그램 수정 | 조병건 | 2018-12-26 |
| 7 | 유스케이스 표화 | 김영민 | 2018-12-26 |
| 8 | 비기능적 요구사항 수정 | 김효성 | 2018-12-26 |
| 9 | 프로토콜 설계서 추가 | 김영민, 김효성, 이승준 | 2018-12-28 |
| 10 | 테스트 항목 추가 | 조병건 | 2018-12-28 |
| 11 | 목차 수정 | 이승준 | 2018-12-28 |
| 12 | 최종업무분담, 작업변경내역 항목 추가 | 조병건 | 2018-12-28 |
| 13 | 목차 수정 | 김효성 | 2018-12-28 |
| 14 | 간트차트 변경 | 조병건 | 2018-12-28 |
|  |  |  |  |
|  |  |  |  |

**목 차**

1. **개요 7**
   1. 프로그램 개요 7
   2. 실무담당자와 면담 내용 7
   3. 개발 범위 7
   4. 타당성 분석 8
      1. 경제적 타당성 분석 8
      2. 기술적 타당성 분석 8
   5. 산출물리스트 8
2. **계획 9**
   1. 조직도 9
   2. 역할 및 책임 9
   3. 인원투입계획 9
   4. 개발 방법론 10
3. **일정계획 11**
   1. WBS (Work Breakdown Structure) 11
   2. 네트워크 차트 12
   3. 간트 차트 13
4. **위험요소 관리 계획 15**
   1. 계획 및 설계 사항의 변경 15
   2. 하드웨어 통신 기술에 대한 배경지식 부족 15
   3. 아두이노 보드 개수의 한계 15
5. **개발환경 15**
   1. 개발 및 설계 도구 15
   2. 프로젝트 개발 목표 16
      1. 주차장 관리시스템 개발 목표 16
      2. 주차장 관리시스템 세부 개발 목표 16
6. **주차장 관리 시스템 기술개발 요구사항 명세 17**
   1. 기능적 요구사항 17
      1. Usecase diagram 17
   2. Usecase모델 17
      1. Actor 명세 17
      2. Usecase 개요 18
   3. Usecase 명세 19
      1. 입차 유즈케이스 19
      2. 출차 유즈케이스 22
      3. 정산 유즈케이스 24
      4. 예약 시스템 유즈케이스 26
      5. 주차 보조 유즈케이스 28
      6. 매출 관리 유즈케이스 30
      7. 주차 현재상태 확인 유즈케이스 31
      8. 예약상태 확인 유즈케이스 32
7. **DFD 33**
   1. Terminator 명세 34
   2. Data Store 명세 34
   3. Process 명세 34
8. **시퀀스 다이어그램 36**
   1. 예약제 주차 시퀀스 다이어그램 36
   2. 비 예약제 주차 시퀀스 다이어그램 36
   3. 시스템 내부 시퀀스 다이어그램 37
9. **비기능적 요구사항 38**
   1. 시스템 품질 모델 38
   2. 상세 품질 요구사항 38
      1. 신뢰성 요구사항 38
      2. 성능 요구사항 38
      3. 보안상 요구사항 39
      4. 사용성 요구사항 39
   3. 제약사항 39
10. **주차장 관리 시스템 프로토콜 설계서 40**
    1. 메인서버 프로토콜 설계서 40
    2. 예약 프로그램 프로토콜 설계서 43
    3. 정산기 프로토콜 설계서 45
    4. 주차장 공간 관리 프로토콜 설계서 47
    5. 주차타워 프로토콜 설계서 48
    6. 관리자 프로토콜 설계서 49
11. **소스 코드 52**
12. **테스트 52**
    1. 단위기능 테스트 52
       1. 단위기능 테스트 시나리오 53
       2. 단위기능 테스트 결과 57
    2. 통합 테스트 57
       1. 통합 테스트 결과 58
    3. 전체 테스트 결과 58
       1. 단위 테스트 결과 58
       2. 통합 테스트 결과 59
       3. 오류 수정 작업 59
13. **프로그램 수행 결과 59**
14. **결론 59**
    1. 작업 변경 내역 59
    2. 실패 요인 분석 59
15. **최종 역할 분담 60**
16. **기존 시스템과의 차별성 61**
17. **느낀점 61**
    1. 천수홍 61
    2. 김영민 62
    3. 김효성 62
    4. 이승준 63
    5. 조병건 63
18. **발표자료 64**
19. **활동사진 64**

**표 목 차**

<표 1> 개발 및 설계 도구 15

<표 2> 주차 관리 시스템 액터 명세 17

<표 3> 주차 관리 시스템 유즈케이스 식별자 목록 18

<표 4> UC-C01 입차 유즈케이스 19

<표 5> UC-C02 출차 유즈케이스 22

<표 6> UC-C03 정산 유즈케이스 24

<표 7> UC-C04 예약 시스템 유즈케이스 26

<표 8> UC-C02 주차 보조 유즈케이스 28

<표 9> UC-A01 매출 관리 유즈케이스 30

<표 10> UC-A02 주차 현재상태 확인 유즈케이스 31

<표 11> UC-A03 예약상태 확인 유즈케이스 32

<표 12> 주차 관리 시스템 Terminator 명세 34

<표 13> 주차 관리 시스템 Data Store 명세 34

<표 14> 주차 관리 시스템 Process 명세 34

<표 15> 품질 모델 기술서 38

<표 16> 신뢰성 요구사항 38

<표 17> 성능 요구사항 38

<표 18> 보안성 요구사항 39

<표 19> 사용성 요구사항 39

<표 20> 메인 서버 프로토콜 설계서 40

<표 21> 예약 프로그램 프로토콜 설계서 43

<표 22> 정산기 프로토콜 설계서 45

<표 23> 주차장 공간 관리 프로토콜 설계서 47

<표 24> 주차타워 프로토콜 설계서 48

<표 25> 관리자 프로토콜 설계서 49

<표 26> 단위기능 테스트 개요 표 52

<표 27> 주차타워 입차 기능 테스트 시나리오 53

<표 28> 주차장 입차 기능 테스트 시나리오 53

<표 29> 주차타워 출차 기능 테스트 시나리오 54

<표 30> 주차장 출차 기능 테스트 시나리오 54

<표 31> 주차타워 정산 기능 테스트 시나리오 55

<표 32> 주차장 정산(정산기) 기능 테스트 시나리오 55

<표 33> 클라이언트 인터페이스 기능 테스트 시나리오 55

<표 34> 관리자 인터페이스 기능 테스트 시나리오 56

<표 35> 메인 서버 기능 테스트 시나리오 56

<표 36> 단위기능 테스트 결과 요약서 57

<표 37> 통합테스트 개요 표 58

<표 38> 통합 테스트 결과 요약서 58

<표 39> 업무 변경 내역 59

<표 40> 최종 업무 분담 60

**그 림 목 차**

<그림 1> 6조 조직도 9

<그림 2> UP (Unified Process) 통합 프로세스 모델 10

<그림 3> UP(Unified Process) 통합 프로세스 방법 10

<그림 4>WBS 11

<그림 5> 네트워크 차트 12

<그림 6> 간트 차트 13

<그림 7> Usecase Diagram 17

<그림 8> Data Flow Diagram 33

<그림 9> 예약제 주차 시퀀스 다이어그램 36

<그림 10> 비예약제 주차 시퀀스 다이어그램 36

<그림 11> 시스템 내부 시퀀스 다이어그램 37

<그림 12> 통합 테스트 일정 요약 그림 58

<그림 13> 최종 업무 분담 60

* 1. **개요**

본 문서는 주차장 관리 시스템 개발에 있어서 개발의 절차와 과정 그리고 일정 등 고객의 이해를 돕고 사용자의 요구사항을 반영하기 위해 작성되었다.

* 1. 프로그램 개요

오늘날 자동차 등록대수는 감소 추세에 있어, 주차자리 확보 비율은 증가하고 있다. 주차장의 생존 경쟁이 치열해지는 상황속에서, 더불어 고임금 시대에 접어들어 주차장 매출에서 인건비가 차지하는 비중이 높아지고 있다. 즉 인건비 감소가 주차장의 생존을 위한 최소한의 필수요소이다. 인건비 감소와 함께 사용자의 만족을 이끌어내기 위하여 새로운 주차관리 시스템이 필요하다.

본 프로젝트에서는 두 가지 목표를 가진다. 첫째로 인건비 감소를 위한 무인 정산 시스템 도입이다. 유인 정산을 없애 인건비 감소를 노린다. 두번째로 현재 대형 주차장의 혼잡도를 해소하기 위한 해결책을 제시한다. 현재 대형 주차장의 혼잡도는 무인 정산 시스템이 존재하더라도 이를 이용하지 않고, 출구에서 유인 정산을 거치기 때문에 출구에서의 혼잡이 일어난다. 본 프로젝트에서는 무인 정산 시스템을 이용하지 않으면 출차할 수 없도록 설계하여 혼잡도를 해소할 수 있도록 한다.

* 1. 실무 담당자와 면담(주차장 아르바이트 경험자와 회의)

본 팀의 동대문 메리어트 호텔 주차장 아르바이트 경험자인 팀원과 회의를 진행하였고, 현장에서 체감한 정보를 바탕으로 자료를 수집하였다.

현장에서는 무인 정산 시스템과 유인 정산을 혼용하는 환경이었지만, 혼용되는 시스템 속에서 고객들이 무인정산을 해야 하는지 유인정산을 해야 하는지 혼동하였고, 무인 정산 시스템이 존재하였지만, 유인 정산 시스템도 존재하여 출차 지연이 존재했음 또한 출차시 정산소의 바리케이트로 인하여 주차장 전반에 혼잡도가 심각하였다.

따라서 본 팀은 주차장 관리 시스템으로 무인 정산 시스템만을 사용하기로 채택하였으며, 정산소를 두지 않는 것으로 결정하였다.

* 1. 개발범위

1. 일반주차공간
2. 예약시스템 및 주차타워
3. 유저 인터페이스
4. 관리자 인터페이스
5. 결제 기능은 간소화하여 개발하고 주차장 관리 시스템에 집중한다.
   1. 타당석 분석
      1. 경제적 타당성

현재 한국에서 주차장 뿐만 아니라 사회 전반적으로 무인 주차장 시스템이 확산되고 있으며, 일본의 경우 이미 무인 주차장 시스템이 활성화 되어있다. 이를 통해, 앞으로도 발전가능성이 높으며, 시장성이 크다고 볼 수 있다. 고임금 시대와 맞물려, 무인 주차장 시스템 구축이 장기적으로 인건비 및 효율성 측면에서 좋기 때문에, 이용가능성이 높다.

본 주차장 관리 시스템 설치비용은 본 시스템에 필요한 기기들과 센서들의 필요성에 의해, 비교적 많은 초기비용이 들어가지만, 무인 시스템의 도입으로 이후 운용에서 더 많은 이익을 창출할 수 있게 된다.

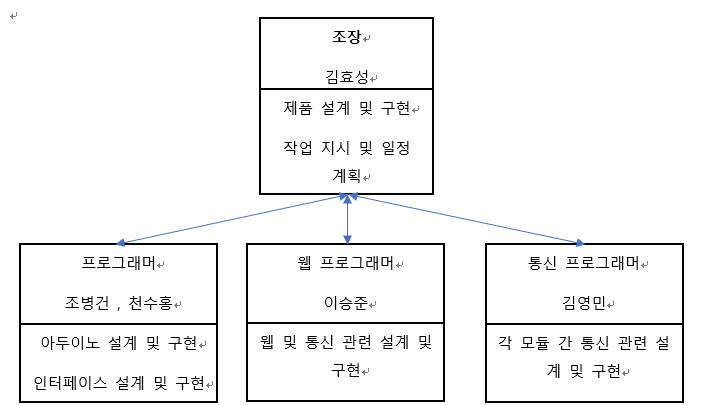
* + 1. 기술적 타당성

본 프로젝트에서는 아두이노를 통해 주차장 모델을 만들어 보일 예정이다. 하드웨어 측면에서 아두이노의 부품 중에서도 간단한 부품을 이용할 것으로 예상되므로, 소프트웨어만 구축이 된다면 실생활에서도 단순한 기계부품으로도 구축이 가능할 것으로 판단된다.

* 1. 산출물리스트

1. 개발 계획서
2. 요구 분석 명세서
3. 설계서
4. 주차 관리 시스템
5. **계획**
   1. 조직도 - 상호 교류식 조직

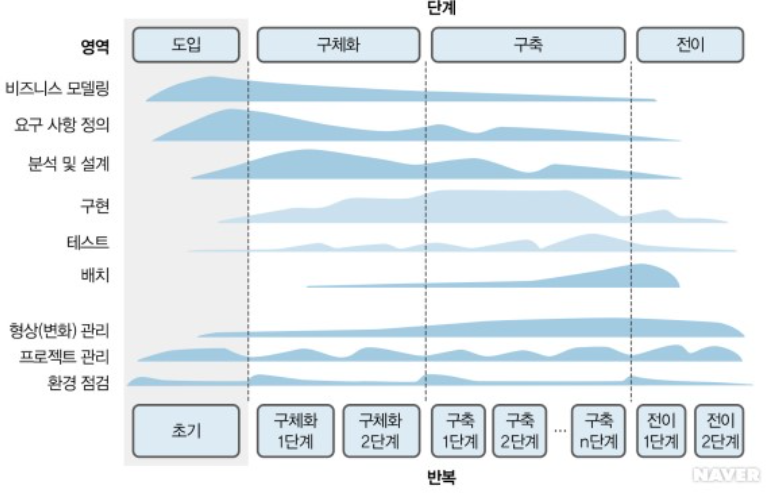
<그림 1> 6조 조직도



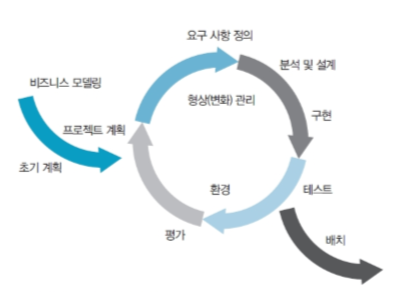
* 1. 역할 및 책임

1. 조장은 프로젝트의 전체적인 부분의 계획, 설계 및 구현을 맡는다.
2. 각 조원은 담당한 세부적인 부분의 계획, 설계 및 구현을 맡는다.
3. 공학 설계 수강으로 아두이노 개발 경험을 가지고 있는 조병건, 천수홍 조원이 아두이노 프로그래밍 및 아두이노와 통신하는 인터페이스 프로그래밍을 맡는다.
4. Hufspoon 개발 경험을 가지고 있는 이승준 조원은 예약시스템에 필요한 웹 프로그래밍 및 웹과 메인컴퓨터 DB 통신 프로그래밍을 맡는다.
5. Phython 개발 경험을 가지고 있는 김영민 조원은 각 모듈을 통합하고 통신하도록 프로그래밍을 하고, 다른 조원을 도울 수 있도록 한다.
   1. 인원투입계획
6. 맡은 부분의 계획 작업량을 조기 완료하였을 경우, 다른 조원을 보조하도록 진행중인 작업에 투입하여 유연성을 높인다.
7. 예약시스템의 설계 및 개발 기간이 길고, 우선순위가 높기 때문에 초반 모든 조원은 이승준 조원을 보조하여 설계 및 개발 기간을 최대한 줄일 수 있도록 한다.
8. 예약시스템의 개발 후에 예정된 작업으로 투입한다.
   1. 개발 방법론

<그림 2> UP (Unified Process) 통합 프로세스 모델



본 프로젝트에서는 설계 시 반복적 개발 방법론을 사용하여 체계개발을 수행하며, 반복적 개발 방법론 중에서도 UP (Unified Process) 방식을 택하여 반복을 통해 점진적인 설계 및 개발을 진행하도록 한다.

