* 소프트웨어공학 프로젝트 계획서 -

텍스트, 표지판이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |
| --- | --- | --- | --- |
| **과목명** | 소프트웨어공학 | | |
| **학과** | 컴퓨터공학과 | | |
| **담당교수** | 장성진 교수님 | | |
| **분반** | 002분반 | | |
| **팀 번호** | 4조 | | |
| **학번 및 성명** | 팀장 | 20163241 | 조영문 |
| 팀원 | 20163265 | 유하영 |
| 20153182 | 유한규 |
| 20193019 | 이상미 |
| **제출일** | 2021.09.30 | | |

**프로젝트 계획서 목차**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1.** | **프로젝트 개요** | | |  |
| 1.1 | | 프로젝트 목적 | |  |
| 1.2 | | 프로젝트 정의 및 시장분석 | |  |
| 1.2.1 | | | 서점 어플 |  |
| 1.2.2 | | | 시장 분석 |  |
| 1.3 | | 프로젝트 산출물 | |  |
|  | |  | |  |
| **2.** | **프로젝트 주요 일정** | | |  |
|  | |  | |  |
| **3.** | **프로젝트 조직** | | |  |
| 3.1 | | 프로젝트 팀 조직 | |  |
| 3.1.1 | | | 범위 및 역할 |  |
| 3.2 | | 개발 방법론 및 도구 선정 | |  |
|  | |  | |  |
| **4.** | **기술적 프로세스(프로젝트 개발 계획)** | | |  |
| 4.1 | | 개발 생명주기 모델 | | |
| 4.1.1 | | | 프로젝트 생명 주기 |  |
| 4.1.2 | | | 생명주기 |  |
| 4.2 | | 개발 방법론 및 도구 선정 | | |
| 4.2.1 | | | 개발 방법론 | |
| 4.2.2 | | | 객체지향 방법론 절차 | |
|  | | |  |  |
| **5.** | **개발 규모 산정** | | |  |
| 5.1 | | WBS | |  |
| 5.2 | | 상세 일정 | |  |
| 5.3 | | 프로젝트 규모 산정 – 간이 기능 점수법 | |  |
|  | |  | |  |
| **6.** | **마일 스톤 및 일정 계획** | | | |

1. **프로젝트 개요**

* 프로젝트명 : 메뉴 추천 프로그램
* 프로젝트 기간 : 21.09.17 ~ 21.12.10
* 개발 대상 : 메뉴 추천 어플리케이션
* 개발 목적 : JAVA를 이용하여 음식 메뉴를 추천하는 기능의 프로그램을 개발하여 프로젝트에 대한 경험을 쌓고 사용자들이 만족할 수 있도록 한다.
* 구현 환경 : Windows 10, Oracle, JDK 1.8
* 개발 언어 : JAVA
* 개발 도구 : Eclipse
* 팀 구성 : 조영문, 유하영, 유한규, 이상미
  1. 프로젝트 목적

본 프로젝트의 목적은 프로그램 사용자가 보다 편리하게 음식 메뉴를 선정할 수 있게 하는 것에 목적을 두고 있다.   
 사용자는 음식 추천 기능을 통해 쉽고 빠르게 음식 메뉴를 선정할 수 있고 선택한 메뉴를 기준으로 주변 음식점 정보를 확인할 수 있도록 한다. 음식점은 매장 정보를 입력하여 사용자의 편의를 돕고 홍보를 통한 매출 증가에 목적을 둔다.

* 1. 프로젝트 정의 및 시장분석
     1. 메뉴 추천 App

사용자가 어플을 통해 음식 메뉴를 추천 받을 수 있는 App이다. 랜덤으로 메뉴를 추천해주는 기능, 추천된 메뉴로 근처 식당을 추천해 주는 기능 등 앱마다 다양한 기능을 가지고 있다.

* + 1. 시장 분석

21.09.20 기준, 구글 스토어에서 메뉴 추천 어플 조사 결과, <점심 뭐 먹지?>, <뭐 먹을까?>, <그거 말고>, <오늘 뭐 먹지?> 등 여러 어플이 시중에 나와있다.  
  
