|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 소프트웨어공학: 프로젝트 제안서 | | | | | | | |
| **기술개발**  **과 제** | AI 기반 미스터 대박 디너 서비스 구축  AI-based Mr. Daebak Dinner Service | | | | | | |
| **과제팀 이름** | 물리와 컴퓨터 | | | 지도교수 | | 이병정 교수님 | |
| **개발기간** | 2025년 9월 ～ 2025년 12월 (총 3 개월) | | | | | | |
| **개발소요비용** | 총 액 | 17,370 (천원) | | | | | |
| **과제팀**  **구성원** | 이름 | 장성우 | 최완재 | |  | |  |
| 학과 | 컴퓨터과학부 | 물리학과 | |  | |  |
| 학번 | 2020920054 | 2023550029 | |  | |  |
| 연락처 | 01092480869 | 01065494312 | |  | |  |
| **소프트웨어 프로젝트 과제를 성실히 수행하고자 과제 제안서를 제출합니다.**  2025년 9월 15일  과제 수행자1: 장성우 (인)  과제 수행자2: 최완재 (인) | | | | | | | |

|  |
| --- |
| **1. 개발 과제의 개요** |

**가. 개발 과제 요약**

미스터 대박 서비스는 사용자가 웹사이트, 또는 모바일 앱을 통해 특별한 날에 맞춤 디너 서비스를 주문하고, 직원이 이를 준비하여 배달하는 시스템이다. 고객은 주문 시 서빙 스타일을 선택할 수 있으며, 주문 후에도 메뉴 수정이 가능하다. 해당 시스템은 고객용 인터페이스와 직원용 관리 인터페이스로 구분되어 각각에게 적절한 서비스를 제공한다.

**나. 개발 과제의 배경 및 효과**

외식 대신 본인의 집에서 프리미엄 디너를 제공받고자 하는 고객의 증가

기존 배달 서비스와 차별화 된 자유로운 메뉴 선택과 유연한 메뉴 변경/맞춤 옵션의 수요

본 서비스는 고객의 특별한 기념일을 타겟으로 한 서비스로, 개인에게 맞춤형 경험을 제공

기업 차원에서 충성 고객 확보와 프로모션을 통해 매출 증대 기대

**다. 개발 과제의 목표와 내용**

목표 : 고객의 특별한 기념일에 제공하는 맞춤 디너 배달 서비스 제공

주요 내용

1. 회원가입/로그인 기능, 고객의 데이터베이스 관리

2. 주문 인터페이스 : GUI 기반 + 음성 인식 기능

3. 주문 수정 기능(자유로운 추가/삭제/변경)