<그림 3> UP(Unified Process) 통합 프로세스 방법

본 프로젝트에서는 초기계획, 요구사항정의, 분석 및 설계, 구현, 테스트, 평가를 각 단계(도입, 구체화, 구축, 전이)에서 반복하며 개발을 진행한다.

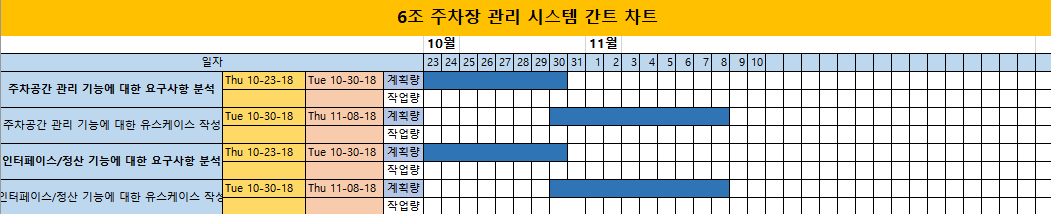
1. **일정계획**
   1. WBS (Work Breakdown Structure) <그림 4>WBS

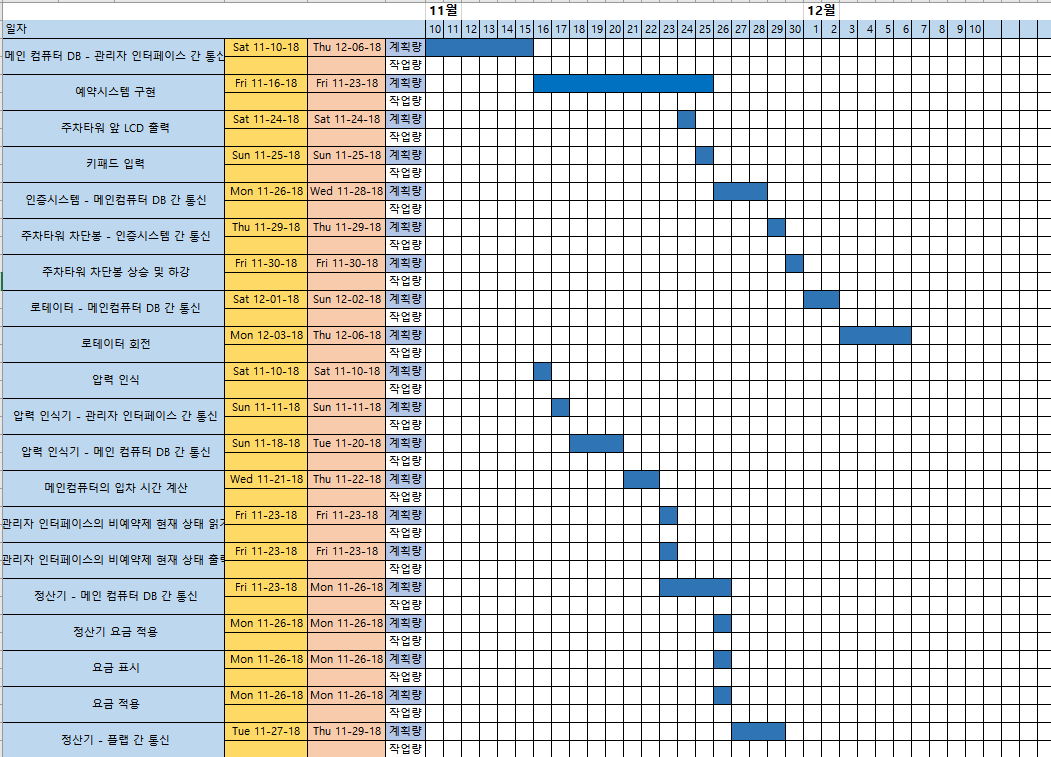


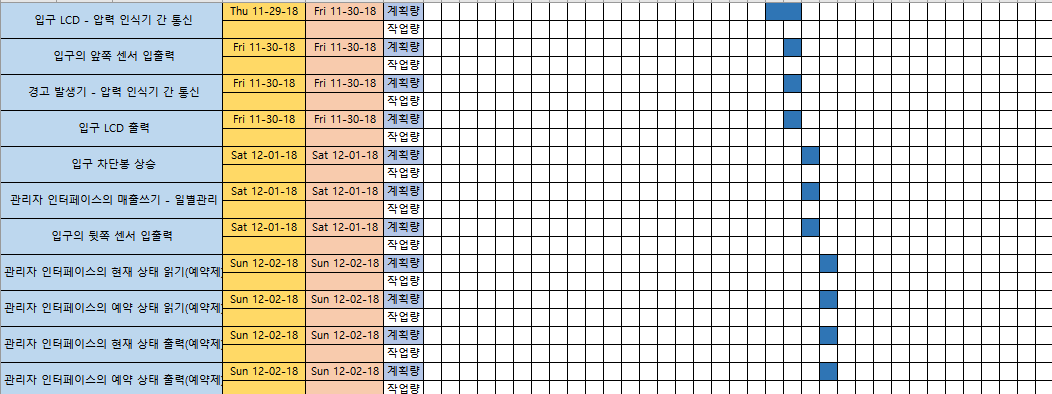
* 1. 텍스트이(가) 표시된 사진

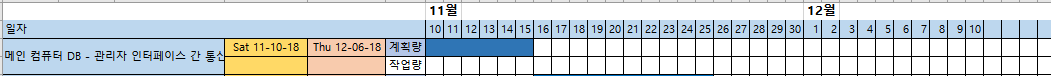
     매우 높은 신뢰도로 생성된 설명 네트워크 차트 <그림5>네트워크 차트
  2. 간트 차트

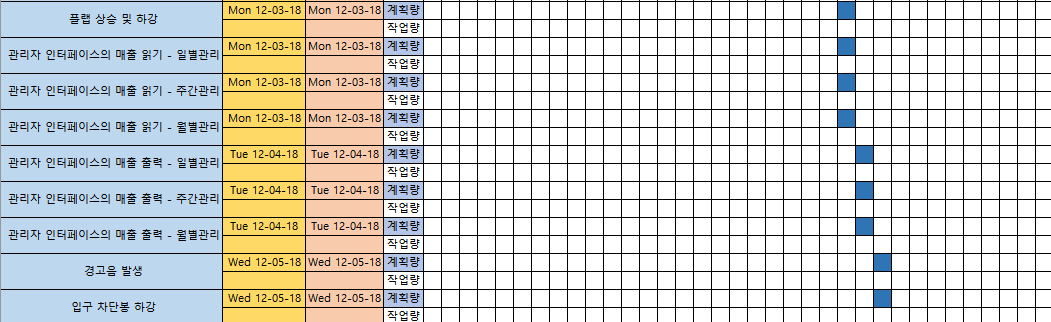
<그림 6> 간트 차트











1. **위험요소 관리계획**
   1. 계획 및 설계 사항 변경

프로젝트 기간 자체가 짧아, 한정적인 계획 기간에 수립한 계획이 프로젝트 진행 중 변경될 가능성이 있으며, 배경지식과 구현에 필요한 적합한 프로그래밍 언어에 대한 기반 지식 부족으로 설계서의 기록한 내용과 변경된 방향으로 프로젝트가 진행될 확률이 크다.

* 현재 팀원들의 프로그래밍 자질을 고려하여, 주차관리 시스템의 복잡한 부분은 간소화, 모델링하여 프로젝트를 진행한다.
  1. 하드웨어 통신 기술 구축에 대한 배경지식 부족

아두이노에 대한 지식이 부족하여, 각 모듈을 개발하고도 하드웨어를 구축하지 못하거나 서브시스템간 통신기술의 부족으로, 작업 소요시간을 지키지 못할 가능성이 크다.

* 하드웨어 부분을 구축할 때는, 책임 업무가 비교적 빨리 끝난 팀원이 조력자가 된다.
  1. 아두이노 보드 개수가 한정적이다.

각 팀원이 업무를 분담 받고, 각 기능을 구현할 때 아두이노 보드가 필수적이다. 그러나 아두이노 보드 개수가 한정적이라 각 팀원이 아두이노 보드가 없어 구현이 어려울 수 있다.

* 각 팀원이 한 지역에 거주하고 있는 것을 이용하여, 매주 매일 모여서 함께 작업을 하기로 하여, 최대한 지연되는 시간을 줄인다.

1. **개발환경**

운영체제: Windows 10

개발언어: 아두이노, 파이썬

이용S/W:

데이터베이스: 텍스트 파일

서 버: 노트북

* 1. 개발 및 설계 도구

프로젝트에서는 아래 <표 1>과 같은 개발도구를 사용하여 분석 및 설계를 함.

<표 1> 개발 및 설계 도구

|  |  |  |
| --- | --- | --- |
| **개발 및 설계 도구** | **용도** | **비고** |
| **Raspberry** | **인터페이스 및 주차장 설계** |  |
| **Python** | **각 모듈 구현** |  |
| **Text file** | **데이터 베이스** |  |

* 1. 프로젝트 개발 목표
     1. 주차장관리 시스템 개발 목표

설계한 디자인을 가시적으로 보여줄 수 있는Arduino, 모든 조원이 프로그래밍이 가능한 언어인 Python을 기반으로, 주차공간, 주차공간을 이용하는 데 필요한 인터페이스 및 인터페이스와 모듈간의 통신을 구현하여 최종 주차장 관리 시스템을 개발함을 목표로 한다.

* + 1. 주차장관리 시스템 세부 개발 목표

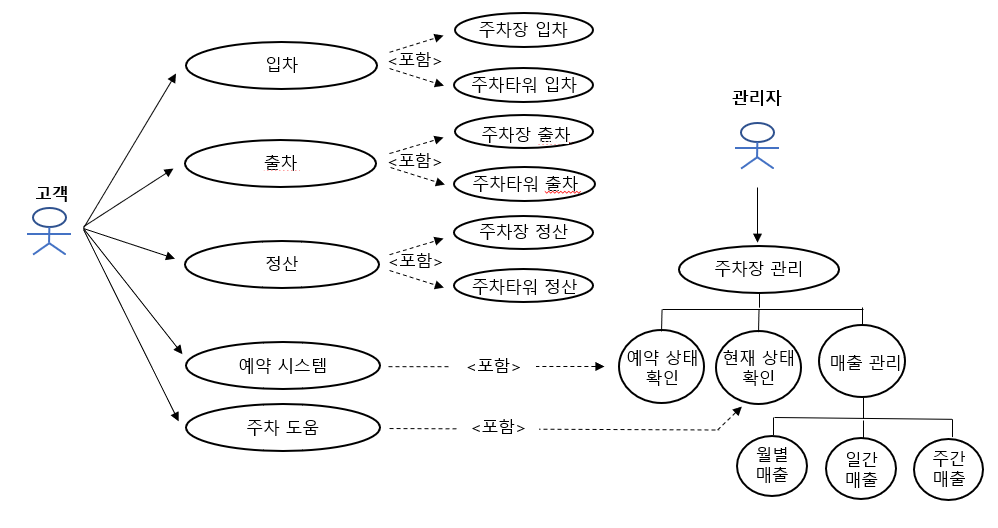
아두이노를 통해 프로젝트의 주차 공간 부분을 가시적으로 구현하고, 파이썬을 통해 노트북인 서버에 주차장 이용 고객이 이용할 수 있는 정산시스템과 같은 사용자 인터페이스를, 각 모듈 간 통신과 데이터베이스의 운용을 통해 관리자가 이용할 수 있는 인터페이스를 구현한다. 또한 TCP 소켓 프로그래밍 또는 웹을 통해 주차장 고객이 예약시스템을 이용할 수 있도록 한다.

주차 공간 부분은 Arduino와 Python을 기반으로 가시적으로 주차장을 구현한다. 구체적으로는 아래와 같다.

1. 주차 공간 표시
2. 차량 입,출차
3. 플랩 작동
4. 주차선 경고
5. 인증 시스템
6. 주차 타워

인터페이스 시스템은 사용자, 관리자 가 볼 수 있도록 하는 것을 목적으로 하며, 요금정산, 차량조회, 매출현황 등 사용자, 관리자 입장에서 편하게 볼 수 있도록 하는 것으로 목적으로 수행을 한다. 구체적으로는 아래와 같다.

1. 사용자 인터페이스
   1. 정산
   2. 요금 표시
   3. 예약 시스템
2. 관리자 인터페이스
   1. 매출 관리
   2. 주차 현재상태 확인
   3. 예약상태 확인
3. **주차장 관리 시스템 요구사항 명세**
   1. 기능적 요구 사항
      1. Usecase Diagram <그림 7> Usecase Diagram



* 1. Usecase 모델
     1. Actor 명세

주차장 관리 시스템에는 고객과 관리자, 두 액터를 갖는다.