- 점심 뭐 먹지? : 랜덤 메뉴 추천, 추천된 메뉴로 근처 식당 추천. 어플 오작동 많음. 별점 2.4/5  
- 뭐 먹을까? : 메뉴 랜덤 고르기, 여러 메뉴 중 다중 선택 후 랜덤 추천, 메뉴 월드컵 기능, 질문을 통한 추천. 별점 4.8/5  
- 그거 말고 : 단순 랜덤 추천. 별점 4.6/5  
- 오늘 뭐 먹지? : 내 주변 식당 랜덤 뽑기 기능, 랜덤 메뉴 추천 기능, 선택된 메뉴의 식당 정보 보기, 리스트 추가를 통해 나만의 랜덤 리스트 제작 기능. 별점 4.9/5  
  
 어플 기능에 따라 이용자들의 만족도는 천차만별이다. 만족도가 낮은 어플들은 대부분 메뉴를 단순히 랜덤으로 추천해 주는 기능만을 가지고 있고 어플의 오작동이 많이 있었다. 반면, 만족도가 높은 어플은 다양한 기능을 통해 이용자들의 만족도를 높였다. 우리는 단순 메뉴 랜덤 추천만이 아니라 차별성 있는 기능들을 두어 이용자들의 만족도를 높일 수 있도록 할 것이다.

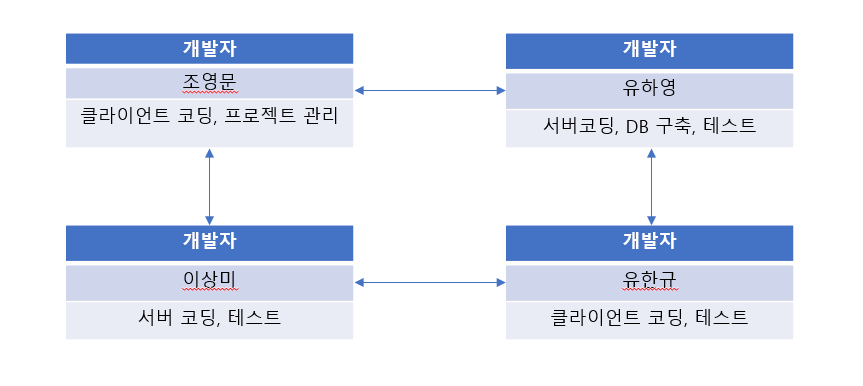
* 1. 프로젝트 산출물  
     - 프로젝트 제안서  
     - 요구사항 명세서  
     - 프로젝트 계획서  
     - 설계 사양서  
     - 소스코드  
     - 테스트 결과 보고서  
     - 프로젝트 완료 보고서

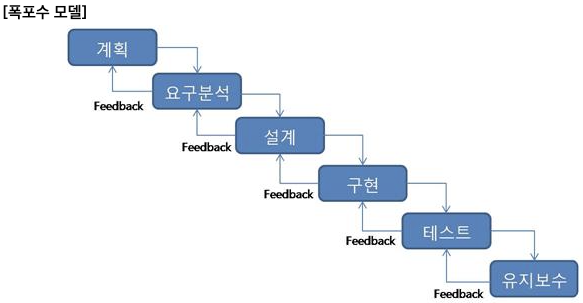
1. **프로젝트 주요 일정**

|  |  |  |
| --- | --- | --- |
| **주요 개발 활동** | **수행 일정** | **주요 산출물** |
| **프로젝트 제안** | 21.09.17 | 프로젝트 제안서 |
| **요구사항 분석** | 21.09.17 ~ 21.09.24 | 요구사항 명세서 |
| **프로젝트 계획** | 21.09.24 ~ 21.09.30 | 프로젝트 계획서 |
| **설계** | 21.10.01 ~ 21.10.15 | 설계 사양서 |
| **구현/테스트** | 21.10.15 ~ 21.11.31 | 소스/단위 테스트, 통합 테스트 결과 보고서, 소스코드 |
| **테스트** | 21.11.31 ~ 21.12.08 | 시스템 테스트 결과 보고서 |
| **프로젝트 종료** | 21.12.08 ~ 21.12.10 | 프로젝트 완료 보고서 |