4. 직원용 인터페이스 : 조리와 배달을 구분

5. 주문 기록 저장 및 재주문 가능

6. 할인/프로모션 시스템

**라. 개발 과제의 산출물**

**-분석산출물**

**-설계산출물**

**-구현산출물**

**마. 용어 정의**

|  |  |  |  |
| --- | --- | --- | --- |
| **용어** | **정의** | **범주** | **비고(제약/규칙)** |
| 고객(Customer) | 회원가입 후 로그인하여 주문하는 사용자 | 참여자 | 성명/주소/연락처 저장, 단골 할인율 보유 가능 |
| 단골 고객 | 누적 주문·정책에 의해 할인율이 적용되는 고객 | 정책 | 할인율 0~50% 내 관리 |
| 주소(Address) | 배달 목적지 정보 | 데이터 | 도로명/상세/도시/우편번호 등 |
| 디너(Dinner) | 미리 구성된 코스(메뉴 묶음) | 제품 | 발렌타인/프렌치/잉글리시/샴페인 축제 등 |
| 메뉴 아이템(MenuItem) | 디너를 구성하는 개별 항목(음식/음료/소품) | 제품 | 이름/타입/수량/단위 포함 |
| 서빙 스타일(Serving Style) | 제공 방식(Simple/Grand/Deluxe) | 옵션 | 스타일에 따라 가격 가중치 다름 |
| 샴페인 축제 디너 | 2인 기준의 프리미엄 디너 | 제품 | **Simple 불가, Grand/Deluxe만** |
| 추가 메뉴(Add-on) | 기본 구성 외에 고객이 추가하는 항목 | 옵션 | 예: 바게트 추가, 샴페인 1→2병 |
| 제거(Removal) | 기본 구성에서 제외하는 항목 | 옵션 | 예: 커피 제외 |
| 수량(Quantity) | 디너 주문 수 또는 인원 수 | 주문 | 1 이상 정수 |
| 주문(Order) | 고객이 특정 시각·주소로 디너 제공을 요청한 행위 | 주문 | 주문 아이템들의 집합, 총액/상태 보유 |
| 주문 아이템(OrderItem) | “디너 + 스타일 + 수량 + 커스터마이즈”의 단위 | 주문 | 한 주문 내 다수 포함 가능 |
| 예약 시간(Scheduled Time) | 배달/도착을 목표로 하는 시각 | 주문 | 30분 단위 권장(과제 정의에 맞춤) |
| 배달 주소 | 주문의 배송 목적지 | 주문 | 고객 기본 주소와 다를 수 있음 |
| 재주문(Reorder) | 이전 주문 이력 기반의 빠른 재주문 | 편의 | 최근 주문 목록에서 선택 |
| 견적(Quote) | 현재 구성 기준 예상 결제 금액 | 결제 | 스타일/추가/할인 반영 |
| 결제(Payment) | 카드 등으로 대금 지불/승인·정산 처리 | 결제 | 승인(Authorize) → 정산(Capture) |
| 취소/환불 | 주문 또는 결제 취소 절차 | 결제 | 상태·정책에 따라 부분 환불 가능 |
| 조리(Preparation) | 주방에서 디너를 준비하는 활동 | 운영 | 재고 확인 후 시작 |
| 배달(Delivery) | 배달원이 고객에게 전달하는 활동 | 운영 | 상태 전이: 픽업→이동중→완료 |
| 배달원(Courier) | 배달 담당 직원 | 참여자 | 주문 단위로 배정 |
| 요리사(Cook) | 조리 담당 직원 | 참여자 | 재고 소진/시간 계획 |
| 재고(Inventory) | 메뉴를 구성하는 원재료 보유 수량 | 운영 | 차감/보충 관리 |
| 재고 보충일(Restock Day) | 공급사로부터 재료를 보충하는 기준일 | 운영 | 주 2회(문제 기술에 근거) |
| 프로모션/할인 | 기간/조건별 가격 인하 정책 | 정책 | 쿠폰/단골 할인 등 |
| 관리자/매니저 | 예약 승인·스케줄 배정 권한자 | 운영 | 과제 설명의 “승인” 주체 |
| **용어** | **정의** | **범주** | **비고** |
| GUI | 그래픽 기반 주문 인터페이스(웹/앱) | 프런트 | HTML/CSS/JS, Android 프로토타입 포함 |
| 음성 인식(STT) | 음성을 텍스트로 변환하여 주문을 처리 | AI | Google Speech-to-Text 사용 |
| API | 모듈 간 호출 규약(REST 등) | 연동 | 결제/음성/서버 간 |
| DBMS | 데이터베이스 관리 시스템 | 인프라 | MySQL |
| 주문 이력 | 고객의 과거 주문 저장 레코드 | 데이터 | 재주문/프로모션 근거 |
| 응답 시간 | 주문 입력 후 결과 화면까지 걸린 시간 | 품질 | 목표 3초 이내(기준 5초) |
| 데이터 처리 정확도 | CRUD 결과가 사양대로 저장·조회되는 비율 | 품질 | 목표 98% |
| 음성 인식 정확도 | 샘플 기준 올바른 인식 비율 | 품질 | 목표 90% |
| GitHub Org | 버전 관리·PR·이슈 기반 협업 공간 | 협업 | 브랜치 전략/코드리뷰 |
| PR(Pull Request) | 변경사항 검토·병합 요청 | 협업 | 2인 상호 리뷰 필수 |
| Issue | 작업/버그/개선 관리 단위 | 협업 | To-Do 보드로 활용 |
| AWS EC2/RDS/S3 | 서버/DB/스토리지 클라우드 자원 | 인프라 | 배포/백업 |
| 로그/모니터링 | 상태/오류 기록 및 관찰 | 운영 | 오류 원인 추적 |
| 테스트 | 유닛/통합/시나리오 테스트 | 품질 | 50회 연속 주문 안정성 검증 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **코드** | **권장 정수** | **의미** | **주요 전이(→)** | **트리거/비고** |
| CREATED | 0 | 장바구니/초안 | → CONFIRMED, → CANCELED | 결제 승인 성공 시 확정 |
| CONFIRMED | 1 | 결제 승인 완료 | → PREPARING, → CANCELED | 주방 착수 가능 |
| PREPARING | 2 | 조리 중 | → OUT\_FOR\_DELIVERY | 조리 완료/포장 완료 |
| OUT\_FOR\_DELIVERY | 3 | 배달 출발 | → DELIVERED, → CANCELED | 고객 부재 등 특별 사유 시 취소 가능(정책) |
| DELIVERED | 4 | 고객 수령 완료 | (종단) | 정산/적립 처리 |
| CANCELED | 5 | 취소 | (종단) | 환불/재고 복구 처리 |
| **코드** | **권장 정수** | **의미** | **주요 전이(→)** | **트리거/비고** |
| READY | 0 | 픽업 대기 | → PICKED\_UP | 포장 완료 |
| PICKED\_UP | 1 | 매장에서 수령 | → EN\_ROUTE | 배달 시작 |
| EN\_ROUTE | 2 | 이동 중 | → DELIVERED, → FAILED | 수령/실패 결과 기록 |
| DELIVERED | 3 | 배송 완료 | (종단) | 서명/사진 등 증빙 |
| FAILED | 4 | 배송 실패 | (종단) | 주소 오류/부재 등, 후속 조치 필요 |
| **코드** | **권장 정수** | **의미** | **주요 전이(→)** | **트리거/비고** |
| INIT | 0 | 결제 준비 | → AUTHORIZED, → FAILED | 결제 시도 |
| AUTHORIZED | 1 | 승인(금액 홀드) | → CAPTURED, → CANCELED | 주문 확정 시 정산 |
| CAPTURED | 2 | 정산(청구) | (종단) | 매출 인식 |
| FAILED | 3 | 실패 | (종단) | 한도/오류 |
| CANCELED | 4 | 취소(승인 취소/환불) | (종단) | 정책에 따라 전액/부분 |