<표 2> 주차장 관리 시스템 액터 명세

|  |  |
| --- | --- |
| Actor | Description |
| 고객 | 비예약제 고객은 예약 시스템을 사용하지 않고 주차장을 방문한다. 해당 고객은 일반 주차장에서 주차 시 각종 편의 기능을 제공받는다. 예약제 고객은 예약 시스템을 사용하여 주차타워를 사용한다. |
| 관리자 | 관리자는 매출관리, 예약상태를 관리하고, 예약제 주차장과 비예약제 주차장 상태를 확인할 수 있다. |

* + 1. Usecase 개요

주차장 관리 시스템의 유즈케이스는 다음 표와 같다.

<표 3> 주차장 관리 시스템 유즈케이스 식별자 목록

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 유스케이스 식별자 목록 | | | | | |
| 식별자 | 행위자 | 설명 | 식별자 | 행위자 | 설명 |
| UC-C01 | 고객 | 입차 | UC-A01 | 관리자 | 매출 관리 |
| UC-C02 | 고객 | 출차 | UC-A02 | 관리자 | 주차 현재상태 확인 |
| UC-C03 | 고객 | 정산 | UC-A03 | 관리자 | 예약상태 확인 |
| UC-C04 | 예약제 고객 | 예약 시스템 |  |  |  |
| UC-C05 | 비예약제 고객 | 주차 보조 |  |  |  |

* 1. Usecase 명세
     1. 입차 유즈케이스

<표 4> UC-C01 입차 유즈케이스

|  |  |  |
| --- | --- | --- |
| **유즈케이스 식별자/명** | UC-C01 입차 유즈케이스 | |
| **개요** | 주차장 - 비예약제 고객의 차량이 주차장에 입차하려고 할 시, 두가지 상황이 존재한다. 첫번째로 주차장이 만차일 때, 차단봉을 올리지 않고 만차라고 표시하여 주차장 내 혼란을 방지한다. 두번째로 주차공간이 남아있을 때, 빈주차공간을 표시하여 보여주고 비예약제 고객을 입차시켜 주차장 내 혼란을 방지한다. 비예약제 고객의 차량이 입차하여 빈주차공간에 주차를 하여, 주차가 완료되면, 플랩이 상승한다. | |
| 주차타워 - 예약제 고객이 웹에서 예약하여 결제를 완료하면 (UC-C05), 인증코드를 발급을 받으며, 예약시간에 예약제 고객이 주차타워 앞에 오면 인증코드를 통해 인증을 한다. 후에 고객이 주차타워 맨 밑 층에 입차를 하여 로테이터가 해당 차량을 윗층으로 올린다. | |
| **주 행위자** | 주차장 - 비예약제 고객 | |
| 주차타워 – 예약제 고객 | |
| **부 행위자** | 해당사항 없음 | |
| **사전 조건** | 주차장 - 먼저 입차되어 있는 주차공간의 플랩이 구동되어 현재 주차상태를 알 수 있으며, 플랩의 구동은 차량이 입차 완료 시에 진행이 된다. | |
| 주차타워 - 고객이 예약 시스템을 통해 예약을 완료하면(UC-C05) 인증코드를 발급한다. 아울러 예약 시간에 온 뒤 인증 절차를 걸친 후 해당 주차타워 앞에 온다. | |
| **기본 흐름 – 주차장** | | |
| **행위자** | | **시스템** |
| 1. 비예약제 고객의 차량이 대기한다. | |  |
|  | | 1. 입구의 앞센서가 차량을 감지한다. |
|  | | 1. 입구의 LCD가 빈 주차 공간을 표시한다.(UC-C02 빈주차공간 표시) |
|  | | 1. 차단봉이 올라간다. |
| 1. 비예약제 고객의 차량이 입차한다. | |  |
|  | | 1. 입구의 뒷센서가 차량을 감지한다. |
|  | | 1. 차단봉이 내려간다. |
| 1. 차량을 주차한다. | |  |
|  | | 1. 플랩이 상승한다. |
| **기본 흐름 – 주차타워** | | |
| **행위자** | | **시스템** |
| 1. 고객이 웹을 통해 예약 시간을 선택한다. | |  |
| 1. 고객이 결제를 완료한다. | |  |
|  | | 1. 인증코드를 발급한다 |
| 1. 예약 시간에 예약제 고객이 주차타워에 도착한다. | |  |
| 1. 인증코드를 통해 인증한다. | |  |
|  | | 1. 주차타워 앞 차단봉이 상승한다. |
| 1. 예약제 고객이 주차타워 1층에 주차한다. | |  |
| 1. 예약제 고객이 예약시간에 맞춰 출차 하지 않는다. | |  |
|  | | 1. 해당 차량을 윗층으로 올린다. |
| **대안 흐름 – 주차장** | | |
| **행위자** | | **시스템** |
| **[A1.** **이미 주차장이 만차인 경우]**   1. 비예약제 고객의 차량이 대기한다. | |  |
|  | | 1. 입구의 센서가 차량을 감지한다. |
|  | | 1. 입구의 LCD가 주차장이 만차라고 표시한다. (UC-C06 주차 보조 ) |
|  | | 1. 차단봉이 올라가지 않는다. |
| **대안 흐름 – 주차 타워** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **예외상황 시나리오 – 주차장** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **예외상황 시나리오 – 주차타워** | | |
| **행위자** | | **시스템** |
|  | | **[A1. 예약제 고객이 발급받은 인증코드를 잘못 입력하였을 경우]**  1. 예약제 고객이 입력한 인증코드에 오류가 있음을 알리고, 인증코드 재입력을 요청한다 |
| **사후조건** | | 주차장 - 현재 주차상태에 따라 차단봉이 동작한다. |
| 주차타워 - 인증을 완료하면 주차타워 앞 차단봉이 올라가 예약제 고객이 주차타워에 입차한다. |

* + 1. 출차 유즈케이스

<표 5> UC-C02 출차 유즈케이스

|  |  |  |
| --- | --- | --- |
| **유즈케이스 식별자/명** | UC-C02 출차 유즈케이스 | |
| **개요** | 주차장 – 비예약제 고객이 정산 인터페이스를 통해 정산을 완료하면 (UC-C03) 플랩이 하강 하여 비예약제 고객의 차량이 출차할 수 있다. 정산을 완료해야만 출차가 가능하기 때문에 출구에서 혼잡도가 감소한다. 이때 관리자 인터페이스에서 주차 현재 상태를 확인할 수 있다. (UC-A02) | |
| 주차타워 – 예약제 고객이 추가 요금에 대한 정산을 완료 시에 또는 예약시간에 맞춰 예약제 고객이 출차하려고 할 때 해당 차량을 1층으로 내린다. | |
| **주 행위자** | 주차장 - 비예약제 고객 | |
| 주차타워 – 예약제 고객 | |
| **부 행위자** | 해당사항 없음 | |
| **사전 조건** | 주차장 – 비예약제 고객이 정산 인터페이스를 통해 정산을 완료한다. | |
| 주차타워 – 예약제 고객이 정산 인터페이스를 통해 정산을 완료한다. | |
| **기본 흐름 – 주차장** | | |
| **행위자** | | **시스템** |
| 1. 비예약제 고객이 정산 인터페이스를 통해 정산을 완료 한다. | |  |
|  | | 1. 플랩이 하강한다. |
| 1. 출차를 수행한다. | |  |
| **기본 흐름 – 주차타워** | | |
| **행위자** | | **시스템** |
| 1. 예약제 고객이 정산을 완료한다. | |  |
|  | | 1. 해당 차량을 1층으로 내린다. |
| 1. 예약제 고객이 출차를 수행한다. | |  |
| **대안 흐름 – 주차장** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **대안 흐름 – 주차 타워** | | |
| **행위자** | | **시스템** |
| **[A1.** 예약 시간에 맞춰 예약제 고객이 출차할 경우**]**  1. 예약제 고객의 차량이 출차한다. | |  |
| **예외상황 시나리오 – 주차장** | | |
| **행위자** | | **시스템** |
| **[A1.** **비예약제 고객이 정산을 완료하여 플랩이 하강되었는데 일정 시간동안 출차 하지 않을 경우]**   1. 정산을 완료한다.(UC-C08) | |  |
|  | | 1. 플랩이 하강한다. |
|  | | 1. 일정 시간동안 압력 인식기에서 압력이 계속 인식된다. |
|  | | 1. 플랩을 상승시키고 주차상태로 전환한다. |
| **예외상황 시나리오 – 주차타워** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **사후조건** | | 주차장 – 플랩이 하강한다. |
| 주차타워 – 해당사항 없음 |

* + 1. 정산 유즈케이스

<표 6> UC-C03 정산 유즈케이스

|  |  |  |
| --- | --- | --- |
| **유즈케이스 식별자/명** | UC-C03 정산 유즈케이스 | |
| **개요** | 주차장 – 비예약제 고객은 정산 시스템을 통해 정산을 완료해야만이 플랩이 하강되어 출차를 수행 할 수 있다. 정산은 정산 인터페이스를 통해 할 수 있고, 요금 표시 기능이 포함되어 있다. 아울러 정산 인터페이스에서 요금 표시 기능을 선택하면, 해당 고객 차량의 입차시간을 계산하여 요금을 적용한 후 출력한다. | |
| 주차타워 – 예약제 고객은 초과된 금액에 대해 정산을 완료해야만이 로테이터가 해당 차량을 1층으로 내려, 예약제 고객이 출차를 수행 할 수 있다. | |
| **주 행위자** | 공통 - 비예약제 고객, 예약제 고객 | |
| **부 행위자** | 해당사항 없음 | |
| **사전 조건** | 주차장 – 비예약제 고객이 입차한다. | |
| 주차타워 – 예약제 고객이 예약 시간을 초과한다. | |
| \*\* 공통 – 정산 인터페이스를 사용한다. | |
| **기본 흐름 – 주차장** | | |
| **행위자** | | **시스템** |
| 1. 비예약제 고객이 정산 인터페이스를 이용한다. | |  |
|  | | 1. 입차로 계산된 시간과 요금을 정산 인터페이스에 출력한다. |
| 1. 고객이 요금을 지불한다. | |  |
|  | | 1. 비예약제 고객의 주차공간 플랩을 하강시킨다. |
| **기본 흐름 – 주차타워** | | |
| **행위자** | | **시스템** |
| 1. 예약제 고객이 정산 인터페이스를 이용한다. | |  |
|  | | 1. 추가된 시간과 요금을 정산 인터페이스에 출력한다. |
| 1. 예약제 고객이 요금을 지불한다. | |  |
|  | | 1. 예약제 고객의 차량을 주차타워 1층으로 내린다. |
| **대안 흐름 – 공통** | | |
| **행위자** | | **시스템** |
|  | | **[A1. 사용자가 더 많은 돈을 내는 경우]**  1. 거스름돈을 출력한다. |
| **예외상황 시나리오 – 주차장** | | |
| **행위자** | | **시스템** |
|  | | **[A1. 사용자가 정산을 한 후, 출차하지 않는 경우]**  1. 압력이 일정시간 지속되면, 플랩을 다시 상승시킨다. |
|  | | **[A1. 잘못된 주차공간을 입력하였을 경우]**  1. 잘못된 주차공간을 입력하였다고 오류를 출력한 후, 재입력을 요청한다 |
| **예외상황 시나리오 – 주차타워** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **사후조건** | | 주차장 – 정산이 완료되면 해당 주차공간의 플랩을 하강 시킨다. |
| 주차타워 – 해당 차량을 주차타워 1층으로 내린다. |
| 공통 – 요금을 출력하여 고객에게 보여준다. |

* + 1. 예약 시스템 유즈케이스

<표 7> UC-C04 예약 시스템 유즈케이스

|  |  |  |
| --- | --- | --- |
| **유즈케이스 식별자/명** | UC-C04 예약 시스템 유즈케이스 | |
| **개요** | 주차타워를 사용하려는 고객은 주차관리웹에서 예약시간을 정하고 인증코드를 발급받고 결제한다. 이때 이미 예약된 시간에는 예약할 수 없다. 이 때, 관리자는 인터페이스를 통해 예약 상태를 확인할 수 있다(UC-A03). 예약제 고객이 예약시간이 되어 주차타워에 도착하면, 주차타워 앞 인증 시스템에서 예약 시 발급받은 인증코드를 입력한다(UC-C06). 인증을 받은 뒤 주차타워 앞 차단봉이 열리고 주차타워에 주차한다. 예약시간이 다 되면 예약제 고객은 주차타워에서 출차한다 | |
| **주 행위자** | 예약제 고객, 관리자 | |
| **부 행위자** | 해당사항 없음 | |
| **사전 조건** | 예약제 고객이 예약 인터페이스를 실행한다. | |
| **기본 흐름** | | |
| **행위자** | | **시스템** |
| 1. 고객이 예약 인터페이스를 실행하면, 예약가능한 시간이 출력된다. | |  |
| 2. 고객이 예약가능한 시간을 선택한다. | |  |
| 3. 고객이 결제를 완료한다. | |  |
|  | | 4. 인증코드를 발급한다. |
| **대안 흐름** | | |
| **행위자** | | **시스템** |
|  | | **[A1.** **예약시간이 모두 차서 예약이 불가능한 경우 ]**  1. 예약이 불가능함을 출력한다. |
|  | | **[A2. 예약인증절차에 실패한 경우 ]**  1. 예약인증이 되지 않았음을 출력한다. |
| **예외상황 시나리오** | | |
| **행위자** | | **시스템** |
|  | | [A1. 예약 시간이 지난 경우]  1. 비어있는 주차타워로 차량을 올린 후, 추가요금을 정산해야 출차를 할 수 있게 한다. |
| **사후조건** | | 인증코드를 발급한다. |