1. **프로젝트 조직**
   1. 프로젝트 팀 조직

민주적 팀 구성

* + 1. 범위 및 역할  
       

1. **기술적 프로세스(프로젝트 개발 계획)**
   1. 개발 생명주기 모델
      1. 프로젝트 생명주기  
         피드백이 있는 폭포수 모델 (Waterfall Model)  
          
      2. 생명주기

|  |  |  |
| --- | --- | --- |
| **개발 단계** | **관련 요소** | **내용** |
| 분석 | 입력 산출물 | • 프로젝트 제안서 |
| 수행 태스크 | • 소프트웨어에 요구되는 기능, 성능, 인터페이스 등 사용자의  요구사항을 구체적으로 이해하고 문서화 한다.  • 프로젝트 제안서를 바탕으로 요구사항 명세서를 작성  - 요구사항 분석  - Use Case 작성  - UI프로토타입 작성  • 무엇을 어떻게 구현되어야 하는지 명확하게 작성한다.  • 생명주기 및 개발 방법론 결정  • 고객의 인터뷰 실시  • ERD작성 |
| 출력 산출물 | • 요구사항 명세서 |
| 시작 기준 | • 고객의 요구사항이 수집되고 기록이 되어 있어야 한다.  • 프로젝트 관계자는 고객의 요구사항을 파악하고 이해한다.  • 프로젝트 정의서가 승인되어야 하고 제안서가 문서화 되어야 한다. |
| 완료 기준 | • 유스케이스 별로 기능이 구체적으로 기술되어야 한다.  • 요구사항 명세서가 작성된다. |
| 계획 | 입력 산출물 | • 요구사항 명세서 |
| 수행 태스크 | • 고객의 요구사항 분석서 작성  • 추진 방안을 제시하고 시스템 개발 비용 및 소요기간, 인력등의 개발 계획을 설립한다.  • 데이터 및 컴퓨터 자원 관리 계획 수립  • 위험 관리 계획 수립  • 상세 수준의 WBS작성  • 프로젝트 계획서 작성 및 검토  • 테스트 계획서 작성 및 검토 |
| 출력 산출물 | • 프로젝트 계획서 |
| 시작 기준 | • 고객으로부터 요구사항이 전달된다.  • 요구사항 정의서가 문서화되고 승인되어야 한다.  • 요구사항 수집 및 정의서 작성 활동을 지원하기 위한 도구가 제공되고 이용 가능해야 한다.  • 요구사항 명세서를 기초로 무엇을 만들어야 하는지 안다. |
| 완료 기준 | • 고객의 요구사항이 변동되지 않고 수집 및 기록되어야 한다.  • 프로젝트 계획서가 작성되고 고객 및 이해 관계자에게 전달된다. |
| 설계 | 입력 산출물 | • 요구사항 명세서  • 프로젝트 계획서 |
| 수행 태스크 | • 아키텍처 정의서 작성 및 DB 설계  • 모든 시스템의 구조 결정(아래 내용을 포함)  - 프로그램의 데이터 구조  - 소프트웨어 구조  - 인터페이스 구조  - 알고리즘  • 클래스 다이어그램 작성  • 설계 명세서 작성 및 검토 |
| 출력 산출물 | • 설계 명세서 |
| 시작 기준 | • 요구사항 명세서가 구체적으로 기술되어 있고 프로젝트 관리자들이 요구사항 명세서를 이해하고 있다.  • 프로젝트 팀의 개발자 및 팀 구성원이 모두 결정되어야 한다.  • 프로젝트의 기간과 일정이 구체적으로 명확하게 결정되어야 한다.  • 프로젝트 개발 도구, 개발 언어, 관리 도구 등이 결정 되어야 한다. |
| 완료 기준 | • 설계 명세서가 작성된다.  • 프로젝트의 위험 요소가 정확하고 구체적으로 파악되고 이에 따른 계획이 수립되어야 한다. |
| 구현 | 입력 산출물 | • 설계 명세서 |
| 수행 태스크 | • 설계 명세서를 바탕으로 시스템의 프로그램 개발  • 개발 코딩 단위 테스트  • 설계 단계에서 논리적으로 결정한 문제 해결 방법을 특정  프로그래밍 언어를 사용하여 실제 프로그램을 작성한다.  • 프로그래밍 언어 선택, 기법, 스타일, 순서 등을 결정.  • 설계 단계에서 정해진 프로그램 인터페이스 및 알고리즘, 데이터 구조를 프로그램으로 실체화 |
| 출력 산출물 | • 소스코드 및 프로그램  • 테스트 보고서 |
| 시작 기준 | • 설계서가 작성된다.  • 프로젝트 팀 내 개발자는 설계서를 이해한다. |
| 완료 기준 | • 구현 후 단위 테스트를 마친다. |
| 테스트 | 입력 산출물 | • 소스코드 |
| 수행 태스크 | • 시스템 테스트 인수 테스트  • 프로그램이 입력에 따라 요구되는 결과대로 작동하는지, 내부적 이상 여부 및 오류 발견(테스트 시나리오 작성)  • 테스트 케이스 설계  • 테스트 결과 보고서 작성 및 검토  • 프로그램 오류 발견 |
| 출력 산출물 | • 테스트 결과 보고서 |
| 시작 기준 | • 통합 테스트를 마친다.  • 단위 테스트를 마친다.  • 테스트 계획을 세운다. |
| 완료 기준 | • 시스템/인수 테스트를 마친다.  • 테스트 결과를 문서화 한다. |
| 유지보수 | 입력 산출물 | • 테스트 보고서 |
| 수행 태스크 | • 프로젝트 변경 사항 파악  • 프로젝트 검토  • 개발된 소프트웨어의 변경 사항을 수정  • 수정 유지보수, 적응 유지보수, 기능 추가 유지보수  • 프로젝트 완료 보고서 작성 및 검토 |
| 출력 산출물 | • 프로젝트 완료 보고서 |
| 시작 기준 | • 모든 테스트를 실패 없이 마친다.  • 테스트 단계를 마치고 테스트 보고서가 문서화되어 있다. |
| 완료 기준 | • 완료 보고서가 작성된다.  • 소프트웨어의 변경 사항을 지속적으로 수정이 가능한 상태 |

