**REPORT**

**음식점 매니지먼트 시스템 (RMS)**



|  |  |
| --- | --- |
| **제출일자** | 2022년 05월 10일 (화) |
| **교과목** | 소프트웨어설계공학 |
| **담당교수** | 장희숙 교수님 |
| **학과** | 컴퓨터소프트웨어공학과 |
| **팀 이름** | 짹짹이 |
| **이름(학번)** | 조진혁 (20203120) 이수찬 (20172128)  최인수 (20173236)  박상현 (20183145) |
| **학년** | 3학년 |

목차

|  |  |  |
| --- | --- | --- |
| 1. **개요** | ------------------------------------------------------------ | p.01 |
| **1.1 프로젝트 개요** | ---------------------------------------------- | p.01 |
| **1.2 적용 패턴** | ---------------------------------------------- | p.02 |
| **1.3 정의, 약어** | ---------------------------------------------- | p.02 |
| 1. **자원 및 일정 예측** | ------------------------------------------------------------ | p.03 |
| **2.1 비용 (LOC)** | ---------------------------------------------- | p.03 |
| **2.2 일정** | ---------------------------------------------- | p.03 |
| **2.3 인력** | ---------------------------------------------- | p.04 |
| 1. **조직 구성 및 인력 배치** | ------------------------------------------------------------ | p.05 |
| **3.1 조직 구성** | ---------------------------------------------- | p.05 |
| 1. **WBS** | ------------------------------------------------------------ | p.06 |
| **4.1 WBS 차트** | ---------------------------------------------- | p.07 |
| **4.2 작업 목록 리스트** | ---------------------------------------------- | p.07 |
| **4.3 CPM 네트워크** | ---------------------------------------------- | p.08 |
| **4.4 임계 경로 및 임계치** | ---------------------------------------------- | p.08 |
| **4.5 간트 차트** | ---------------------------------------------- | p.09 |

|  |  |  |
| --- | --- | --- |
| 1. **기술 관리 방법** | ------------------------------------------------------------ | p.10 |
| **5.1 변경 관리** | ---------------------------------------------- | p.10 |
| **5.2 위험 관리** | ---------------------------------------------- | p.10 |
| **5.3 비용 및 진도 관리** | ---------------------------------------------- | p.11 |
| 1. **표준 및 개발 절차** | ------------------------------------------------------------ | p.13 |
| **6.1 개발 방법론** | ---------------------------------------------- | p.13 |
| 1. **검토 회의** | ------------------------------------------------------------ | p.14 |
| **7.1 개발 주요 환경** | ---------------------------------------------- | p.14 |
| **7.2 주요 개발 회의 장소** | ---------------------------------------------- | p.14 |
| **7.3 개발 도구** | ---------------------------------------------- | p.14 |
| 1. **개발 환경** | ------------------------------------------------------------ | p.15 |
| **8.1 단위 테스트** | ---------------------------------------------- | p.15 |
| **8.2 통합 테스트** | ---------------------------------------------- | p.15 |
| **8.3 시스템 테스트** | ---------------------------------------------- | p.15 |
| 1. **참고문헌 및 부록** | ------------------------------------------------------------ | p.16 |

|  |  |
| --- | --- |
| **1** | **개요** |

|  |  |
| --- | --- |
| **1.1** | **프로젝트 개요** |

* **배달의 민족, 요기요 등 여러 플랫폼을 하나로 관리하는 동시에 홀 주문 관리 및 매장 관리**
  + 요즘 음식점들의 거의 배달의 민족, 요기요 서비스를 사용하여 주문을 관리하고 있다.
  + 실제 각각의 서비스마다 다른 주문 환경을 가지고 있어, 다른 태블릿 PC를 설치하고 사용하여 주문받고 있다.
  + 각기 다른 서비스 환경을 하나로 합치면 보다 수월한 주문 관리를 할 수 있을 것 같다는 생각하였다.
* **음식점에서 일하는 직원들의 근태, 월급 관리할 수 있다.**
  + 단순히 매장 내의 포장 주문, 홀 주문만 관리하는 시스템이 아닌, 매장에서 일하는 직원들의 근무 시간, 근무 날짜 등을 기록할 수 있는 서비스를 만들어서 관리하고자 한다.
* **음식점에 필요한 재고 관리를 할 수 있다.**
  + 많은 매장에서 필요한 재고를 Excel를 활용하여 관리하며, 매주 특정한 날짜를 정하여 하나하나 확인해보고 관리하고 있다는 것을 알게 되었다.
  + 직접 눈으로 확인하다 보면 사람인지라, 누구나 실수하고 잘못된 재고 관리를 할 수 있는데, 이러한 문제점을 해결하기 위해 재고 관리에 대한 시스템도 함께 만들고자 한다.