* + 1. 주차 보조 유즈케이스

<표 8> UC-C05 주차 보조 유즈케이스

|  |  |  |
| --- | --- | --- |
| **유즈케이스 식별자/명** | UC-C05 주차 보조 유즈케이스 | |
| **개요** | 비예약제 고객의 차량이 주차장에 입차할 시, 입구의 LCD가 주차공간 중 비어있는 공간을 표시하여 주차장 내 혼란을 방지한다. 만약 비어있는 주차공간이 없을 시, 주차장이 만차라고 표시한다. 아울러 주차선에 벗어나도록 주차할 시, 양 옆 주차고객이 불편함을 느끼며, 접촉 사고등 사고 가능성이 발생한다. 따라서, 비예약제 고객이 주차선에 벗어나도록 주차하면 경고음을 울려 비예약제 고객을 정확하게 주차하도록 유도한다. | |
| **주 행위자** | 비예약제 고객 | |
| **부 행위자** | 해당사항 없음 | |
| **사전 조건** | 먼저 입차되어 있는 주차공간의 플랩이 구동되어 현재 주차상태를 알 수 있다. | |
| **기본 흐름** | | |
| **행위자** | | **시스템** |
| 1. 비예약제 고객의 차량이 대기한다. | |  |
|  | | 1. 입구의 앞센서가 차량을 감지한다. |
|  | | 1. 입구의 LCD가 빈 주차공간을 표시한다. |
| 1. 비예약제 고객이 빈 주차공간에 주차선을 침범하여 주차한다. | |  |
|  | | 1. 경고음이 울린다. |
| 1. 정확하게 주차한다. | |  |
|  | | 1. 경고음이 꺼진다. |
| **대안 흐름** | | |
| **행위자** | | **시스템** |
| **[A1.** **이미 주차장이 만차인 경우]**   1. 비예약제 고객의 차량이 대기한다. | |  |
|  | | 1. 입구의 앞센서가 차량을 감지한다. |
|  | | 1. 입구의 LCD가 주차장이 만차라고 표시한다. |
| **예외상황 시나리오** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **사후조건** | | 현재 주차상태에 따라 빈주차공간을 표시하거나 만차라고 표시하며, 주차가 정확히 완료되면 경고음이 꺼져야 한다. |

* + 1. 매출 관리 유즈케이스

<표 9> UC-A01 매출 관리 유즈케이스

|  |  |  |
| --- | --- | --- |
| **유즈케이스 식별자/명** | UC-A01 매출 관리 유즈케이스 | |
| **개요** | 관리자가 관리자 인터페이스에 접속하여 매출을 관리한다. 이 때 관리자는 월별, 주간, 일별로 매출을 관리할 수 있다. | |
| **주 행위자** | 관리자 | |
| **부 행위자** | 해당사항 없음 | |
| **사전 조건** | 고객이 정산 인터페이스를 통해 정산을 완료한다 | |
| **기본 흐름** | | |
| **행위자** | | **시스템** |
|  | | 1. 정산 인터페이스를 통해 정산 결과를 전달받는다. |
|  | | 2. 데이터 베이스에 정산 결과를 저장한다. |
| 3. 관리자가 관리자 인터페이스에 접속한다. | |  |
|  | | 4. 월별, 주간, 일별로 매출 관리 한다. |
| **대안 흐름** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **예외상황 시나리오** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **사후조건** | | 해당사항 없음 |

* + 1. 주차 현재상태 확인 유즈케이스

<표 10> UC-A02 주차 현재상태 확인 유즈케이스

|  |  |  |
| --- | --- | --- |
| **유즈케이스 식별자/명** | UC-A02 주차 현재상태 확인 유즈케이스 | |
| **개요** | 관리자가 관리자 인터페이스로 접속하여 현재 비예약제 주차 공간 상태를 확인할 수 있다. | |
| **주 행위자** | 관리자 | |
| **부 행위자** | 해당사항 없음 | |
| **사전 조건** | 플랩 구동을 통해 비예약제 주차 공간 상태를 전송받는다(UC-.C04). | |
| **기본 흐름** | | |
| **행위자** | | **시스템** |
|  | | 1. 플랩 구동을 통해 비예약제 주차 공간 상태를 전송 받는다. |
| 2. 관리자가 관리자 인터페이스로 접속한다. | |  |
|  | | 3. 관리자가 주차 현재상태를 확인한다. |
| **대안 흐름** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **예외상황 시나리오** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **사후조건** | | 관리자 인터페이스에서 주차 현재상태를 출력한다. |

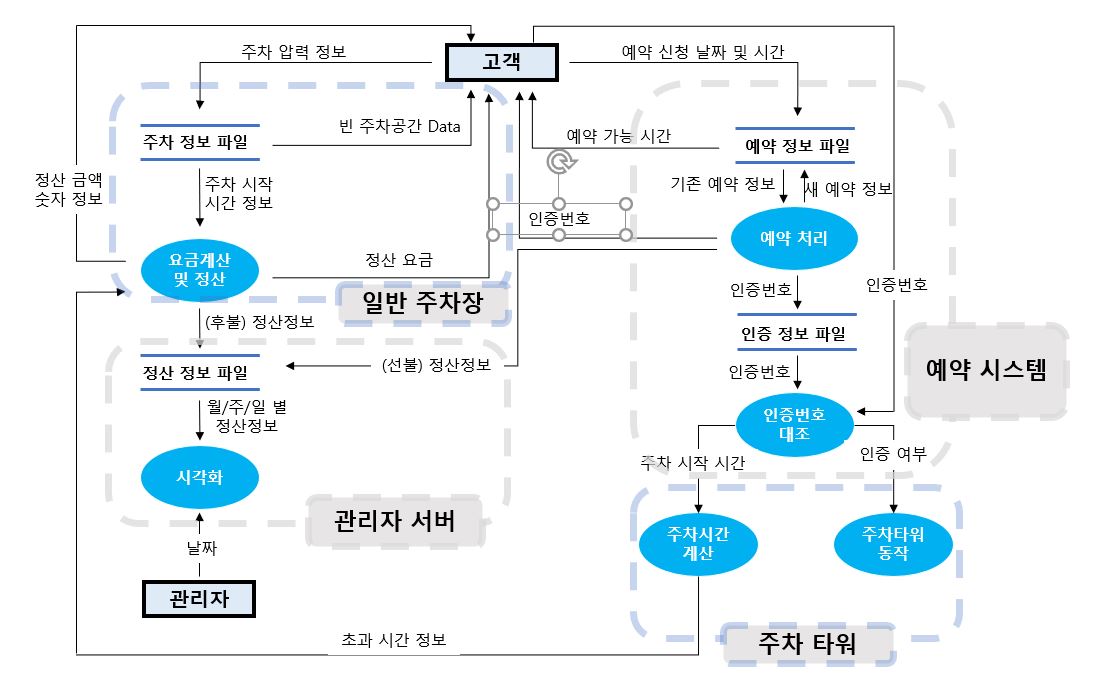
* + 1. 예약상태 확인 유즈케이스

<표 11> UC-A03 예약상태 확인 유즈케이스

|  |  |  |
| --- | --- | --- |
| **유즈케이스 식별자/명** | UC-A03 예약상태 확인 유즈케이스 | |
| **개요** | 관리자가 관리자 인터페이스로 접속하여 예약상태를 확인할 수 있다. | |
| **주 행위자** | 관리자 | |
| **부 행위자** | 해당사항 없음 | |
| **사전 조건** | 예약제 고객이 예약 인터페이스(UC-C05)를 통해 예약을 완료한다. | |
| **기본 흐름** | | |
| **행위자** | | **시스템** |
| 1. 예약제 고객이 예약 인터페이스(UC-C05)를 통해 예약을 완료한다. | |  |
|  | | 2. 관리자가 관리자 인터페이스로 접속한다. |
|  | | 3. 관리자가 예약상태를 확인한다. |
| **대안 흐름** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **예외상황 시나리오** | | |
| **행위자** | | **시스템** |
| 해당사항 없음 | | |
| **사후조건** | | 관리자 인터페이스에 예약상태를 출력한다. |

1. **DFD**

<그림 8> Data Flow Diagram



* 1. Terminator 명세

주차장 관리 시스템에는 다음과 같은 터미네이터가 존재함.

<표 12> 주차장 관리 시스템 Terminator 명세

|  |
| --- |
| Terminator |
| 고객 |
| 관리자 |

* 1. Data Store 명세

주차장 관리 시스템에는 다음과 같은 자료 저장소가 존재하고, 모든 자료 저장소는 주차 관리 서버를 통해 접근할 수 있다.

<표 13> 주차장 관리 시스템 Data Store 명세

|  |  |
| --- | --- |
| Data store | Description |
| 주차 정보 파일 | 주차공간 별 주차 시작 시간을 저장함. |
| 예약 정보 파일 | 날짜, 시간 별 예약 상태를 저장함. |
| 정산 정보 파일 | 날짜 별 정산 금액을 저장함. |
| 인증 정보 파일 | 날짜, 시간 별 예약에 발급된 인증 번호를 저장함. |

* 1. Process 명세

주차장 관리 시스템에는 다음과 같은 처리가 존재함.

<표 14> 주차장 관리 시스템 Process 명세

|  |  |
| --- | --- |
| Process | Description |
| 요금계산 및 정산 | 주차 정보 파일로부터 주차 시작 시간을 받아 정산 금액을 고객에게 알려주며, 정산 요금을 받으면 정산 정보 파일에 금액을 저장한다. |
| 예약 처리 | 예약 정보 파일로부터 기존 예약 정보를 받아와 예약을 하고, 해당 예약 정보를 예약 정보 파일에 저장한다. 또한 해당 예약에 발급한 인증 번호를 인증 정보 파일에 저장한다. |
| 인증 번호 대조 | 고객이 인증 번호를 입력하면, 인증 정보 파일에서 입력한 시간대의 인증 번호와 대조하여 주차타워에 인증 여부를 전달한다. |
| 주차시간 계산 | 정산기로부터 주차타워 정산 신호가 들어오면 초과 시간 정보를 정산기에 전달한다. |
| 주차타워 동작 | 전달받은 인증 여부에 따라 차단봉을 동작시킨다. |
| 시각화 | 관리자의 요청에 따라 정산 정보 파일로부터 정산 정보를 전달받아 출력한다. |

1. **시퀀스 다이어그램**
   1. 예약제 주차 시퀀스 다이어그램

<그림 9> 예약제 주차 시퀀스 다이어그램



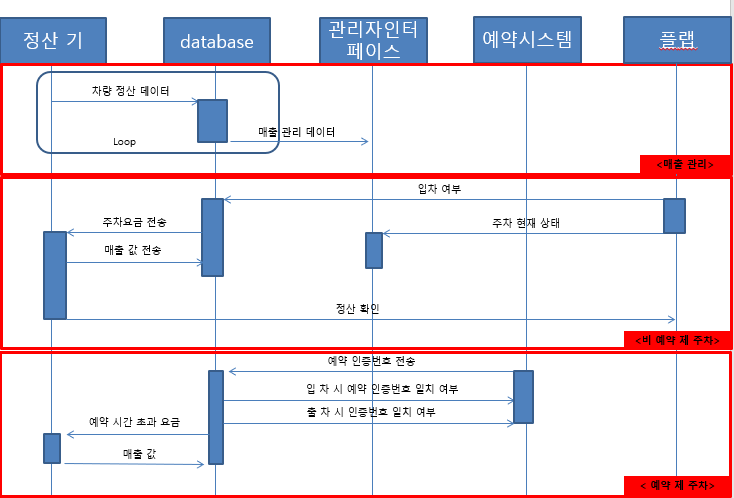
8.2 비 예약제 주차 시퀀스 다이어그램

<그림 10> 비예약제 주차 시퀀스 다이어그램



8.3 시스템 내부 시퀀스 다이어그램

<그림 11> 시스템 내부 시퀀스 다이어그램



1. **비기능적 요구사항**
   1. 시스템 품질 모델

<표 15> 품질 모델 기술서

|  |  |  |  |
| --- | --- | --- | --- |
| **품질목표** | 사용자가 만족하는 품질을 도출 | | |
| **품질특성** | **요구사항내용** | **품질목표(%)** | **우선순위** |
| **신뢰성** | 요구 기능을 정확하고 일관되게 수행하는 능력 | 90 | 1 |
| **성능** | 사용자가 원하는 조건을 만족시키는 능력 | 90 | 2 |
| **보안성** | 허가받은 접근자만 접근시킬 수 있는 능력 | 90 | 3 |
| **사용성** | 사용자가 사용과 이해를 용이하게 할 수 있는 성질 | 70 | 4 |

* 1. 상세 품질 요구사항
     1. 신뢰성 요구사항

<표 16> 신뢰성 요구사항

|  |  |  |
| --- | --- | --- |
| 신뢰성 | | |
| 측정 대상 | MTBF | 고장에서 다음 고장까지의 평균 시간 |
| MTTF | 수정 후 다음 고장까지의 평균 시간 |
| MTTR | 고장 발생 시점에서 수리 시까지의 평균 시간 |
| 측정 방법 | 각 기능 당 특정 횟수를 수행하여 측정한다. | |

* + 1. 성능 요구사항

<표 17> 성능 요구사항

|  |  |  |
| --- | --- | --- |
| 성능 | | |
| 측정 대상 | 응답시간 | 사용자의 요청에 따른 응답이 2초 이내 |
| 데이터 처리량 | 30명의 동시 접속자 |
| 측정 방법 | 예약 시스템에 동시 접속한 상태에서 측정한다. | |

* + 1. 보안성 요구사항

<표 18> 보안성 요구사항

|  |  |  |
| --- | --- | --- |
| 보안성 | | |
| 측정 대상 | 인증번호 | 잘못된 인증 번호를 입력할 시 차단 |
| 허가 받지 않은 접근 | 접근 차단 |
| 측정 방법 | 설계한 API 또는 프로그램 외의 접근자가 서버에 접근한다. | |

* + 1. 사용성 요구사항

<표 19> 사용성 요구사항

|  |  |  |
| --- | --- | --- |
| 사용성 | | |
| 측정 대상 | 관리자 인터페이스 | 메뉴 사용 |
| 예약 시스템 | 날짜,시간 입력 |
| 측정 방법 | 설계자 외의 테스터가 사용한다. | |

* 1. 제약사항

1. Python언어로 사용해 개발하고, CBD개발 방법론을 적용해야 한다.
2. 서버가 사설 IP를 쓰기 때문에, 사용자가 접근 가능하도록 조치해야 한다.
3. 윈도우 운영체제에서 실행할 수 있어야 한다.
4. 개발자들의 데이터베이스 관련 지식 부족으로, 데이터베이스 구축에 관한 대안을 마련해야 한다.
5. **주차장 관리 시스템 프로토콜 설계서**
   1. 메인 서버 프로토콜