* 1. 개발 방법론 및 도구 선정
     1. 개발 방법론 **객체지향 방법론**  
         객체지향 방법인 UML을 사용한다.  
         - 프로젝트 개발 언어가 객체지향 언어인 JAVA 이기에 메소드와 속성이 결합한 객체를 표현하기에 적합하다.

|  |  |
| --- | --- |
| **개발 단계** | **상세 내용** |
| 객체지향 분석 | 프로그램 사용자(User)와 프로그램의 기능을 UML 언어의 Use Case와 Sequence Diagram을 이용하여 시스템 요구사항을 분석하고, ER 다이어그램과 테이블 목록 및 기술을 하여 데이터 베이스 요구사항을 분석한다. |
| 객체지향 설계 | 시스템 설계 및 객체 설계를 분리하여 설계한다.  • 시스템 설계-DB 서버를 성능 및 최적 방안을 모색해 설계한다.  • 객체 설계-UML 언어의 Class Diagram을 이용 class를 정의하고 class들 간의 관계 및 알고리즘 그리고 자료구조를 설계한다. 또한 Sequence Diagram을 이용하여 메신저 프로그램의 흐름을 설계한다.  • DB 설계-ER 다이어그램을 이용하여 테이블을 상세 기술한다. |
| 객체지향 프로그래밍 | • 객체지향 설계를 바탕으로 Java 프로그램 언어를 이용하여 데이터 구조 부분과 관련 메소드를 포함하는 여러 객체들로 구현한다. |