|  |  |  |  |
| --- | --- | --- | --- |
| **코드** | **권장 정수** | **의미** | **비고** |
| SIMPLE | 0 | 플라스틱 접시/컵, 종이 냅킨 | **샴페인 축제 디너는 불가** |
| GRAND | 1 | 도자기 접시/컵, 면 냅킨 | 와인 잔 플라스틱 |
| DELUXE | 2 | 꽃병 + 도자기/린넨 | 와인 잔 유리 |
| **코드** | **권장 정수** | **의미** | **비고** |
| FOOD | 0 | 음식류 | 가격 규칙: 기본 단가/가산 |
| DRINK | 1 | 음료류 | 샴페인/와인 포함 |
| ACCESSORY | 2 | 소품/식기/장식 | 스타일 구성품 등 |

|  |
| --- |
| **2. 완료작품의 평가방법** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **평 가 항 목** | **평 가 방 법** | **적용기준** | **개 발**  **목표치** | **비중**  **(%)** |
| **1. 응답 시간** | 주문 입력 후 결과 화면 표시 시간 측정 | 5초 이내 응답 | 3초  이내 | 30 |
| **2. 데이터 처리 정확도** | 주문 내역 DB 저장 후 불러오기 검증 | 95% 이상 정상 처리 | 98% | 25 |
| **3. 음성 인식 정확도** | 샘플 음성 테스트 | 80% 이상 정확도 | 90% | 25 |
| **5. 시스템 안정성** | 50회 연속 주문 | 오류 발생률 5% 이하 | 2% 이하 | 15 |

