|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 소프트웨어공학: 프로젝트 제안서 | | | | | | | | | | |
| **기술개발**  **과 제** | AI 기반(음성인식) 미스터 대박 디너 서비스 | | | | | | | | | |
| **과제팀 이름** | 추앙해 | | | | | 지도교수 | | 교수 이병정 | | |
| **개발기간** | 2022년 9월20일～ 2022년 11월28일 (총 2 개월) | | | | | | | | | |
| **개발소요비용** | 총 액 | | 22,079(천원) | | 학교부담금 | | | | 22,079천원 | |
| 과제팀부담금 | | | | 천원 | |
| **과제팀**  **구성원** | 이름 | 이상민 | | 이승민 | | | 이진수 | | |  |
| 사진 |  | |  | | |  | | |  |
| 학번 | 2018920038 | | 2018920042 | | | 2018920046 | | |  |
| 연락처 | 01033702740 | | 01066275777 | | | 01079198377 | | |  |
| **소프트웨어 프로젝트 과제를 성실히 수행하고자 과제 제안서를 제출합니다.**  2022년 9월 25일  과제 수행자1 : 이상민 (인)  과제 수행자2 : 이승민 (인)  과제 수행자3 : 이진수 (인)  **서울시립대학교 컴퓨터과학부 귀중** | | | | | | | | | | |

|  |
| --- |
| **1. 개발 과제의 개요** |

**가. 개발 과제 요약**

비즈니스와 소프트웨어 요구사항에 맞춰 AI 기반(음성인식) 미스터 대박 배달 서비스를 구축한다. 비즈니스 요구사항은 고객들은 웹 사이트에서 배달시간, 장소, 신용카드번호를 사용하여 메뉴를 선택하여 주문하면 디너가 배달 된다. 이때 디너에 포함되는 음식 항목을 변경 및 삭제가 가능하고 디너와 함께 제공되는 스타일도 고객이 선택할 수 있다. 접수된 주문은 직원이 직원 인터페이스를 통해서 수락한다. 직원의 주문 수락 전까진 주문 내 메뉴 변경이 가능하지만, 직원이 수락한 이후에는 주문 내 메뉴 변경이 불가능하다. 고객은 접수한 주문이 조리 중인지 배달 중인지 확인이 가능하다. 고객은 내 정보 버튼을 클릭하면 고객의 주문 내역 조회를 할 수 있고 고객의 정보를 수정할 수 있다. 이때 고객 정보란 주소와 카드 번호이다. 직원은 직원 인터페이스를 통해 접수된 주문을 수락 또는 거절한다. 주문목록확인 버튼과 재고관리 버튼이 존재하는데 주문목록확인 버튼을 누르면 직원은 수락 전과 수락 된 주문 목록을 각각 확인할 수 있다. 재고 관리 버튼을 클릭하면 수락 완료된 주문에 필요한 재료의 양과 현재 가게에 있는 재료의 재고가 각각 나온다. 이때 가게 재고가 부족하면 직원은 재료 추가 버튼을 통하여 재료를 주문할 수 있다.

**나. 개발 과제의 배경 및 효과**

배경

- 특별한 날에 집에서 의미있게 보낼 순 없을까?

* 좋은 사람들과 근사하게 보내고 싶지만 많은 사람이 있는 곳으로 가긴 싫은 상황, 코로나에 걸리면 안되는 상황에서 의미있는 하루를 보내고 싶은 사람들의 행복을 위해 미스터 대박 디너 서비스를 개발하게됐다.

효과

- 집에서 근사한 저녁을 먹고싶은 고객을 위한 서비스 제공

- 불가피하게 집에 있어야 하는 고객의 만족도 증가

- 음성인식 주문을 통해 편리한 주문시스템 개발

**다. 개발 과제의 목표와 내용**

"특별한 날에 집에서 편안히 보내면서 당신의 남편 아내 엄마 아버지 또는 친구를 감동시켜라" 는 요구 철학에 맞게, 사용자의 요청에 완벽히 부합하는 디너 제공 서비스를 바탕으로 한 매출 극대화를 가장 큰 목표로 한다.

- UML과 API명세를 바탕으로 효율적인 개발이 진행되도록 한다.

- 주기적인 검토와 피드백/테스팅을 통해 프로젝트를 유지/보수한다.

세부 내용은 다음과 같다.

- 고객의 주문과 직원관리를 위한 회원가입/ 로그인 기능을 구현한다.

- 디너 정보와 스타일 정보를 제공한다.