<표 20> 메인 서버 프로토콜 설계서

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 메인 서버 | | | | |
| 발신 내용 | ResList@parkLot1@parkLot2@parkLot3@parkLot4 | | 기능 | 예약 리스트 발송 |
| 전달 값 | 날짜  유효 | 헤더@1번예약현황@2번예약현황@3번예약현황@4번예약현황 | 출력 | 없음 |
| 날짜  비유효 | 헤더@새로운예약현황 |
| 받는 값 | 날짜 | | 연결 대상 | 예약 프로그램 |
| 설명 | 예약 프로그램에서 클라이언트가 입력한 날짜를 전송받아 예약 정보 파일에 접근하여 해당 날짜의 예약 리스트를 예약 프로그램에 발송한다. | | | |
| 메인 서버 | | | | |
| 발신 내용 | ResSuc@ACode | | 기능 | 인증코드 발송 |
| 전달 값 | 예약  유효 | 헤더@인증코드 | 출력 | 없음 |
| 예약  비유효 | 예약 프로그램에서 비유효한 예약은  거르고 전송하기 때문에  비유효할 수 없다. |
| 받는 값 | 날짜 | | 연결 대상 | 예약 프로그램 |
| 설명 | 전송받은 예약 정보에 따라 예약 정보 파일에 해당 날짜와 주차 공간에 예약한 후, 인증코드를 만들어 인증 정보 파일에 저장한다. 해당 인증코드를 예약 프로그램에 발송한다. | | | |
| 메인 서버 | | | | |
| 발신 내용 | inqTime@resDate | | 기능 | 예약 정보 발송 |
| 전달 값 | 코드  유효 | 헤더@예약날짜@예약시간@예약공간 | 출력 | 없음 |
| 코드  비유효 | 헤더@None@None@None |
| 받는 값 | 인증코드 | | 연결 대상 | 예약 프로그램 |
| 설명 | 예약 프로그램에서 클라이언트가 입력한 인증코드를 전송받아 해당 인증코드에 해당하는 예약날짜,시간,주차공간을 발송한다. | | | |
| 메인 서버 | | | | |
| 발신 내용 | ACode@isPass@reqParkLot | | 기능 | 인증코드 일치 여부와  주차공간 발송 |
| 전달 값 | 코드  유효 | 헤더@True@예약주차공간 | 출력 | 없음 |
| 코드  비유효 | 헤더@False@None |
| 받는 값 | 인증코드 | | 연결 대상 | 주차 타워 |
| 설명 | 주차 타워의 키패드에서 예약 고객이 인증코드를 누르면, 주차타워에서 메인 서버로 인증코드를 발송하고, 인증 정보 파일에서 해당 인증코드가 발송한 시각의 인증코드와 일치하는지 판단 후, 맞다면 예약주차공간을, 틀리다면 False를 발송한다. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 메인 서버 | | | | |
| 발신 내용 | NonResList@NonResParkingLot | | 기능 | 주차상태 확인 |
| 전달 값 | 코드유효 | 헤더@주차장상태 | 출력 | 없음 |
| 코드비유효 | 오늘의 날짜에 맞게 내부적으로 생성해서 보여주기때문에 유효하지 않은 경우가 없다. |
| 받는 값 | 없음 | | 연결 대상 | 관리자 프로그램 |
| 설명 | 비예약 주차상태를 확인하기 위해 접근하면, 오늘(현재)날짜의 주차상태 데이터를 불러와 전달한다. | | | |
| 메인 서버 | | | | |
| 발신 내용 | EarningList@earningList\_month@earningList\_date | | 기능 | 일별 매출확인 |
| 전달 값 | 코드유효 | 헤더@월@일 | 출력 | 없음 |
| 코드비유효 | 헤더@None@None |
| 받는 값 | 날짜 | | 연결 대상 | 관리자 프로그램 |
| 설명 | 확인하고자 하는 날짜의 월,일을 받아서 매출데이터를 확인해서 전달한다. | | | |
| 메인 서버 | | | | |
| 발신 내용 | EarningList\_month@earningList\_month | | 기능 | 일별 매출확인 |
| 전달 값 | 코드유효 | 헤더@월 | 출력 | 없음 |
| 코드비유효 | 헤더@None |
| 받는 값 | 월 | | 연결 대상 | 관리자 프로그램 |
| 설명 | 확인하고자 하는 날짜의 월을 받아서 매출데이터를 확인해서 전달한다. | | | |

* 1. 예약 프로그램 프로토콜

<표 21> 예약 프로그램 프로토콜 설계서

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 예약 프로그램 | | | | |
| 발신 내용 | inqResTime@clientACode | | 기능 | 인증코드 발송 |
| 전달 값 | 코드  유효 | 헤더@입력받은인증코드 | 출력 | 없음 |
| 코드  비유효 | 같음 |
| 받는 값 | 없음 | | 연결 대상 | 메인 서버 |
| 설명 | 예약 프로그램에서 관리자가 예약 조회시, 입력한 인증코드를 메인 서버에 발송한다. | | | |
| 예약 프로그램 | | | | |
| 발신 내용 | reqResList@resDate | | 기능 | 예약희망날짜 발송 |
| 전달 값 | 날짜  유효 | 헤더@예약희망날짜 | 출력 | 없음 |
| 날짜  비유효 | 같음 |
| 받는 값 | 없음 | | 연결 대상 | 메인 서버 |
| 설명 | 예약 프로그램에서 관리자가 예약할 시, 입력한 예약희망날짜를 메인 서버에 발송한다. | | | |
| 예약 프로그램 | | | | |
| 발신 내용 | postRes@resDate@resParkLot@resStart@resEnd | | 기능 | 예약 정보 발송 |
| 전달 값 | 코드  유효 | 헤더@예약날짜@예약공간@예약시작시간@예약종료시간 | 출력 | 없음 |
| 코드  비유효 | 같음 |
| 받는 값 | 없음 | | 연결 대상 | 메인 서버 |
| 설명 | 예약 프로그램에서 관리자가 예약할 시, 입력한 예약정보를 메인 서버에 발송한다. | | | |

* 1. 정산기 프로토콜

<표 22> 정산기 프로토콜 설계서

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 정산기 | | | | | | |
| 발신 내용 | Calculator | | 기능 | | 메인 서버 연결 | |
| 전달 값 | 알림 유효 | 정산기라고 알림 | 출력 | | “서버 접속” | |
| 알림 비유효 | 없음 |
| 받는 값 | 없음 | | 연결 대상 | | 메인 서버 | |
| 설명 | 정산기 프로그램 실행을 위해서 처음에 서버와 연결을 해야 한다. 그러므로 정산기에서는 서버와 연결을 시도한다. 서버는 모듈의 구분을 위해서, 정산기는 정산기 자체인걸 알리기 위해서 서버와 연결 후 데이터를 전송한다. | | | | | |
| 정산기 | | | | | | |
| 발신 내용 | carIn@parking\_num | | 기능 | | 관리자 상태확인 | |
| 전달 값 | 주차 유효 | 헤더@주차공간 | 출력 | | 없음 | |
| 주차  비유효 | 없음 |
| 받는 값 | 없음 | | 연결 대상 | | 메인 서버 | |
| 설명 | 주차장 공간 관리 프로그램으로부터 받은 데이터 값(주차장 자리)은 관리자가 현재상태 확인을 위해, 정산기가 메인 서버에게 전달 한다. | | | | | |
| 정산기 | | | | | | |
| 발신 내용 | parkingFee@parking\_num@money | | | 기능 | | 정산 관리 |
| 전달 값 | 요금 유효 | 헤더@주차공간@요금 | | 출력 | | ‘정산 완료 하였습니다. 감사합니다’ |
| 요금  비유효 | 없음 | |
| 받는 값 | 없음 | | | 연결 대상 | | 메인 서버 |
| 설명 | 정산해야하는 요금과 실질적인 요금이 일치할 경우 정산기는 서버에게 정산 관련된 정보(해당 주차장 자리, 정산된 요금)를 보내준다. 이 정보로 관리자는 매출 상태를 확인할 수 있다. | | | | | |
| 정산기 | | | | | | |
| 발신 내용 | parking\_num | | 기능 | | 출차 알림 | |
| 전달 값 | 출차  유효 | 주차공간 | 출력 | | 없음 | |
| 출차  비유효 | 없음 |
| 받는 값 | 없음 | | 연결 대상 | | 주차공간 관리 프로그램 | |
| 설명 | 정산기는 서버에게 정산완료 되었다는 정보를 보내준 후에, 해당 주차장 자리를 다시 주차공간 관리 프로그램에게 보낸다. 이 정보를 가지고 해당 주차장을 사용하는 사용자는 출차를 수행할 수 있다. | | | | | |

* 1. 주차장 공간 관리 프로토콜

<표 23> 주차장 공간 관리 프로토콜

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 주차장 공간 관리 | | | | |
| 발신 내용 | True | | 기능 | 정산기 연결 |
| 전달 값 | 연결  유효 | 정산기 접속 | 출력 | 없음 |
| 연결 비유효 |  |
| 받는 값 | 없음 | | 연결 대상 | 정산기 |
| 설명 | 주차장 공간 관리 프로그램 실행을 위해 정산기 서버에 연결을 해야한다. | | | |
| 주차장 공간 관리 | | | | |
| 발신 내용 | parking\_num | | 기능 | 입차(플랩 상승) |
| 전달 값 | 입차  유효 | 해당 주차자리에 대한 주차 여부 | 출력 | 없음 |
| 입차  비유효 |  |
| 받는 값 | 없음 | | 연결 대상 | 정산기 |
| 설명 | 입차가 완료되면 플랩이 상승하여, 해당 주차장 자리를 정산기에 전달을 한다. 이 값이 정산기에 들어가야만이 정산기는 시간과 요금계산을 할 수 있다. | | | |

* 1. 주차타워 프로토콜

<표 24> 주차타워 프로토콜 설계서

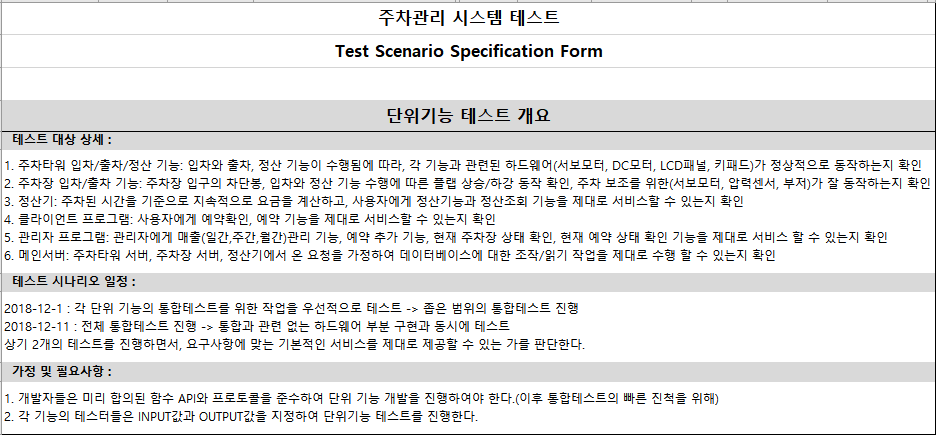
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 주차타워 | | | | |
| 발신 내용 | checkACode@certifi | | 기능 | 인증코드 전달 |
| 전달 값 | 코드  유효 | 헤더@인증코드 | 출력 | 없음 |
| 코드  비유효 | 없음 |
| 받는 값 | 없음 | | 연결 대상 | 메인 서버 |
| 설명 | 사용자가 키패드를 통해 인증코드를 입력하면 주차타워는 메인 서버에 전달을 한다. | | | |