|  |  |
| --- | --- |
| **1.2** | **적용 패턴** |

|  |  |  |
| --- | --- | --- |
| 작업 목록 | **적용 패턴** | **설명** |
| 주문 입력 | 커맨드 패턴 | 커맨드 패턴을 활용하여, 주문 완료를 입력하면, 실제 주문이 주방에 들어갈 수 있도록 한다. |
| 주문한 영수증 관리 | 옵저버 패턴 | 각 매장 테이블에 대하여, 손님들의 영수증을 출력하기 위하여 옵저버 패턴을 활용한다. 매장 시스템에서 손님 테이블로 영수증을 보낸다. |
| 결제 관리 | 스트래티지 패턴 | 카드 결제, 현금 결제 등 다양한 결제 방법에 대하여 각기 다른 할인율을 적용하기 위해서 스트래티지 패턴을 사용한다. |
| 추가 주문 | 데코레이터 패턴 | 매장에서 식사하시는 손님이 여러 주문을 넣기 위해서, 데코레이터 패턴을 사용하여, 주문한 음식에 대한 추가 옵션에 대해서 확장 가능하게 설계한다. |
| 주방 및 주문 관리 | 싱글턴 패턴 | 매장 시스템을 운영하기 위한 각각의 클래스 객체를 싱글턴 패턴을 활용하여, 인스턴스를 하나만 사용하여 관리할 수 있도록 한다. |

|  |  |
| --- | --- |
| **1.3** | **정의, 약어** |

|  |  |
| --- | --- |
| **정의 & 약어** | **설명** |
| **POS 시스템** | Point of Sale의 약어로, 비즈니스 사업에 필요한 재정 및 관리 작업을 처리해준다. |
| **동백전** | 소상공인과 시민, 전통시장이 함께 소비의 선순환을 만들어 주는 부산지역화폐 |
| **누비전** | 소상공인의 매출증대와 지역소비를 통한 지역경제 활성화를 위한 창원지역화폐 |
| **CRM** | Customer Relationship Management의 약어로, 고객과의 관계를 관리하는  소프트웨어 시스템 |
| **SKU** | Stock Keeping Unit의 약어로, 재고를 관리하는 데에 있어서 필요한 데이터를  판별하는 코드 |

|  |  |
| --- | --- |
| **2** | **자원 및 일정 예측** |

|  |  |
| --- | --- |
| **2.1** | **비용 (LOC)** |

* **개발 비용 산정**

|  |
| --- |
| **LOC 산정 계산식** |
| * **노력 (M/M)**   + 참여인원 x 개발 기간 * **개발비용**   + 노력 x 단위비용(1인당 월평균 인건비) |

|  |
| --- |
| **LOC 산정 과정** |
| 1. **노력 산정**    1. **4(M/M) = 4 x 1** 2. **개발 비용 산정**    1. 개발 비용 = 4 x (최저시급(9,160원\*5시간 (근무시간)) x (일주일 :5일) x (한달 4주기준))    2. 3,664,000원 = (4 x (45,800 x 5 x 4)) |

|  |
| --- |
| **LOC 산정 후 최종 개발 비용** |
| * **최종 개발 비용 (4인 개발자 기준)**   + **총 3,664,000원** |

|  |  |
| --- | --- |
| **2.2** | **일정** |

* **더욱 체계적인 프로젝트 진행을 위해 일정 계획을 세분화한다**
  + **WBS**
    - 4.1 WBS의 구조도 참조
  + **CPM 네트워크 차트**
    - 4.3 네트워크 차트 참조
  + **임계 경로 및 임계치**
    - **임계 경로** : C → D → B → F → H → I → J → K → M → N
    - **임계치** : 총 19일 예정
  + **간트 차트**
    - 4.5 간트 차트 참조