|  |
| --- |
| **3. 개발일정 및 추진체계** |

**가. 개발 일정**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **단계별 세부개발 내용** | **담당자** | **개발기간 (월단위)** | | | | **비 고** |
| **9** | **10** | **11** | **12** |
| 프로젝트 팀 구성 및 역할 분담 | 팀 |  |  |  |  |  |
| 프로젝트 제안서 준비 | 팀 |  |  |  |  |  |
| 요구 분석 및 UML 모델링 | 장성우 |  |  |  |  |  |
| DB 설계 및 구축 | 최완재 |  |  |  |  |  |
| GUI 주문 인터페이스 구현 | 장성우 |  |  |  |  |  |
| 중간 발표 준비 및 보고서 제출 | 팀 |  |  |  |  |  |
| 음성 인식 주문 기능 구현 | 장성우 |  |  |  |  |  |
| 직원용 인터페이스 구현 | 최완재 |  |  |  |  |  |
| 통합 테스트 | 팀 |  |  |  |  |  |
| 최종 발표 준비 | 팀 |  |  |  |  |  |

**나. 구성원 및 추진체계**

장성우

- 요구사항 분석 및 UML 모델링

- 음성 인식 기능 구현

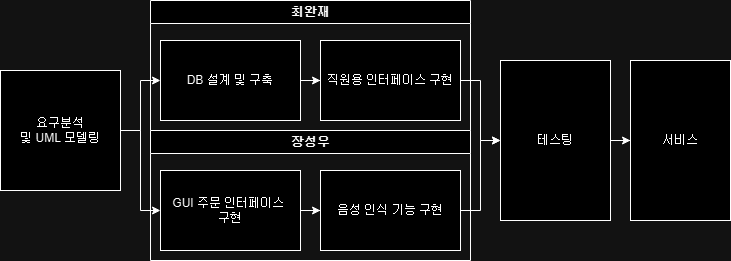
- GUI 주문 인터페이스 개발

최완재

- DB 설계 및 구축

- 직원용 인터페이스 개발

- 최종 통합 및 품질 관리



OOP / 클래스 스켈레톤

com.mrdinner

├─ app

│ └─ Main.java

├─ domain

│ ├─ common

│ │ ├─ Money.java

│ │ └─ Address.java

│ ├─ customer

│ │ └─ Customer.java

│ ├─ menu

│ │ ├─ ServingStyle.java

│ │ ├─ MenuItem.java

│ │ ├─ ItemType.java

│ │ ├─ Dinner.java (abstract)

│ │ ├─ ValentineDinner.java (concrete)

│ │ ├─ FrenchDinner.java (concrete)

│ │ ├─ EnglishDinner.java (concrete)

│ │ └─ ChampagneFeastDinner.java (concrete, 스타일 제약)

│ ├─ order

│ │ ├─ Order.java

│ │ ├─ OrderItem.java

│ │ └─ OrderStatus.java

│ ├─ staff

│ │ ├─ Staff.java (abstract)

│ │ ├─ Cook.java

│ │ └─ Courier.java

│ ├─ delivery

│ │ ├─ Delivery.java

│ │ └─ DeliveryStatus.java

│ ├─ payment

│ │ ├─ Payment.java

│ │ ├─ PaymentStatus.java

│ │ └─ PaymentMethod.java

│ └─ inventory

│ ├─ StockItem.java

│ └─ Inventory.java

└─ service

├─ PricingService.java

├─ OrderService.java

├─ InventoryService.java

└─ DeliveryService.java

링크

<https://gitmind.com/app/docs/m128fzog>

|  |
| --- |
| **4. 위험 분석** |

|  |  |  |
| --- | --- | --- |
| 위험 | 심각성 | 해결방법 |
| 팀원 건강 문제 | 심각함 | 역할 분담 시 중복 커버 가능하도록 설계 |
| 시험 기간 겹침 | 심각함 | 시험기간 전후로 작업량 조절, 마일스톤 미리 앞당겨 준비 |
| 장비 고장 | 심각함 | 클라우드/깃허브를 통한 코드 백업 |
| 과제/알바 등 외부 일정 | 심각함 | 팀 내 작업 우선순위 조율 |
| 팀원간 커뮤니케이션 부족 | 중 | 정기회의(온라인/오프라인) |