- 추가와 삭제, 변경이 가능한 메뉴 선택 기능을 제공한다.

- 주문시 음성인식 기능을 삽입하여 사용자 친화적인 목표에 부합하도록 한다.

- 고객의 주문을 정확히 파악하여 그에 부합하는 시간과 장소, 배달내용을 산출해 낸다.

- 고객의 정보를 바탕으로 과거 주문 내역 조회가 가능하도록 하고,

이를 바탕으로 단골 고객에겐 할인기능도 제공한다.

- 편성된 직원 수와 음식 재고에 맞게 요리/배달에 맞도록 주문을 접수할 수 있도록 한다.

- 직원은 음식 재고에 맞게 적당량을 추가 주문 할 수 있도록 한다.

- 고객의 주문에 적합한 직원 스케줄링을 할 수 있도록 한다.

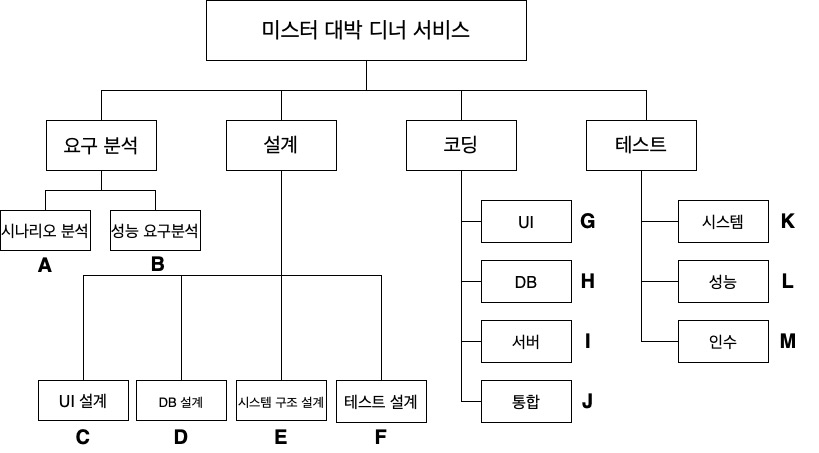
최종적으로 위의 요구사항들을 사용자 친화적인 UI로 구현하여 편리한 디너 서비스를 제공한다.

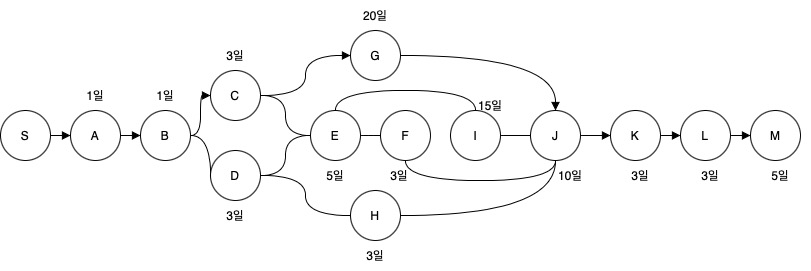
|  |
| --- |
| **2. 완료작품의 평가방법** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **평가 항목** | **평 가 방 법** | **적용기준** | **개 발**  **목표치** | **비중**  **(%)** |
| **로그인** | 1. 등록된 정보를 바탕으로 로그인이 되는가?  2. 직원(요리/배달)과 고객을 구분해서 로그인이 가능한가? | 1. 등록되지 않은 고객은 로그인 불가능한가?  2.로그인시 직원과 고객의 인터페이스가 구별되어 나타나는가? | 적용기준에 완벽히 부합하도록 구현한다. | 10% |
| **회원가입** | 1. 개인의 단일한 정보(주민번호 또는 핸드폰번호) 를 통해 개개인을 식별하여 회원가입 처리 할 수 있는가 [중복을 방지하는가?] | 1. 동일한 정보를 바탕으로 중복 로그인이 되지 않는가? | 적용기준에 완벽히 부합하도록 구현한다. | 10% |
| **음성인식** | 1. 음성을 제대로 인식하는가? | 1. 인식한 음성대로 메뉴 주문을 정확히 받을 수 있는가? | 적용기준에완벽히 부합하도록 구현한다. | 10% |
| **주문** | 1. 주문 내역이 접수되는가?  1-1. 스타일에 맞는 주문인가?  2. 주문 내역의 변경이 가능한가?  2-1. 메뉴추가, 삭제 변경이 가능한가?  2-2. 일정 시간이 지나면 변경을 막을 수 있는가?  3. 직원 수에 맞게 요리/배달 시간을 고려했는가?  4. 음식 재고를 고려했는가?  5. 회원가입을 하지 않는 고객을 위해 비회원 주문이 가능한가? | 1-1. 샴페인 축제 디너 메뉴를 고려했는가?  2-1. 예약 시간 N시간 전까지는 주문 내역 변경이 가능한가?  2-2. N시간 미만으로 남은 경우 그때부터 주문내역은 변경 불가능하게 처리할 수 있는가?  3. 동시간대에 요리 또는 배달 직원을 5명까지만 받을 수 있게 하고 그를 넘어서는 주문은 반려시킬 수 있는가?  4. 재고가 부족한 음식은 주문을 반려시킬 수 있는가?  5. 비회원 주문은 로그인 절차가 없이도 주문이 정상적으로 접수되는가? | 적용기준에 완벽히 부합하도록 구현한다. | 45% |
| **재고관리** | 1. 직원 인터페이스에서 일주일에 두 번 재고 주문이 가능한가?  2. 현 재고 바탕으로 재료 추가/삭제가 잘 되는가? | 1. 일주일에 두 번을 초과하는 재고 주문은 불가능하게 처리할 수 있는가?  2. 재료 추가/삭제시 현 재고량이 UI에 드러나는가? 그리고 추가/삭제한 재고도 반영할 수 있는가? | 적용기준에 완벽히 부합하도록 구현한다. | 10% |
| **주문내역** | 1. 로그인 했을 때 과거 고객 주문 내역이 잘 나오는가? | 1. 과거 주문 내역이 일자별로 정확한 메뉴와 함께 UI에 드러나는가? | 적용기준에 완벽히 부합하도록 구현한다. | 10% |
| **단골고객** | 1. 단골 고객에게 할인 혜택을 제공할 수 있는가? | 1. 일정 주문량을 넘어선 단골고객에게는 시스템 자체에서 할인가를 적용하여 주문을 받을 수 있는가? | 적용기준에 완벽히 부합하도록 구현한다. | 5% |

