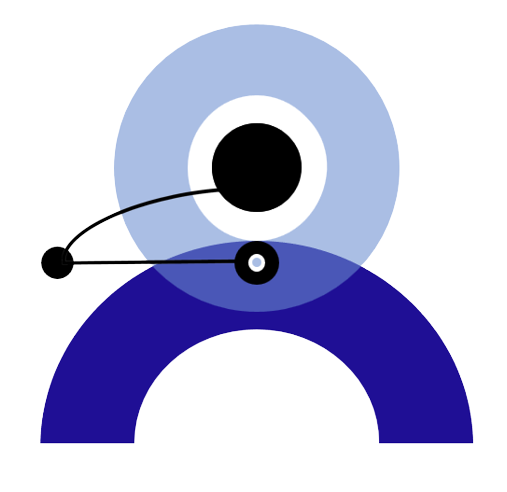
|  |
| --- |
|  |
| Self-Life 프로젝트 보고서 |
| 편의기능이 추가된 가계부 애플리케이션 |



**목 차**

**프로젝트 계획**

1. **프로젝트개요**
2. **개발계획**
3. **기술관리**
4. **검토회의**
5. 원, 그래픽, 스크린샷, 만화 영화이(가) 표시된 사진

   자동 생성된 설명**유지보수**
6. **성능시험**
7. **인수계획**

**분석 및 설계**

1. **요구분석정의(기능)**
2. **요구분석명세(기능)**
3. **요구분석명세(비기능)**
4. **기능흐름도**
5. **유스케이스**
6. **사용자 인터페이스 설계도**
7. **클래스 다이어그램**
8. **DB설계도**
9. **활동 다이어그램**

Self-Life

# 프로젝트 개요

|  |  |
| --- | --- |
|  | 차트이(가) 표시된 사진  자동 생성된 설명텍스트이(가) 표시된 사진  자동 생성된 설명  **1인가구 증가**(출처:KOSIS) **가계부브이로그**(유튜브채널’자취린이’) |

**사회적으로 1인가구가 큰 폭으로 증가하고 있으며, 잘못된 소비습관을 가지고 계획에 의한 올바른 소비형태가 아닌 충동에 의한 무분별한 소비형태가 다양해짐에 따라 자금이 부족한 상황에 마주치게 되는 사람들이 발생하고 있습니다. 이에 따라 가계부와 관련된 컨텐츠가 증가하고 새로운 자금관리 정보들이 전파되는 추세입니다.**

**따라서 저희는 자금관리 앱을 개발하여 여러가지 편의적의 서비스를 제공할 예정입니다.**

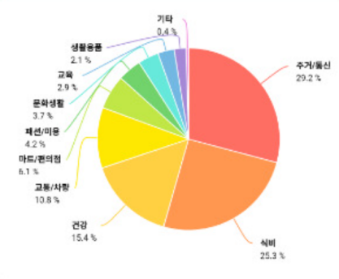
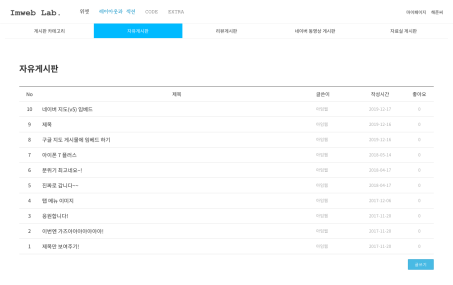
## 프로젝트 목표

* **기존의 자금관리 앱에서 편리하지만 구현되지 않았던 기능 제공**
* **자금 관련 데이터 값이 한 치의 오차 없이 연산 되도록 개발**
* **최적의 지출·소비 계획 현황 UI 제공**
* **배포까지 완료하여 실제 서비스 운용**

## 주요 고객층

|  |  |
| --- | --- |
|  | * **지출 소비량을 직접 관리하는 1인가구** * **생활비를 자신이 직접 관리하는 젋은 학생층** * **경제활동을 시작한 사회초년생** |

## 기능 구성 및 기대효과

달력이(가) 표시된 사진

자동 생성된 설명

**└** 자금 통계

**└**게시판

**└**캘린더

**└**생활용품 통계

1. **지출·수입 통계 그래프 : 사용자가 직접 입력하여 통계 그래프를 만들 수 있습니다. 주로 사용자의 수입과 소비내역이 작성되며, 이를 다양한 항목으로 나누어 표시하여 사용자가 소비의 방향성을 재설정하거나 소비계획을 수정하는 등의 도움을 얻을 수 있습니다.**
2. **게시판 : 게시판 개발을 통해 사용자 간의 소통과 정보 공유를 쉽게 할 수 있도록 합니다.**
3. **캘린더 : 캘린더는 일별로 사용자의 지출 내역을 보여주는 것이 주된 목적입니다. 통계그래프와는 달리 일별로 소비 내역을 확인할 수 있어 사용자는 자신의 소비습관을 다른 방향에서 파악할 수 있습니다. 또한, 캘린더를 통해 소비내역을 기록하면 사용자는 여러 지난 이벤트를 파악하는 데 도움을 받을 수 있으며, 본인의 예상 지출 금액을 입력하여 큰 지출이 예상되는 날을 대비할 수 있습니다.**
4. **계정관리/마이 페이지 : 사용자 회원가입, 로그인/로그아웃, 회원탈퇴를 통해 계정 관리를 구현하고, 앱 기본 서비스를 제공하는 페이지입니다. 사용자는 마이 페이지에서 자신의 계정 정보를 관리하고, 앱을 이용하는 데 필요한 기본적인 서비스를 이용할 수 있습니다.**