|  |  |
| --- | --- |
| **2.3** | **인력** |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **인력 별 개발 추진 계획(안)** | | | | | | | |
| **이름** | **작업 설명** | **적용 패턴** | **총 작업수** | **이름** | **작업 설명** | **적용 패턴** | **총 작업수** |
| **최인수** | 주문관리 | 데코레이터 패턴 | **3개** | **조진혁** | 결제 | 스트래티지 패턴 | **1개** |
| 메뉴추가 |  |
| 메뉴삭제 |  |
| **박상현** | 주문 관리 | 옵저버 패턴 | **3개** | **이수찬** | 결제 방식별 매출 출력 | 싱글턴  패턴 | **3개** |
| 주문 출력 | 현금 잔액 관리 |
| 영수증 | 상품권 잔액 관리 |

|  |  |
| --- | --- |
| **3** | **조직 구성 및 인력 배치** |

|  |  |
| --- | --- |
| **3.1** | **조직 구성** |

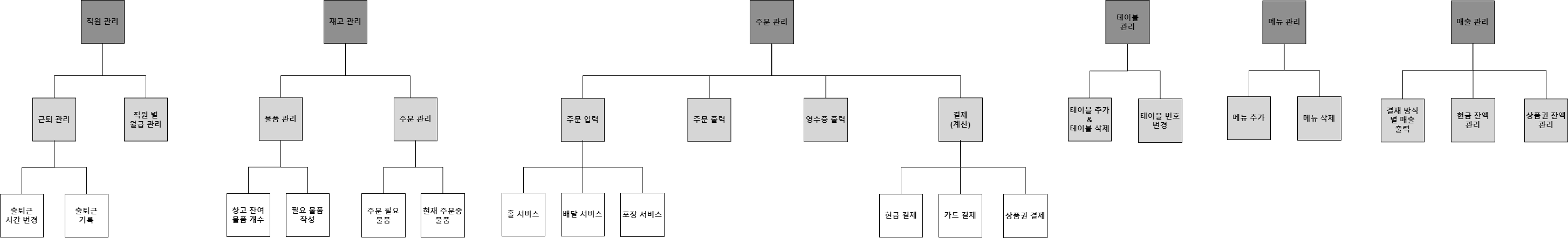
Diagram

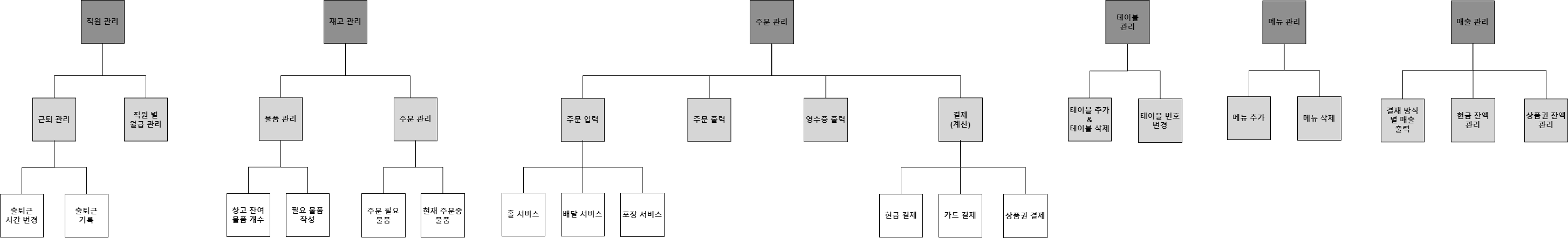
Description automatically generated

* **프로젝트 관리자**
  + 프로젝트의 전반적인 부분을 관리. 일정, 요구사항, 개발 비용 등을 관련 관리자들과 토론/의견교환 등을 통해 조율하여 프로젝트 전체를 제어한다.
* **일정 관리자**
  + 프로젝트의 전체적인 일정을 관리한다. 프로젝트의 지연/잉여 시간 관리, 병렬 작업 수행 등에서의 책임을 진다.
* **요구사항 관리자**
  + 프로젝트의 사용자 요구 사항 관리, 사용자와의 협의 등을 담당한다. 프로젝트 내 요구사항 변경이 있을 시 일정 관리자와 협의하여 새로운 일정을 이끌어내거나 기존 일정에 변경을 가해야 한다.
* **개발 비용 관리자**
  + 프로젝트 진행 중의 개발 비용/필요한 리소스 관리를 담당한다.