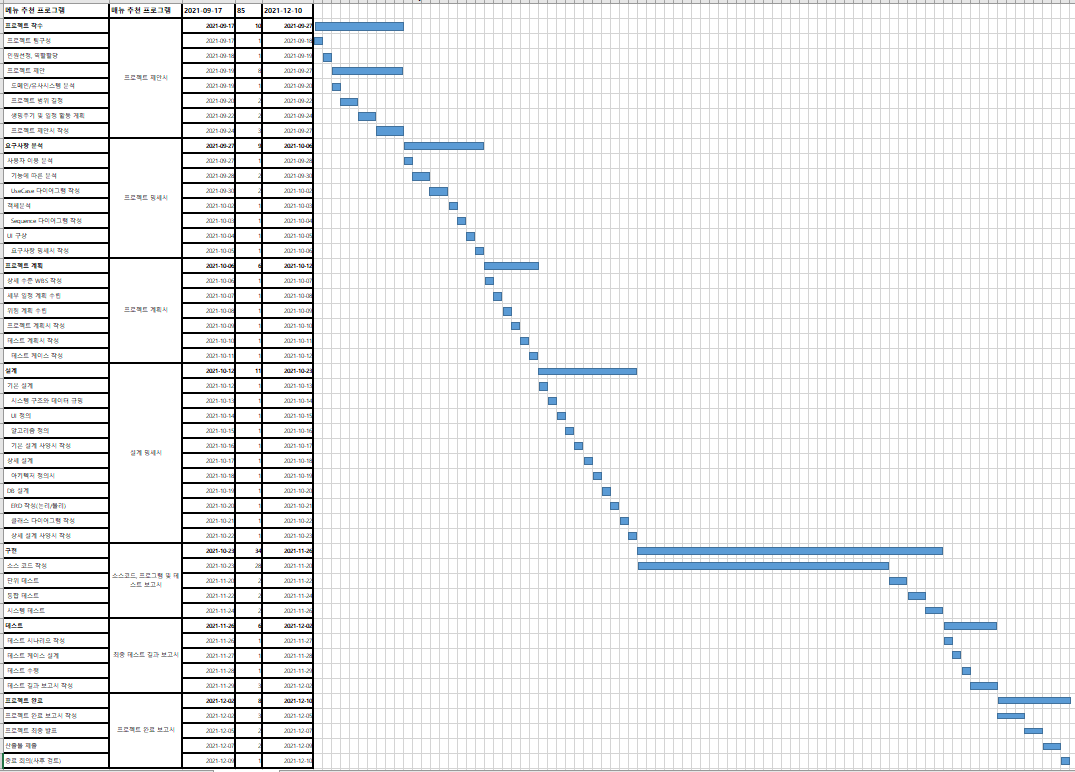
* + 1. 객체지향 방법론 절차  
        계획-설계-개발-테스트-검토(피드백) 순으로 반복적 진행