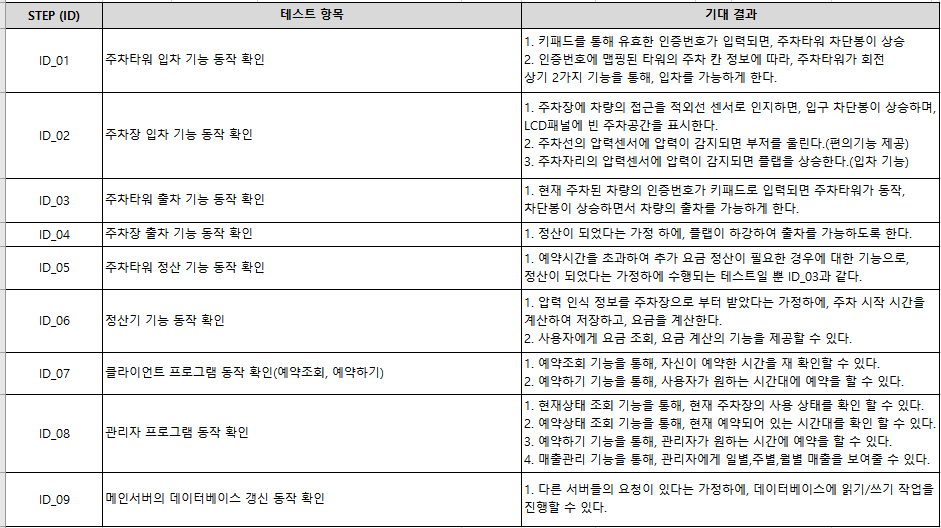
* 1. 관리자 프로토콜

<표 25> 관리자 프로토콜 설계서

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 관리자 프로그램 | | | | |
| 발신 내용 | reqNonResList | | 기능 | 비예약주차장 상태 불러오기 |
| 전달 값 | 없음 | | 출력 | 없음 |
| 받는 값 | NonResList@NonResParkingLot | | 연결 대상 | 메인 서버 |
| 설명 | 관리자가 비예약주차장의 현재 주차상태를 확인하기 위하여, 서버에 데이터를 요청한다. | | | |
| 관리자 프로그램 | | | | |
| 발신 내용 | reqEarningList@resDate | | 기능 | 해당 날짜의 일별 매출기록 불러오기 |
| 전달 값 | 날짜  유효 | 헤더@매출확인날짜 | 출력 | 없음 |
| 날짜  비유효 | 헤더@None |
| 받는 값 | EarningList@earningList | | 연결 대상 | 메인 서버 |
| 설명 | 관리자가 확인하고자 하는 월과 일을 입력하여 서버로부터 해당 날짜의 매출기록을 전송받는다. | | | |
| 관리자 프로그램 | | | | |
| 발신 내용 | reqEarningList\_month@seeingMonth | | 기능 | 해당 날짜의 월별 매출기록 불러오기 |
| 전달 값 | 날짜  유효 | 헤더@해당 월 | 출력 | 없음 |
| 날짜  비유효 | 헤더@None |
| 받는 값 | EarningList\_month@earningList\_month | | 연결 대상 | 메인 서버 |
| 설명 | 관리자가 확인하고자 하는 월을 입력하여 서버로부터 해당 월의 매출기록을 전송받는다. | | | |
| 관리자 프로그램 | | | | |
| 발신 내용 | inqResTime@clientACode | | 기능 | 인증코드 발송 |
| 전달 값 | 코드  유효 | 헤더@입력받은인증코드 | 출력 | 없음 |
| 코드  비유효 | 같음 |
| 받는 값 | 없음 | | 연결 대상 | 메인 서버 |
| 설명 | 예약 프로그램에서 관리자가 예약 조회시, 입력한 인증코드를 메인 서버에 발송한다. | | | |
| 관리자 프로그램 | | | | |
| 발신 내용 | reqResList@resDate | | 기능 | 예약희망날짜 발송 |
| 전달 값 | 날짜  유효 | 헤더@예약희망날짜 | 출력 | 없음 |
| 날짜  비유효 | 같음 |
| 받는 값 | 없음 | | 연결 대상 | 메인 서버 |
| 설명 | 예약 프로그램에서 관리자가 예약할 시, 입력한 예약희망날짜를 메인 서버에 발송한다. | | | |
| 관리자 프로그램 | | | | |
| 발신 내용 | postRes@resDate@resParkLot@resStart@resEnd | | 기능 | 예약 정보 발송 |
| 전달 값 | 코드  유효 | 헤더@예약날짜@예약공간@예약시작시간@예약종료시간 | 출력 | 없음 |
| 코드  비유효 | 같음 |
| 받는 값 | 없음 | | 연결 대상 | 메인 서버 |
| 설명 | 예약 프로그램에서 관리자가 예약할 시, 입력한 예약정보를 메인 서버에 발송한다. | | | |
| 관리자 프로그램 | | | | |
| 발신 내용 | delRes@resDate@resParkLot@resStart@resEnd | | 기능 | 예약 정보 삭제 |
| 전달 값 | 코드  유효 | 헤더@예약날짜@예약공간@예약시작시간@예약종료시간 | 출력 | 없음 |
| 코드  비유효 | 같음 |
| 받는 값 | 없음 | | 연결 대상 | 메인 서버 |
| 설명 | 예약 프로그램에서 관리자가 예약을 삭제할 때, 이미 예약되어 있는 정보가 삭제되어 코드 및 예약상태 그리고 매출기록이 삭제된다. | | | |

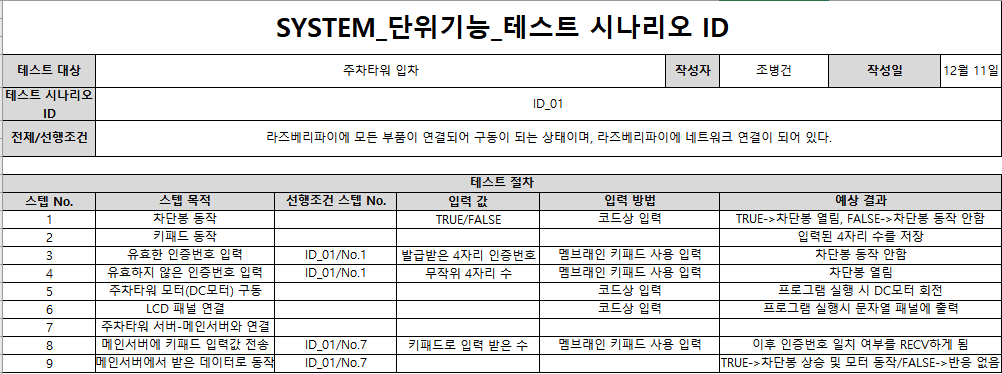
1. **소스 코드 (별도 첨부)**
2. **테스트** 
   1. 단위기능 테스트 <표 26> 단위기능 테스트 개요 표