|  |  |
| --- | --- |
| 4 | **WBS** |

|  |  |
| --- | --- |
| **4.1** | **WBS 차트** |

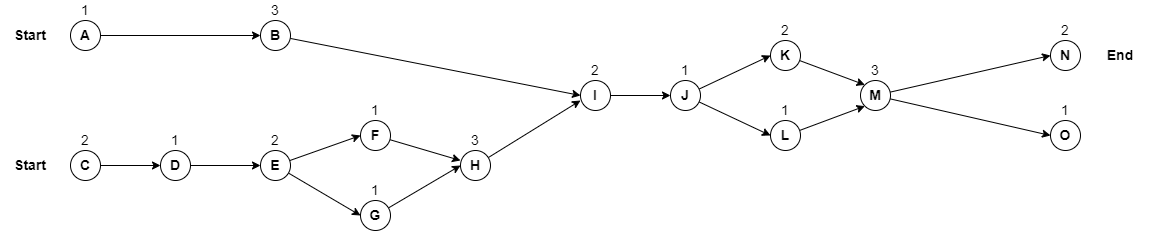




|  |  |
| --- | --- |
| **4.2** | **작업 목록 리스트** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **작업** | **작업 설명** | **선행 작업** | **소요 기간(일)** | **적용 패턴** |
| A | 근퇴 관리 | - | 1 | 미정 |
| B | 직원 별 월급 관리 | A | 3 | 미정 |
| C | 물품 관리 | - | 2 | 미정 |
| D | 주문 관리 | C | 1 | 데코레이터  옵저버 패턴 |
| E | 주문 입력 | D | 2 | 커맨드 패턴 |
| F | 주문 출력 | E | 1 | 옵저버 패턴 |
| G | 영수증 | E | 1 | 옵저버 패턴 |
| H | 결제 | F, G | 3 | 스트래티지 패턴 |
| I | 테이블 추가 & 삭제 | D, H | 2 | 미정 |
| J | 테이블 번호 변경 | I | 1 | 미정 |
| K | 메뉴 추가 | J | 2 | 데코레이터 패턴 |
| L | 메뉴 삭제 | J | 1 | 데코레이터 패턴 |
| M | 결재 방식 별 매출 출력 | K, L | 3 | 싱글턴 패턴 |
| N | 현금 잔액 관리 | M | 2 | 싱글턴 패턴 |
| O | 상품권 잔액 관리 | M | 1 | 싱글턴 패턴 |

|  |  |
| --- | --- |
| **4.3** | **CPM 네트워크** |



|  |  |
| --- | --- |
| **4.4** | **임계 경로 및 임계치** |

* **임계 경로**
  + C → D → E → F → H → I → J → K → M → N
* **임계치**
  + 총 19일

|  |  |
| --- | --- |
| **4.5** | **간트 차트** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **작업명** | **시작일** | **기간(주)** | **5월** | | | **6월** | |
| **2** | **3** | **4** | **1** | **2** |
| 1 | 직원 관리 | 22.05.09 | 1 |  |  |  |  |  |
| 2 | 재고 관리 | 22.05.09 | 1 |  |  |  |  |  |
| 3 | 주문 관리 | 22.05.16 | 2 |  |  |  |  |  |
| 4 | 테이블 관리 | 22.05.23 | 1 |  |  |  |  |  |
| 5 | 메뉴 관리 | 22.05.30 | 1 |  |  |  |  |  |
| 6 | 매출 관리 | 22.06.06 | 1 |  |  |  |  |  |

|  |  |
| --- | --- |
| 5 | **기술 관리 방법** |

|  |  |
| --- | --- |
| **5.1** | **변경 관리** |

* 프로젝트의 진행/유지보수 중 변경될 수 있는 부분에 관련한 내용으로, 다음과 같다.