|  |
| --- |
| **3. 개발일정 및 추진체계** |

**WBS(Work Breakdown Structure)는 다음과 같다.**



**CMP 네트워크를 그려보면 아래의 그림과 같다. 크리티컬 경로는 화살표로 나타내었다. 이를 바탕으로 전체 서비스 개발 일정은 총 46일을 계획하였다**

**A는 이진수, B는 이승민, C는 이상민이 맡았으며, 일정을 표현한 간트 차트는 아래와 같다.**

**텍스트, 건물, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

|  |
| --- |
| **4. 개발 환경** |

1. **SW 명세**

운영체제: Mac OS, Windows

IDE: Visual Studio Code, IntelliJ

Language: Java, JavaScript

Framework: Spring Boot

Library: React.js

협업 툴: Git, Github, Notion

1. **HW 명세**

MacBook Air(M1), Samsung ion 2, 삼성 NT500R5N-X78, AWS EC2, AWS RDS

|  |
| --- |
| **5. 개발사업비 산정내역서** |

( 단위 : 천원 )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **항 목**  **(품명, 규격)** | | **수 량** | **총 금액** | **비 고** |
|
| 직  접 개 발 비 | 인건비 | 3 | 18,000,000 | 1인당 월 300 |
| MacBook Air(M1) | 1 | 1,390,000 |  |
| Samsung ion 2 | 1 | 1,349,990 |  |
| 삼성 NT500R5N-X78 | 1 | 1,339,390 |  |
| AWS EC2 | 1 | 0 | Free Tier |
| AWS RDS | 1 | 0 | Free Tier |
| 중앙도서관 스터디룸 | 1 | 0 | 학교시설 |
| 인문학관 라운지 | 1 | 0 | 학교시설 |
| 합 계 |  | **22,079,380** |  |

|  |
| --- |
| **6. 위험 관리** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **단계** | **위험** | **발생 가능성** | **영향도** | **대응 방안** |
| 분석/설계 | 분석 및 설계 오류 | 중 | 하 | 요구사항 명세서를 꼼꼼히 읽으며 유즈케이스 명확히 파악 |
| 구현 | 일정 지연 | 상 | 하 | 팀원들의 일정 확인 후 진척사항 관리 |
| 테스트 | 구현 오류 | 중 | 상 | 해당 부분 담당 팀원의 코드 수정 및 진행상황과 결과 공유 |
| 기타 | 시험기간 | 상 | 중 | 프로젝트를 고려하여 평소에 시험공부 |