1. **개발 규모 산정**
   1. WBS



* 1. 상세일정

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **메뉴 추천 프로그램** | **매뉴 추천 프로그램** | **2021-09-17** | **85** | **2021-12-10** |
| **프로젝트 착수** | 프로젝트 제안서 | **2021-09-17** | **10** | **2021-09-27** |
| 프로젝트 팀구성 | 2021-09-17 | 1 | 2021-09-18 |
| 인원선정, 역할할당 | 2021-09-18 | 1 | 2021-09-19 |
| 프로젝트 제안 | 2021-09-19 | 8 | 2021-09-27 |
| 도메인/유사시스템 분석 | 2021-09-19 | 1 | 2021-09-20 |
| 프로젝트 범위 결정 | 2021-09-20 | 2 | 2021-09-22 |
| 생명주기 및 일정 활동 게획 | 2021-09-22 | 2 | 2021-09-24 |
| 프로젝트 제안서 작성 | 2021-09-24 | 3 | 2021-09-27 |
| **요구사항 분석** | 프로젝트 명세서 | **2021-09-27** | **9** | **2021-10-6** |
| 사용자 이용 분석 | 2021-09-27 | 1 | 2021-9-28 |
| 기능에 따른 분석 | 2021-09-28 | 2 | 2021-9-30 |
| UseCase 다이어그램 작성 | 2021-09-30 | 2 | 2021-10-2 |
| 객체분석 | 2021-10-2 | 1 | 2021-10-3 |
| Sequence 다이어그램 작성 | 2021-10-3 | 1 | 2021-10-4 |
| UI 구상 | 2021-10-4 | 1 | 2021-10-5 |
| 요구사항 명세서 작성 | 2021-10-5 | 1 | 2021-10-6 |
| **프로젝트 계획** | 프로젝트 계획서 | **2021-10-6** | **6** | **2021-10-12** |
| 상세 수준 WBS 작성 | 2021-10-6 | 1 | 2021-10-7 |
| 세부 일정 계획 수립 | 2021-10-7 | 1 | 2021-10-8 |
| 위험 계획 수립 | 2021-10-8 | 1 | 2021-10-9 |
| 프로젝트 계획서 작성 | 2021-10-9 | 1 | 2021-10-10 |
| 테스트 계획서 작성 | 2021-10-10 | 1 | 2021-10-11 |
| 테스트 케이스 작성 | 2021-10-11 | 1 | 2021-10-12 |
| **설계** | 설계 명세서 | **2021-10-12** | **11** | **2021-10-23** |
| 기본 설계 | 2021-10-12 | 1 | 2021-10-13 |
| 시스템 구조와 데이터 규명 | 2021-10-13 | 1 | 2021-10-14 |
| UI 정의 | 2021-10-14 | 1 | 2021-10-15 |
| 알고리즘 정의 | 2021-10-15 | 1 | 2021-10-16 |
| 기본 설계 사양서 작성 | 2021-10-16 | 1 | 2021-10-17 |
| 상세 설계 | 2021-10-17 | 1 | 2021-10-18 |
| 아키텍처 정의서 | 2021-10-18 | 1 | 2021-10-19 |
| DB 설계 | 2021-10-19 | 1 | 2021-10-20 |
| ERD 작성(논리/물리) | 2021-10-20 | 1 | 2021-10-21 |
| 클래스 다이어그램 작성 | 2021-10-21 | 1 | 2021-10-22 |
| 상세 설계 사양서 작성 | 2021-10-22 | 1 | 2021-10-23 |
| **구현** | 소스코드, 프로그램 및 테스트 보고서 | **2021-10-23** | **34** | **2021-11-26** |
| 소스 코드 작성 | 2021-10-23 | 28 | 2021-11-20 |
| 단위 데스트 | 2021-11-20 | 2 | 2021-11-22 |
| 통합 테스트 | 2021-11-22 | 2 | 2021-11-24 |
| 시스템 테스트 | 2021-11-24 | 2 | 2021-11-26 |
| **테스트** | 최종 테스트 결과 보고서 | **2021-11-26** | **6** | **2021-12-2** |
| 테스트 시나리오 작성 | 2021-11-26 | 1 | 2021-11-27 |
| 테스트 케이스 설계 | 2021-11-27 | 1 | 2021-11-28 |
| 테스트 수행 | 2021-11-28 | 1 | 2021-11-29 |
| 테스트 결과 보고서 작성 | 2021-11-29 | 3 | 2021-12-2 |
| **프로젝트 완료** | 프로젝트 완료 보고서 | **2021-12-2** | **8** | **2021-12-10** |
| 프로젝트 완료 보고서 작성 | 2021-12-2 | 3 | 2021-12-5 |
| 프로젝트 최종 발표 | 2021-12-5 | 2 | 2021-12-7 |
| 산출물 제출 | 2021-12-7 | 2 | 2021-12-9 |
| 종료 회의(사후 검토) | 2021-12-9 | 1 | 2021-12-10 |



* 1. 프로젝트 규모 산정 – 간이 기능 점수법

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 유형 | 내부 논리 파일 | 외부 연계 파일 | 외부 입력 | 외부 출력 | 외부 조회 |
| 가중치 | 7.5 | 5.4 | 4.0 | 5.2 | 3.9 |

|  |  |  |  |
| --- | --- | --- | --- |
| 기능 | 기능 유형 | 기능 명 | 기능 유형 |
| 사용자 관리 | 사용자 정보 관리 | 사용자 기본 정보 | 내부 논리 파일 |
| 추천 음식 관리 | 추천 음식 정보 관리 | 추천 음식 기본 정보 | 내부 논리 파일 |
| 게시판 관리 | 게시판 정보 관리 | 게시판 기본 정보 | 내부 논리 파일 |
| 음식점 관리 | 음식점 정보 관리 | 음식점 기본 정보 | 외부 연계 파일 |