1. **결재 서비스**
   1. 새로운 결재 서비스 제공자 추가/기존의 결재 서비스 제공자의 종료에 따른 결재 관리 시스템 추가/제거
2. **영수증 출력 기능**
   1. 영수증 형식의 변경이 있을 경우 기존 소스코드의 변경이 필요하다
3. **메뉴 관리**
   1. 메뉴에 추가적인 정보(알레르기 등)를 제공해야 할 때 클래스 field의 변경이 있을 수 있다.
4. **주문 관리**
   1. 프로그램 실행 환경에 따라 주문이 데이터베이스에 저장될 수도, 메모리에 저장될 수도 있으며 저장 매체 또한 개발 중 변경될 수 있다.

|  |  |
| --- | --- |
| **5.2** | **위험 관리** |

* 프로젝트 진행/유지보수 중 일어날 수 있는 위험 요소와 이 위험요소들의 대응 방안으로, 다음과 같다.
  + **개발 기간 부족** (프로젝트 개발에서의 내/외부 요소에 의한 개발 지연)
    - **위험 발생 가능성** : 보통 (25%~50%)
    - **원인** : 타 프로젝트와의 동시 진행으로 인한 일정 관리 실패, 외부 라이브러리의 에러에 의한 개발 지연 등
    - **해결 방법** : 요구사항 조정/일정 재검토 등을 통해 마감일에 위반되지 않도록 조치한다.
  + **팀원 간 불화** (팀원 간의 의견 충돌/불화에 의한 위기)
    - **위험 발생 가능성** : 매우 낮음 (0%~10%)
    - **원인** : 프로젝트 진행에 대한 팀원 간의 의견 충돌, 사적인 감정 격화 등 팀원과의 관계가 악화될 수 있는 요소
    - **해결 방법** : 적당한 정도로 팀원간 교류 장려, 한 팀원에게 과도하게 업무가 집중되는 것을 합리적인 일정 계획을 통해 방지
  + **요구사항 변경** (기존에 정의된 요구사항에 추가적으로 요구사항이 증가)
    - **위험 발생 가능성** : 높음 (50%~75%)
    - **원인** : 클라이언트, 팀원 내부 회의 등으로 인한 부족한/추가적으로 필요한 요구 사항의 추가
    - **해결 방법** : 요구사항 변경에 덜 위험한 개발 방법론 사용

|  |  |
| --- | --- |
| **5.3** | **비용 및 진도 관리** |

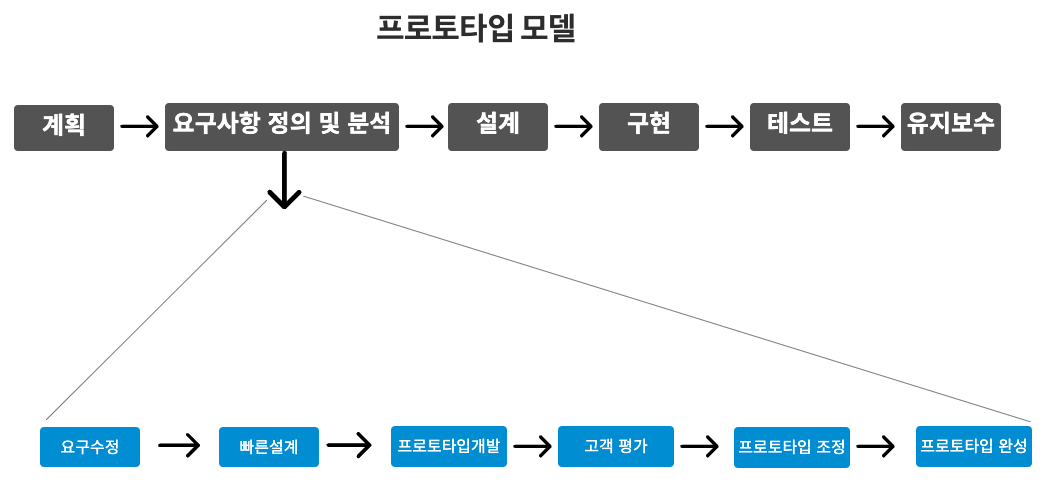
* 프로젝트에 사용될 금액적/시간적 비용과 그 진행도를 나타낸다.
  + COCOMO II 방법에 의한 개발 비용 예측 (간이 기능 점수법 이용(KOSA 사업대가 산정 Excel File)