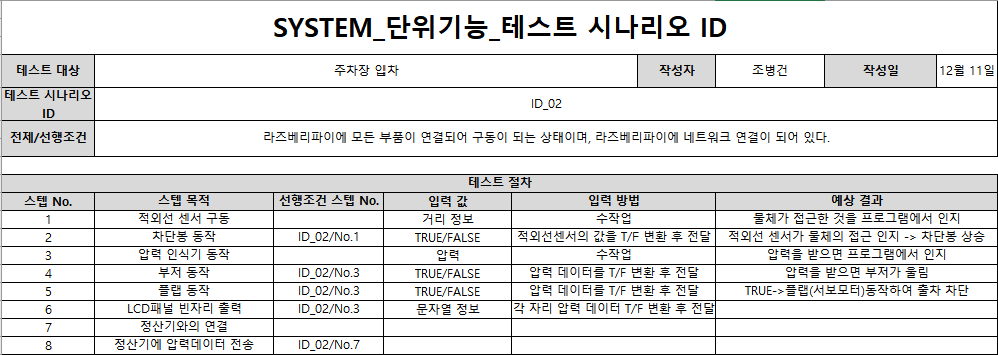
* + 1. 단위기능 테스트 시나리오

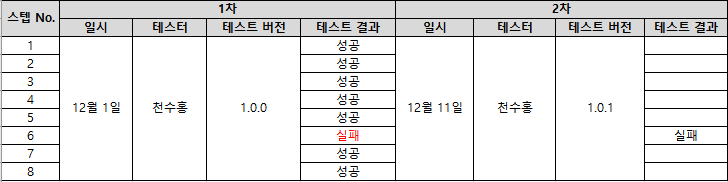
<표 27> 주차타워 입차 기능 테스트 시나리오



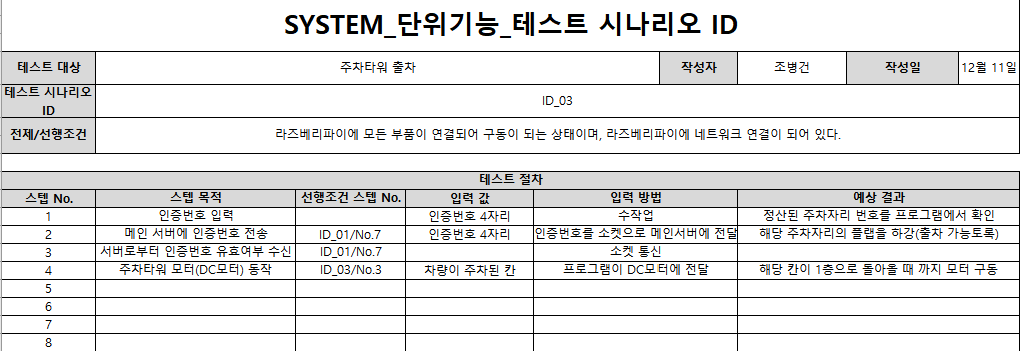


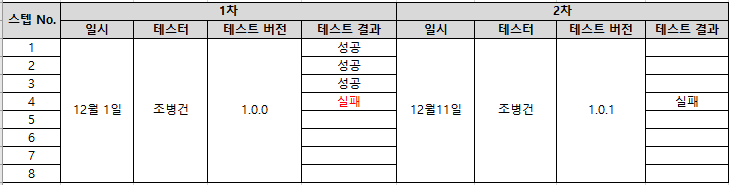
<표 28> 주차장 입차 기능 테스트 시나리오



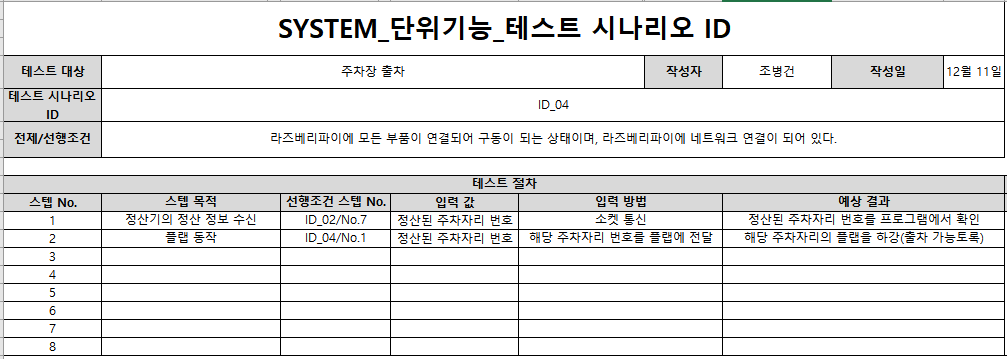


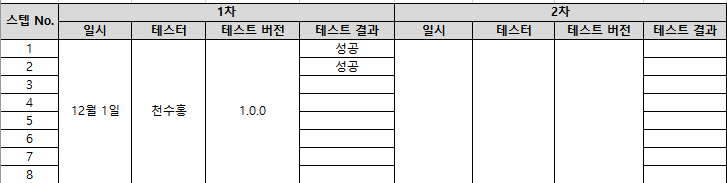
<표 29> 주차타워 출차 기능 테스트 시나리오



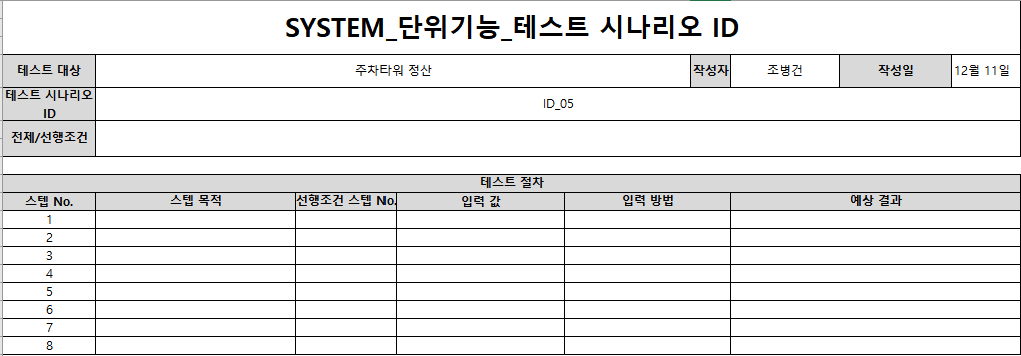


<표 30> 주차장 출차 기능 테스트 시나리오



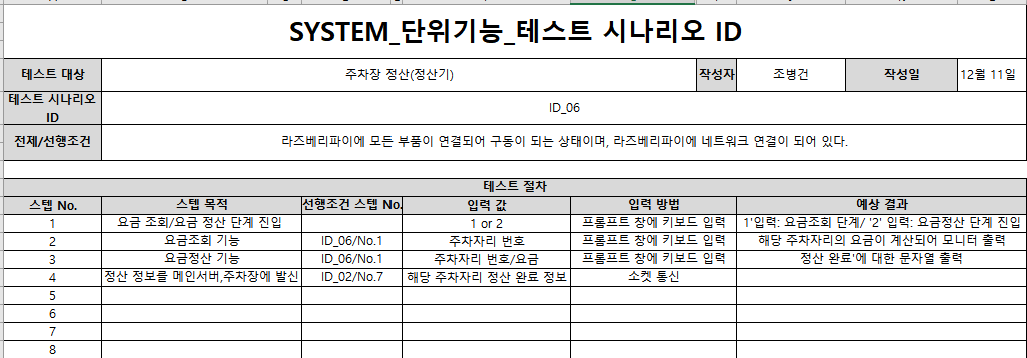


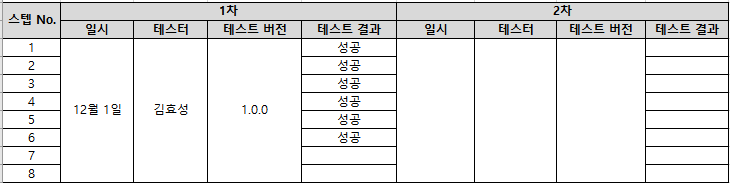
<표 31> 주차타워 정산 기능 테스트 시나리오



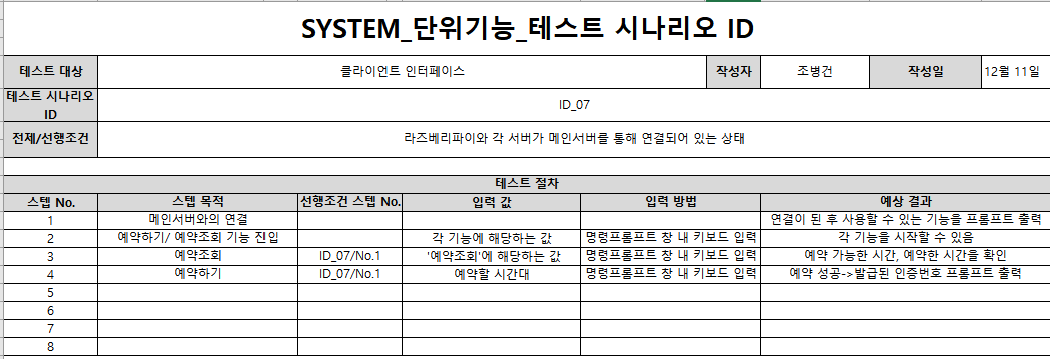
미구현으로 테스트 불가

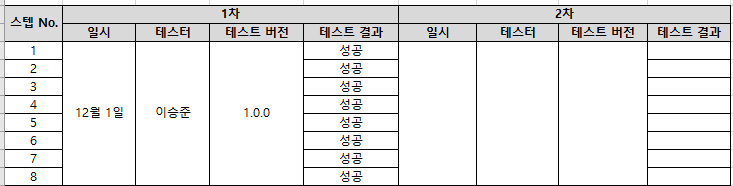
<표 32> 주차장 정산(정산기) 기능 테스트 시나리오



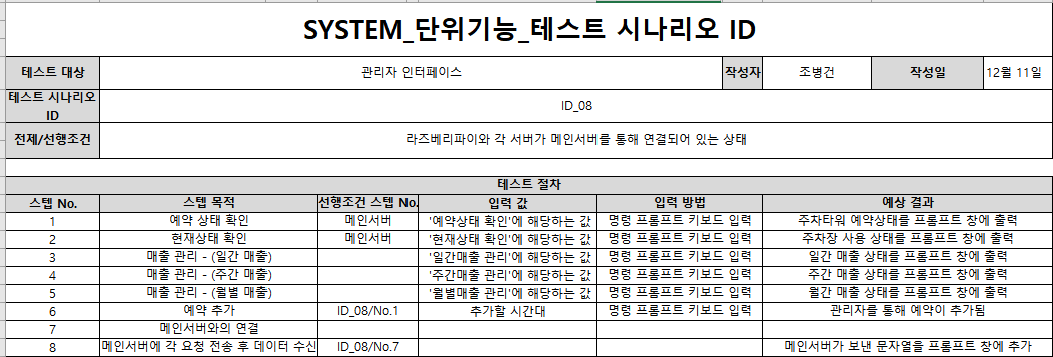


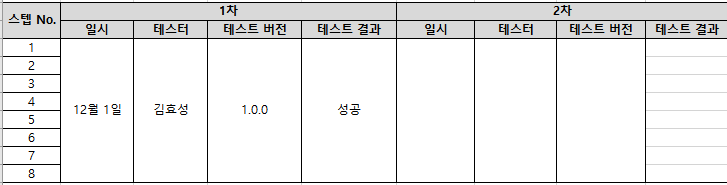
<표 33> 클라이언트 인터페이스 기능 테스트 시나리오



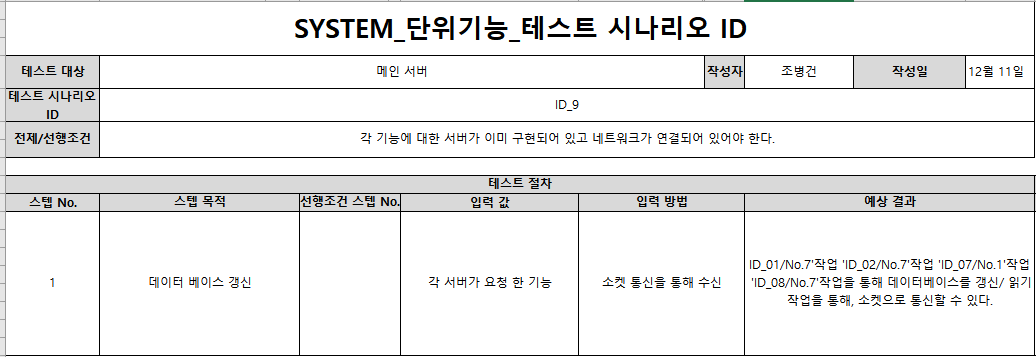


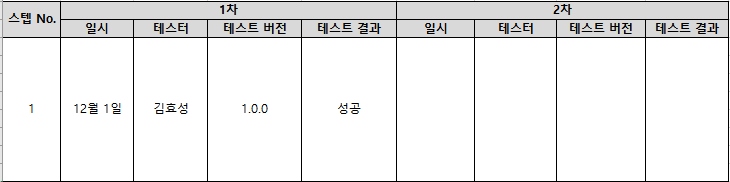
<표 34> 관리자 인터페이스 기능 테스트 시나리오



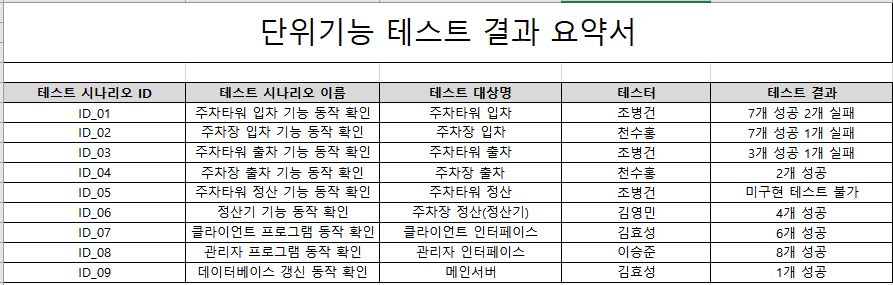


<표 35> 메인 서버 기능 테스트 시나리오

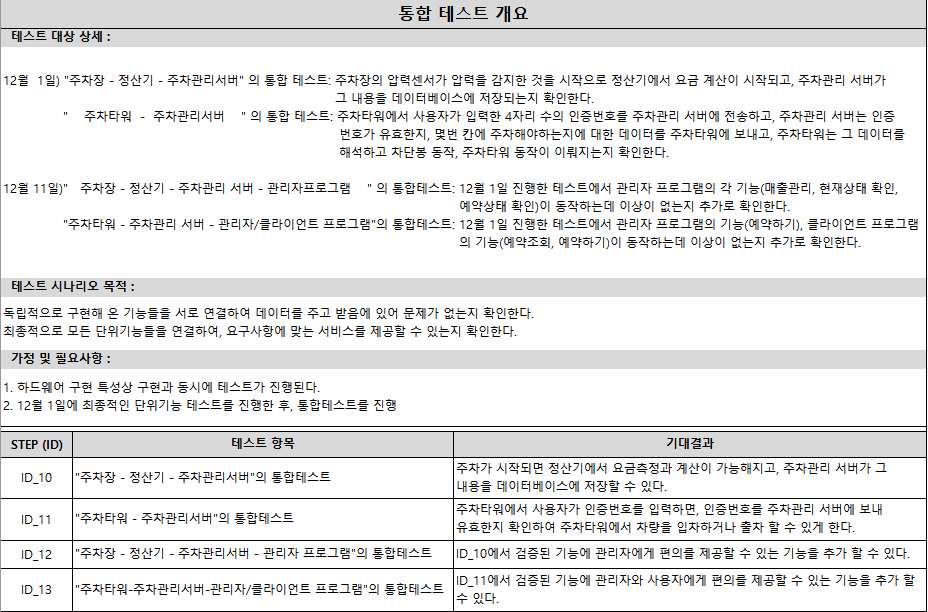




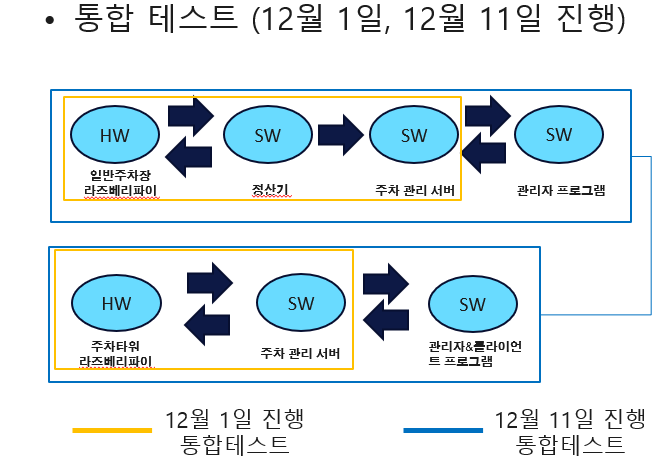
* + 1. 단위기능 테스트 결과 <표 36> 단위기능 테스트 결과 요약서



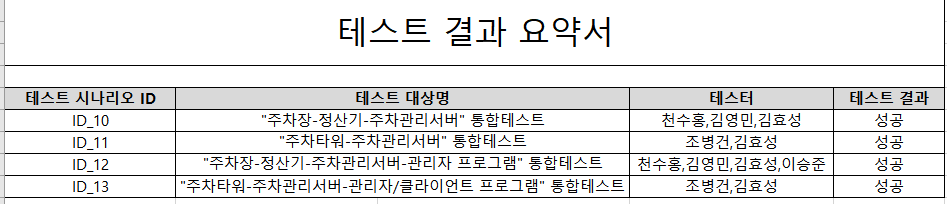
* 1. 통합 테스트 <표 37> 통합 테스트 개요 표



<그림 12> 통합 테스트 일정 요약 그림



* + 1. 통합 테스트 결과 <표 38>통합 테스트 결과 요약서



* 1. 전체 테스트 결과
     1. 단위 테스트 결과

1)주차타워의 인증시스템에서의 LCD패널 출력 실패

2)DC모터 전압 부족으로 주차타워 구동 실패

3)일반 주차장의 주차 가능 자리 표시를 위한 LCD패널 출력 실패

* + 1. 통합테스트 결과

1)모든 기능이 데이터 결합되어, 통합테스트에서 오류 발견이 적었다.

* + 1. 오류 수정 작업

통합테스트)주차관리서버의 멀티쓰레딩이 안되는 문제 해결

단위테스트)주차타워와 일반주차장 서보모터 동작 오류를 해결

1. **프로그램 수행 결과 (별도 동영상 첨부)**
2. **결론**
   1. **업무 변경 내역**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **작업 이름** | **기존 담당자** | **변경 내용** | **변경 된 담당자** | **변경 사유** |
| 예약시스템 구현 | 이승준 | 장고로 구현하려던 예약 시스템을 소켓 통신을 통해 구현하는 것으로 변경 | 김효성 | 개발자의 능력과 경험 부족  -> 일정을 맞추기 위해 변경 |
| 정산기 | 김영민 | 라즈베리파이로 구현하려던 정산기를 소프트웨어로 대체 | 변경 없음 | 라즈베리파이 전원, GPIO 포트 부족의 문제를 해결하지 못함 |
| H/W 구현  (LCD패널) | 조병건, 천수홍 | 하드웨어 구현 포기 |  | 전원은 들어오나 문자열이 출력되지 않는 문제를 해결하지 못함 |
| H/W 구현  (DC 모터) | 조병건 | 하드웨어 구현 포기 |  | DC모터 전압이 낮아 주차타워를 돌리지 못함, 추가 부품 배송을 기다릴 일정 부족 |

* 1. **실패 요인 분석**

1. 개발 능력 부족

클라이언트 예약 관련 되어 웹페이지로 구성하려고 노력을 했으나, 팀원 모두 웹에 대해 지식 부족으로 인해 임계점에 도달했을 때 결국은 소켓 프로그래밍을 이용하여 구현을 했다.

1. 하드웨어
   1. 라즈베리 파이 개수가 한정적이라, 한사람당 하나의 라즈베리 파이를 분담하여 작업 하였다.(난관에 부딪혔을 때 도와줄 사람이 없음) 또한 라즈베리 파이 포트 부족 + 브래드보드에 대한 지식 부족으로 정산기구현을 소프트웨어로 변경 했다.
   2. 다른 부품이 필요하다거나, 호환이 되지 않음을 늦게 발견 했다. 결국 주차타워 구현을 포기하게 되었다.
   3. 라즈베리 부품 연결에 대한 두려움(전압등), 이는 시도 횟수를 감소하게 하여, 개발 일정에 영향을 끼쳤다.
   4. 일정 계획 시 라즈베리파이 부품에 대한 별도의 일정을 고려하지 않았다.

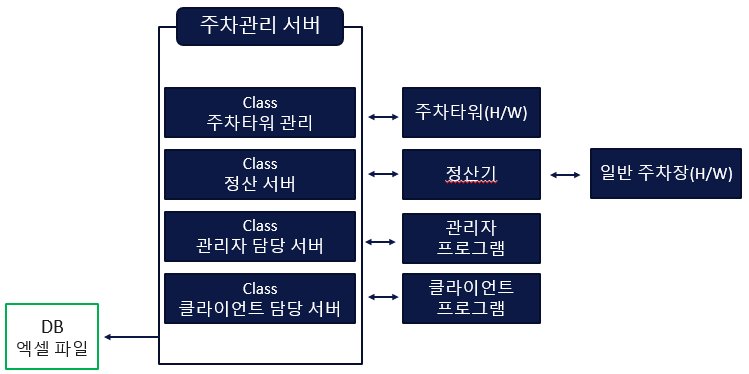
\* 결국 계획단계서 실패한 것이며 위험요인 대응책을 적절히 마련하지 못했다

3) 비용

프로젝트를 위해 25만원을 투자하여 개발을 했으나, 예외 상황으로 부품 부족 문제가 발생하면 추가 비용에 대한 부담이 생김.

1. **최종 업무 분담**

<그림 13> 최종 업무 분담



상기 그림은 최종 시스템 구조도를 나타낸다. 이에 따른 최종 업무 분담은 아래와 같다.

<표 39> 최종 업무 분담

|  |  |
| --- | --- |
| **담당자** | **담당 업무** |
| 김효성 | 1. 주차관리서버에 속한 각 클래스를 동시에 실행하기 위한, 멀티 쓰레딩 구조 설계와 엑셀파일(데이터베이스)의 읽기/쓰기 작업 구현 2. 주차타워 관리 Class 구현 3. 클라이언트 담당 서버 Class 구현 4. 클라이언트 프로그램 구현 |
| 이승준 | 1. 관리자 프로그램 구현 2. 정산 서버 Class 구현 3. 관리자 담당 서버 Class 구현 |
| 김영민 | 1. 정산기 프로그램 구현 |
| 천수홍 | 1. 주차장의 H/W 구현(라즈베리파이) |
| 조병건 | 1. 주차타워의 H/W 구현(라즈베리파이) |

1. **기존 시스템과의 차별성**

1)예약 주차장

기본 주차장에 더해 날짜, 시간별로 예약해서 사용할 수 있는 예약 주차장이 존재한다.

2)예약 프로그램

예약 주차장을 사용하기 위해 사용자는 예약 프로그램을 사용하여 예약을 할 수 있다. 예약 뿐 아니라, 자신이 예약한 시간을 확인할 수 있다.

3)차단(플랩)이 주차 공간마다 존재

기존 주차장은 모두 출구에서 차량이 몰려 혼잡도를 유발한다. 이는 무인 정산기에서 정산을 하는 사용자도 있지만, 하나의 길 밖에 없는 출구에서 정산을 하는 사용자가 존재하여 출구 쪽의 길이 막히고, 이로 인해 입구까지 혼잡을 유발하게 된다. 이러한 문제점을 근본적으로 해결하기 위해 각 주차 공간에서 무인 정산기를 이용하여 정산을 하지 않으면, 해당 주차 공간에서 나갈 수 없도록 한다.