**데이터 기능 점수**

데이터 기능 점수 = (ILF 개수 \* 7.5)+(EIF 개수 \* 5.4)

= (3\*7.5)+(1\*5.4)

= 22.5 + 5.4 = 27.9

**트랜잭션 기능 점수**

|  |  |  |  |
| --- | --- | --- | --- |
| 기능 명 | 기능 유형 | 기능 명 | 기능 유형 |
| 사용자 관리 | 사용자 정보 관리 | 사용자 등록 | EI |
| 사용자 정보 수정 | EI |
| 사용자 정보 조회 | EQ |
| 사용자 기본 정보 테이블 | 내부 논리 파일 |
| 비밀번호 관리 | 비밀번호 조회 | EQ |
| 비밀번호 변경 | EI |
| 추천 음식 관리 | 추천 음식 정보 관리 | 음식 추천 | EQ |
| 음식 등록 | EI |
| 음식 삭제 | EI |
| 추천 음식 정보 테이블 | 내부 논리 파일 |
| 게시판 관리 | 게시판 정보 관리 | 게시판 작성 | EI |
| 게시판 수정 | EI |
| 게시판 삭제 | EI |
| 게시판 조회 | EQ |
| 게시판 기본 정보 테이블 | 내부 논리 파일 |
| 음식점 관리 | 음식점 정보 관리 | 음식점 정보 조회 | EQ |
| 음식점 정보 테이블 | 외부 연계 파일 |

트랜잭션 기능 점수 = (EI개수 \* 4.0) + (EO개수 \* 5.2) + (EQ개수 \* 3.9)

= (8 \* 4.0) + (0 \* 5.2) + (5 \* 3.9)

= 32 + 0 + 19.5

= 51.5

**미조정 기능 점수 계산 = (데이터 기능 점수 + 트랜잭션 기능 점수)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 기능 | | 기능 수 | 평균 복잡도 | 기능 점수 | |
| 데이터 기능 점수 | ILF | 3 | 7.5 | 22.5 | 27.9 |
| EIF | 1 | 5.4 | 5.4 |
| 트랜잭션 기능 점수 | EI | 8 | 4.0 | 32 | 51.5 |
| EO | 0 | 5.2 | 0 |
| EQ | 5 | 3.9 | 19.5 |
| 계 | | | | | 79.4 |

**보정 전 개발 원가**

보정 전 개발 원가 = 미조정 기능 점수 \* 기능 점수당 단가

테이블이(가) 표시된 사진

자동 생성된 설명79.4 \* 519203 = 41,224,718.2

**보정 후 개발 원가**

보정 후 개발 원가 = 보정 전 개발 원가 \* (규모 보정 계수 \* 애플리케이션 보정 계수 \* 언어 보정 계수 \* 품질/특성 보정 계수)

= 41,224,718.2 \* (0.65 \* 1.0 \* 1.2 \* 1.0) = 32,155,280.196

1. **마일 스톤 및 일정 계획**

|  |  |  |
| --- | --- | --- |
| **마일 스톤** | | **일정 계획** |
| 착수 단계 | 프로젝트 제안서 | 21.09.17 ~ 21.09.20 |
| 분석 단계 | 요구사항 명세서 | 21.09.20 ~ 21.09.24 |
| 계획 단계 | 프로젝트 계획서 | 21.09.24 ~ 21.10.01 |
| 설계 단계 | 설계 사양서 | 21.10.01 ~ 21.10.15 |
| 구현 단계 | 소스 코드 | 21.10.15 ~ 21.11.31 |
| 테스트 단계 | 테스트 결과 보고서 | 21.11.31 ~ 21.12.08 |