|  |
| --- |
| **간이 FP 산정표** |
| Table  Description automatically generated |

|  |
| --- |
| **산정된 FP의 집계** |
|  |

|  |  |
| --- | --- |
| 6 | **표준 및 개발 절차** |

|  |  |
| --- | --- |
| **6.1** | **개발 방법론** |



* **진화적 프로세스 모델**
  + 반복적인 프로토타입을 사용자에게 보여주어, 개발 중에 원활한 대화가 가능하다
  + 제품 완료가 아닌 프로토타입은 전문자가 아닌 사용자는 그대로 받아들이는 단점이 있어 주의가 필요하다

|  |  |
| --- | --- |
| 7 | **개발 환경** |

|  |  |
| --- | --- |
| **7.1** | **개발 주요 환경** |

* **IDE**
  + IntelliJ IDEA, NetBeans
* **OS 환경**
  + Windows, MacOS

|  |  |
| --- | --- |
| **7.2** | **주요 개발 회의 장소** |

|  |  |
| --- | --- |
| **대면 회의** | **비대면 회의** |
|  |  |

|  |  |
| --- | --- |
| **7.3** | **개발 도구** |

|  |  |
| --- | --- |
| **Github** | **MySQL** |
|  |  |

|  |  |
| --- | --- |
| **8** | **성능 시험 방법** |

|  |  |
| --- | --- |
| **8.1** | **단위 테스트** |

* **각자 개발 환경(IDE)에서 정상적으로 작동하는지 테스트 한다**
  + VSCode(박상현), Eclipse(조진혁), IntelliJ IDEA(최인수), NetBeans(이수찬)
* **OS별 환경에서 정상적으로 작동하는지 테스트 한다**
  + Windows(박상현, 조진혁, 이수찬), MacOS(최인수)

|  |  |
| --- | --- |
| **8.2** | **통합 테스트** |

* **매 회의 마다 코드 리뷰를 진행한다**
  + GitHub의 Pull Request를 사용할 예정

|  |  |
| --- | --- |
| **8.3** | **시스템 테스트** |

* **OS별 환경에서 정상적으로 작동하는지 테스트 한다**
  + Windows(박상현, 조진혁, 이수찬), MacOS(최인수)

|  |  |
| --- | --- |
| **9** | **참고문헌 및 부록** |

1. [쉽게 배우는 소프트웨어 공학. 김치수. 2021](http://www.kyobobook.co.kr/product/detailViewKor.laf?mallGb=KOR&ejkGb=KOR&barcode=9791156645429)
2. [Head First Design Patterns: 스토리가 있는 패턴 학습법. 에릭 프리먼. 2005](https://www.hanbit.co.kr/store/books/look.php?p_code=B9860513241)
3. [ECOMMERCE PLATFORMS](https://ecommerce-platforms.com/)
4. [부산 지역화폐 동백전](https://busandong100.kr/)
5. [창원사랑상품권(누비전)](https://www.changwon.go.kr/biz/contents.do?mId=0605020100)
6. [Pull Request를 통해 코드리뷰(Code Review)하는 법. Gyun’s 개발일지. 2020](https://devlog-wjdrbs96.tistory.com/231)
7. [Loyverse POS System](https://loyverse.com)
8. [KOSA 사업지원 – SW업 대가산정 가이드(2022 개정판)](https://www.sw.or.kr/site/sw/ex/board/View.do?cbIdx=276&bcIdx=51935&searchExt1=)
9. [KOSA 사업지원 - SW사업대가 산정 방식별 엑셀 템플릿 (2022 개정판)](https://www.sw.or.kr/site/sw/ex/board/View.do?cbIdx=276&bcIdx=49636&searchExt1=)