1. **느낀점**
   1. **천수홍**

단순한 프로젝트를 진행할때조차도 팀을 필수적으로 구성해야함을 느꼈습니다. 초기 계획 구성시 단 기간이 걸릴것이라고 예상했던 사항들조차 예상치 못한 변수로 인해 지연이 된다는 사실을 깨달았습니다. 저는 라즈베리를 사용한 하드웨어 부분을 담당하였는데, 초기에 라즈베리가 아두이노와 매우 흡사하다고 생각했던 것에서부터 잘못되었다는걸 라즈베리를 사용하기로 결정한 후, 깨달았습니다. 기존에 라즈베리에 대한 지식이 부족했던점, 특히 하드웨어적인 문제와 라즈베리가 하나의 OS를 통해 동작한다는 점을 배우는 데 까지 긴 시간이 걸렸습니다. 하드웨어쪽에서는 GPIO 핀 연결법과 저항연결법, 가장 걸림돌이 었던 라즈베리 파이 관련 부품 확보였습니다. 하드웨어적인 측면에서의 지연이 다른팀원들의 프로젝트진행에도 영향을 미쳤고, 이는 전체적인 프로젝트의 지연으로 연결되었습니다. 이를 통해 아무리 작은 역할을 분담받은 팀원일지라도, 팀원 한사람 당 모든 역할을 제대로 수행해야만이 프로젝트가 전체적으로 수월하게 진행될 수 있음을 깨달았습니다. 그리고 마지막 문제는 소통의 문제였었습니다. 제가 여기서 말씀드리고자 하는 것은 글자 그대로의 소통이 아닌 정보교환입니다. 팀원간에 서로 교류가 이전에 있었기 때문에 소통에는 별 문제가 없을 것이라고 생각했습니다. 하지만 자신이 직접 맡고 있은 부분이 아니라면, 다른 팀원들에게 설명을 하더라도 제가 전달하고자 하는 바가 모두 전달되지 않을 수있다는 것을 알았습니다. 이에 대한 요인은 프로젝트 전반적인 내용들이 모두 다 처음 접하는 부분이었기 때문이었다고 생각합니다. 이번 학기에 팀프로젝트를 진행함으로써 실무에서의 프로젝트 진행이 대략적으로 어떻게 진행되는 지를 알 수있었으며, 우리가 간과할 수 있는 허점들이 다수 존재하기 때문에, 단순한 프로젝트라 할지라도 세밀한 분석이 필요함을 느꼈습니다.

* 1. **김영민**

이번 프로젝트 진행을 하면서 수업에서 배운 이론과는 정말 똑같이 다양한 위험들이 존재한다는 것을 알게 되었습니다. 처음에는 위험요소가 그렇게나 많을까 생각을 했엇는데, 계획단계부터 테스트 단계까지 각자 일정에 맞춰서 진행 한다는 것 자체가 너무나 어려운 사실이라는 것을 깨닫게 되었습니다. 이번 프로젝트 진행 결과 제일 중요한 것을 팀원들의 합이 얼마나 잘 맞는지가 프로젝트의 성공과 실패를 나눈다는 것을 느꼈습니다. 저희 프로젝트에 대해서 저는 성공이라고 말씀드리고 싶고, 실제로 대부분의 팀원들이 이문동에 살기 때문에 매일매일 만나서 회의를 하며, 같이 밤새면서 하는 그 과정들이 저에게는 너무나도 큰 경험이었습니다 (매우 힘들었습니다….) 좀만 더 시간이 있었고, 비용에 대해서도 좀만 더 여유가 있었더라면 아마 완성을 하지 않았을까 라는 생각을 하게 되었습니다. 이번 프로젝트는 생산된 소프트웨어의 결과로서의 만족이 아닌 저는 그 과정으로서 정말로 만족 합니다. 끝으로 교수님 정말 수고 많으셨습니다. 감사합니다.

* 1. **김효성**

프로젝트 조장을 정할 때, 이번 학기가 널널하다고 생각하였고, 조원들도 친하니 빡빡하게 쪼면서 하면 프로젝트 완성도를 보다 높일 수 있겠다고 생각하여 조장을 맡았습니다. 4명의 조원도 맡은 일을 하는 것이 상당히 힘들었다는 것을 알 수 있었지만, 조장은 그보다 많은 짐을 짊어지고 프로젝트를 수행해야 한다는 것을 깨달았습니다. 먼저 해야할 일을 정의하는 것부터 정의한 해야할 일을 조원들에게 분배하고, 간트 차트에 따라 잘되고 있는 지 체크하는 것, 잘되고 있지 않을 시 조장으로서 대처할 방안을 생각하고 시행하는 것까지.. 프로젝트 시작부터 끝까지 스트레스와 갈등이 만연하였습니다. 5명 모두 이 수업 하나만을 듣는 것이 아니라 모두 바쁜 상태에서 이 프로젝트를 적극적으로 수행하기에 무리라는 것을 알고 있음에도 간트 차트에 따라 진행이 되지 않을 때 받는 스트레스는 예상보다 거대하였습니다. 이것이 1차 스트레스였다면, 2차 스트레스는 이 상황을 조원들에게 간트 차트와 함께 설명할 때, 조원들의 개의치 않는 모습이었습니다. 물론 결과물 발표 날이 임박하였을 때는 모두 급박하였지만, 간트차트가 나온 이후 2~3주 동안은 저 혼자 스트레스를 받는다는 것을 깨달았을 때부터 갈등도 시작되었습니다.

하드웨어를 다룬다는 것 또한 어렵다는 것을 깨달았습니다. 소프트웨어는 오류가 생겼을 때 수정하는 시간이 상당히 짧고 그 자리에서 바로 할 수 있지만, 하드웨어는 오류가 생겼을 때 추가 부품이 필요하거나, 하드웨어가 망가졌을 때, 하드웨어에 대한 지식이 전무할 때 관련 지식을 찾을 수 있는 소스가 적다는 점 등등 애로 사항이 많아서 프로젝트 진행 시에 차질을 많이 빚게 만들었습니다. 이번 프로젝트로부터 하드웨어는 배우고 시작하거나, 아예 다루지 말아야 겠다는 점을 느꼈습니다.

다른 조의 발표를 보면서, 공부를 더 열심히 해야 한다는 것을 느꼈습니다. 우리의 발표 결과물과 다른 조의 결과물을 비교해보면서 우리 조는 어떤 점이 부족하다는 것을 알 수 있었고, 후에 이런 부족한 점을 보완하려면 아직 배우지 않았던 GUI와 같은 도구들을 경험해보고 많이 다뤄봐야 스스로 도움이 많이 될 것 같다고 생각하였습니다.

* 1. **이승준**

처음 조를 짜고 프로젝트 주제를 짤 때, 사실 소프트웨어로 진행하는게 낫지 않을까라는 생각을 했지만, 막상 소프트웨어로 개발을 해본 경험이 있는 친구가 없어서 당황했었습니다. 하지만 하드웨어로 구현해 본 팀원들이 있어서, 소프트웨어쪽을 혼자 담당하고 진행하면 괜찮을 거란 생각을 했습니다. 소프트웨어 개발은 파이썬과 장고 프레임워크를 사용해 개발하려 했었습니다. 재작년에 군대를 가기전에 웹개발을 했었는데, 당시에는 루비 온 레일즈로 개발한 경험이 있었습니다. 루비와 파이썬이 문법과 여러 요소가 비슷하다는 말을 많이 들어서, 쉽게 학습하겠다는 생각을 갖고 개발을 진행했는데, 막상 언어적인 측면에서는 루비와 비슷한 점이 많이 있었지만, 프레임워크 측면에서 기능구현을 할 때 레일즈와 다른 요소가 너무 많아서 사실상 장고웹개발을 처음 배우는 것과 다름이 없었습니다. 그렇다고 해서 다른 친구들과 같이 진행하기에는 각자 맡은 바가 있었기 때문에 웹은 혼자 책임져서 개발하자는 생각을 했던 것 같습니다. 어느정도 장고에 대한 학습을 하고, 예약주차장 CRUD인 예약추가, 예약수정, 예약삭제, 예약확인 기능까지 완료한 상황이 개발진행상의 시간을 조금 오버했던 것 같습니다. 당시에 하드웨어와 장고를 연동하여 기능을 수행하도록 하는 방식 역시 새로 배우고 구현해야했는데, 저는 하드웨어와의 통신에 대해 아는 바가 없었고 다른 친구들 역시 장고에 대해 아는 바가 없어서 이 상태로 웹을 밀고 나가기에는 큰 무리가 따를 것 같다는 생각이 들었습니다. 마침 김효성 학우가 소켓통신으로 구현하는게 어떻겠냐는 의견을 내서 큰 무리를 감수하고 할 바에는 다른 학우들이 다 경험이 있는 소켓통신으로 하는게 좋을 것 같다는 생각을 했습니다. 결과적으로 기능적인 측면에서는 구현을 완료하게 되어서, 웹개발의 속도때문에 불안해하던 팀원에게도 마음이 놓이기도 했고, 소켓개발을 할 때에는 이전 개발을 포기했다는 생각에 밤을 새더라도 더 속도를 내서 개발을 하게 된 것 같습니다.

모든 측면에서 완벽하지도 않고 멋진 결과물을 이뤄냈다고 생각하진 않지만, 좋은 팀원들과 서로 말은 다 안해도 힘든 내색 안하고 서로에게 힘든 얘기 안하고 꾸준하게 할 일을 해준 것이 너무 고맙고 미안한 것 같습니다. 언제나 일을 나눠 하게되는 상황에서는 자기가 한 일이 커 보이고 자랑하고싶고, 남이 한 일은 사소해보이기 마련이지만, 일시적인 팀원이기 이전에 원래 알던 친구들이기도 했고 서로 존중하는 마음이 깔려있어서 다들 남탓없이 되려 자기 일보다 더 많은 일을 하고 다른 팀원이 해야할 일까지도 부담없이 해줬던 것 같습니다. 이번 프로젝트를 통해서 장고와 소켓통신에 대해 많이 알게되었고, 문서작성에 대해서도 다시 한 번 생각해 보게 된 것 같습니다. 이번 프로젝트를 통해서 분업이든 혼자 개발을 함에 있어서의 마음가짐을 다시 한 번 생각해 볼 수 있었고, 어려움을 느끼면서도 막상 시도하면 어떤 게 가능한지도 알게 된 것 같습니다.

* 1. **조병건**

처음 프로젝트 과제를 부여 받았을 때 두 달이면 넉넉히 프로젝트를 완성하고도 남을 시간이라 여겨, 여유를 부렸습니다. 그 여유와 함께, 프로젝트 계획 시 아두이노에 대한 기초 지식만 알고 있음에도 불구하고 라즈베리파이로 프로젝트를 진행하자는 객기를 부리기도 하였습니다. 또 그 동안 융복합소프트웨어 수업에서 과제를 순조롭게 해왔기 때문에, 이번에도 순조로울 것이라는 생각을 가지고, 하드웨어 지식이 전혀 없는 상태로 두 달 안에 모든 것을 끝내겠다고 자신하였습니다. 그러나 당시에 하드웨어 구현을 위해서는 수 많은 사전 작업이 필요하고, 단순히 어떤 카테고리의 부품이 필요한지만 아는 것이 아니라, 모델명에 대한 세세한 정보까지 알아야 한다는 것을 몰랐습니다. 그렇게 저의 오만함 때문에 프로젝트가 밀려갔고, 처음으로 학부 수업의 과제에서 제 능력 부족으로 목표한 것을 끝내지 못하게 되었습니다.

그러면서도 다른 팀들 또한 우리와 상황이 같을 것이라 자위를 했습니다. 그러나 다른 팀들의 발표를 보면서 그들은 적당히 타협을 하며 개발이라고 할 수 있는 것들을 해왔지만 저희는 그렇지 못한 것 같습니다. 프로젝트 계획부터 체계적이어야 하였으며, 개발 일정 또한 합리적으로 수립해야 했던 것은 물론, 각 업무의 인원을 분배하는 것 또한 중요했던 것임을 알면서도 가볍게 생각했던 것입니다.

저는 그동안 학부 수업만을 들으면서 혼자 진행할 수 있는 얕은 과제만을 수행하고, 3학년 수업을 들어오면서 성적이 잘 나왔다는 것만으로 만족하며 우물 안 개구리가 되어 있음을 발견했습니다. 개발에 있어서 개발자의 학부 성적 보다 개발 능력과 경험이 우선 된다는 것을 깨달을 수 있었고, 저의 오만에 대한 반성을 할 수 있게 된 수업이었습니다.

1. **발표자료 (별도 첨부)**
2. **활동 사진**

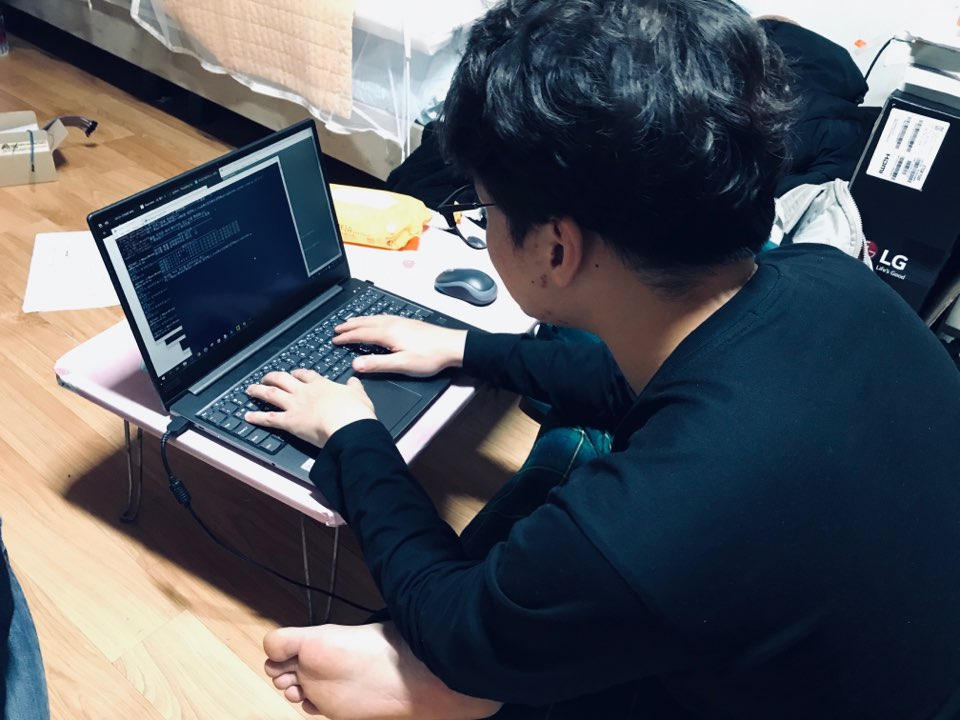
**<최종 보고서 작업>**

****

**<통합 테스트>**

****

****

****

****