|  |
| --- |
| **5. 개발 환경** |

**가. SW 명세**

**개발 언어**

**Java, JavaScript**

**프레임워크 & 툴**

**Flask / Django (웹 서버 구축)**

**Android Studio (모바일 앱 프로토타입)**

**HTML/CSS/JavaScript (GUI 웹 인터페이스)**

**DBMS**

**MySQL (고객/주문 데이터 저장)**

**음성 인식 API**

**Google Speech-to-Text API (음성 주문 처리)**

**UML 모델링 툴**

**StarUML (분석/설계)**

**협업 툴**

**GitHub (버전 관리)**

**Notion (문서 관리)**

**나. HW 명세**

**개발용 PC**

**CPU : Intel i5 이상**

**RAM :16GB 이상**

**Storage : SSD 512GB 이상**

**OS : Windows 10 / Ubuntu 20.04**

**테스트 장비**

**Android 스마트폰 (테스트 용)**

**서버 환경**

**AWS EC2 (배포용)**

|  |
| --- |
| **6. 개발사업비 산정내역서** |

( 단위 : 천원 )

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **항 목**  **(품명, 규격)** | | **수 량** | **단 가** | **금 액** | | | **비 고** |
| **계** | **현금** |  |
| 직접  개발비  (인건비  포함) | 초급 개발자 인건비 | 2 | 2,800,000 | 16,800,000 | 16,800,000 |  | KOSA  2025 초급 단가 기준 |
| 개발용 노트북 | 2 | 1,000,000 | 2,000,000 | 0 |  | 보유 장비 |
| 테스트 스마트폰 | 1 | 50,000 | 50,000 | 0 |  | 보유 장비 |
| 마이크/헤드셋 | 1 | 50,000 | 50,000 | 0 |  | 보유 장비 |
| GitHub Private Repo | 2\*2 | 10,000 | 20,000 | 20,000 |  | 버전 관리 |
| Google STT API |  | 20,000 | 20,000 | 20,000 |  | 무료  크래딧  초과 시 |
| AWS EC2 | 1 | 80,000 | 240,000 | 240,000 |  | 3개월 분 |
| AWS RDS(MySQL) | 1 | 60,000 | 180,000 | 180,000 |  | 3개월 분 |
| AWS S3 | 1 | 30,000 | 90,000 | 90,000 |  | 3개월 분 |
| 도메인 | 1 | 20,000 | 20,000 | 20,000 |  | 가비아  GoDaddy |
| 합 계 |  |  | 19,470,000 | 17,370,000 |  |  |

|  |
| --- |
| **7. 보고 및 모니터링 체계** |

1. 모니터링 체계

* GitHub Organization 저장소
  + 코드와 문서를 모두 공유
  + Branch 단위 분업: 팀원 A는 feature-A, 팀원 B는 feature-B 브랜치에서 작업
  + Commit 기록으로 각자 작업량과 진척 상황 모니터링
  + Issue 기능으로 남은 할 일(To-Do), 버그, 개선사항 관리
* 상호 모니터링
  + 서로의 PR(Pull Request)을 반드시 확인 & 리뷰 후 병합
  + → 팀원 수가 적으므로 “1명 개발 → 1명 검토” 식으로 품질 관리 강화

2. 보고 체계

* 카카오톡 그룹방
  + 팀원 2명이므로 보고 간소화:
    - “작업 완료 → GitHub Push → 카톡으로 완료 메시지”
    - 예: [9/21] 주문 클래스(Order.java) 기본 구조 완성, PR #3 생성
  + 문제 발생 시 즉시 공유 → 상대방이 바로 확인 가능
* 주간 회의
  + 매주 정해진 시간(예: 주말 저녁) 온라인/오프라인 짧게 미팅
  + GitHub commit 내역 + 카톡 보고 종합해 한 주 진행 상황 요약
  + 역할/일정 재조정

3. 역할 분담 & 흐름 예시

* 팀원 A (예: 백엔드/도메인 설계 담당)
  + 주문(Order), 메뉴(Dinner) 관련 클래스 구현
* 팀원 B (예: 서비스/프론트 담당)
  + GUI, OrderService, DeliveryService 등 구현
* 진행 흐름

팀원 A: Order.java 작업 → Push → PR 생성 → 카톡 보고

팀원 B: PR 리뷰 후 Merge → DeliveryService 개발 착수

주간 회의에서 서로 작업 진행 상황 리뷰 & 다음 주 일정 조정

**미스터 대박 프로젝트**

분석 산출물

Team : 물리와 컴퓨터

최완재 2023550029

장성우 2020920054

최종 문서 변경일 : 2025.09.20

요약

미스터 대박 프로젝트의 분석 산출물을 기술.

-유즈케이스, 액티비티 다이어그램 작성

|  |  |  |  |
| --- | --- | --- | --- |
| 문서 이름 | 미스터 대박 프로잭트\_물리와 컴퓨터\_분석 산출물 | | |
| 버전 | | 변경일 | 설명 |
| 1 | 0 | 2025.09.20 | 유즈케이스, 고객 액티비티 다이어그램 작성 |

표 1. 문서 변경 기록

1. 개요

1.1 목적

본 문서는 “미스터 대박 프로젝트”의 분석 산출물을 기록한다.

미스터 대박 프로젝트의 요구 사항에 따라 외부 기능을 기술한다.

1.2 참고문헌

없음

2. USE-CASE DIAGRAM

스케치, 도표, 그림, 텍스트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

3. ACTIBITY DIAGRAM

3.1 고객

텍스트, 도표, 그림이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

4. 액티비티도 설명

4.1 고객

고객이 주문을 시작할 때부터 예약이 확정되기까지의 액티비티 다이어그램이다. 우선 배달 주소지와 예약 일자를 설정하여 해당 예약 일자에 예약이 모두 차서 예약이 불가능 한지 확인한 후, 배달 소요 시간을 계산하여 근무 시간 내에 배달이 완료될 수 있는지 확인한다. 만일 불가하다면 예약 불가 알람을 띄우고 예약 일자를 다시 설정하도록 한다. 메뉴를 선택한 후 재고를 채울 수 있는 기간이 아닌 경우 남은 재고를 파악한 후 예약 가능 여부를 판단한다. 이때, 남은 재고는 저장된 모든 예약을 수행하고 남을 재고를 뜻한다. 결제 이후 예약 신청을 하고 매니저가 승인할 경우 예약이 확정되며 예약이 저장되고 직원들에게 스케쥴이 배당된다.

-주문

고객이 로그인 이후 이전 주문 목록을 통해 불러오거나 주문 버튼을 눌렀을 경우 띄워질 인터페이스이다.

-배달 주소지 설정

배달이 도착할 배달 도착지를 설정한다

-예약 일자 설정

배달이 도착할 예약일자를 년/월/일/시/분 단위로 입력하며 30분 단위로 예약이 가능하다.

-에약 목록 조회

설정된 예약일자를 확인하고 조리와 배달 스케쥴을 확인한 뒤 예약 가능 여부를 판단한다.

-디너 선택

제시된 4개의 디너 중 원하는 디너를 선택할 수 있는 인터페이스를 제공한다.

-서빙 스타일 및 추가 메뉴 선택

디너를 선택한 후 서빙 스타일과 추가 메뉴 혹은 메뉴 제거를 선택하고 인원 수를 결정한다.

-재고 조회

예약 일자와 선택된 메뉴를 통해 조리 가능 여부를 판단한다.

-결제

계좌이체와 신용카드 결제로 나뉘며 결제가 진행된다.

-예약 신청

선택된 예약 일자와 메뉴로된 예약이 메니저에게 전송된다.

-예약 승인

매니저가 예약 일자, 고객 정보, 메뉴 등을 확인하고 예약을 승인을 결정한다.

-예약 확정

매니저가 예약을 승인하면 예약을 확정한다.

-스케쥴 배당

확정된 예약에 직원들의 스케쥴을 고려한 뒤 스케쥴을 배당한다.

-예약 저장

예약이 확정되면 예약 일자, 고객 정보, 메뉴, 배당된 직원, 결제 수단 등을 저장한다.