# 개발계획

## 개발방법론

**▶폭포수모델**

도표이(가) 표시된 사진

자동 생성된 설명

* **순차적으로 진행**
* **각 단계 종료 시 검증 후에 다음 단계 진행**
* **전 단계 결함 발견 시 피드백**
* **각 단계 별 산출물을 명확히 정의**

1. **한 단계를 종료할 때마다 검증회의를 수행하고, 부족한 부분이 있다면 검토 후 재 수행**
2. **검증회의에서 가결된 단계는 산출물을 최종적으로 완성 후, 다음 단계 실행**

## 산출물

|  |  |
| --- | --- |
| 결과물 | 설명 |
| **프로젝트 기획서** | 프로젝트의 개요, 범위, 일정, 예산, 위험 등을 정의합니다. |
| **요구사항 분석 정의서** | 요구사항을 수집하고 분석하여 문제를 파악하고, 이에 대한 개선 방안을 제시합니다. |
| **요구사항 분석 명세서** | 요구사항 분석 정의서에서 수집한 요구사항을 더 상세하게 기술합니다. |
| **설계서** | 시스템을 구현하기 위한 구체적인 설계 방안을 기술합니다. |
| **테스트 검증서** | 구현된 시스템에 대하여 테스트를 실시하고, 도출된 결과를 기술합니다. |
| **Self-Life 사용자 매뉴얼** | 시스템의 모든 기능 사용 방법들을 설명합니다. |
| **프로젝트 최종보고서** | 프로젝트의 결과를 종합하여 기록합니다. |

## 개발환경

|  |  |
| --- | --- |
|  | **운영체제 : Window11**  **개발언어 : Java (JDK 19.0.2)**  **개발 툴 : Android Studio (Android 13.0), eclipse**  **서버 및 데이터베이스 : Firebase**  **버전관리 : Git** |

## 조직 구성

|  |  |  |
| --- | --- | --- |
| 성 명 | 업 무 | 수 행 내 용 |
| **장기태(PM)** | **백엔드 개발**  **(프로젝트 총괄)** | **시스템 설계와 요구분석을 바탕으로 DB를 설계 및 구현하고 기능 구현 후 테스트, 디버깅 수행** |
| **김현동** | **백엔드 개발** |
| **최우혁** | **프론트엔드 개발** | **전체적인 시스템 설계와 요구분석을 수립하고 이를 바탕으로 UI를 설계 및 구현 작업을 수행** |
| **최현도** | **프론트엔드 개발** |
| **손동인** | **프론트엔드 개발** |
| **유환일** | **프론트엔드 개발** |

## 업무 구성(WBS)

도표이(가) 표시된 사진

자동 생성된 설명

## 개발일정

|  |  |
| --- | --- |
|  | 계획, 요구사항 분석, 설계 기간동안 매주 월/화요일마다 3시간씩 진행하고 구현 기간부터는 요일에 관계없이 진행되며, 총 일정은 업무 구성을 바탕으로 다음 표와 같이 진행됩니다. |

차트이(가) 표시된 사진

자동 생성된 설명

**주요 프로젝트 일정은 아래에 설명되어 있습니다.**

|  |  |  |  |
| --- | --- | --- | --- |
| 설명 | 시작 날짜 | 끝 날짜 | 기간 |
| **<프로젝트 계획>** | **3/6** | **3/14** | **2주** |
| **<요구 분석>** | **3/15** | **3/28** | **2주** |
| **<설 계>** | **4/3** | **6/9** | **10 주** |
| **<구 현>** | **6/12** | **9/15** | **14 주** |
| **<테스트>** | **9/18** | **10/6** | **3주** |
| **<운영/유지보수>** | **10/10** | **11/30** | **8주** |

**\* 3월부터 시작하여 9월 중순까지 개발을 마치고, 테스트 및 운영/유지보수 포함11월까지 진행하여 총 9개월 동안 진행될 예정입니다.**

# 기술 관리

## 변경 관리

* **분석 및 설계**

**요구사항 및 설계 내용이 변경될 때마다 회의를 거쳐 변경 심사 내용을 바탕으로 변경 수락 유무를 결정합니다. 변경이 수락됐을 시, 회의록에 해당 내용을 명세합니다.**

**변경 심사**

|  |  |
| --- | --- |
|  | **변경 이유의 타당성**  **변경 범위**  **변경이 미치는 영향(기간, 예산, 인력 등)** |

* **개발 단계**

**Git을 사용하여 개발물의 형상을 관리합니다. 각 목표에 따라 버전 번호를 부여하고, 목표를 이룰 때마다 다음 버전 번호를 명명합니다. 버전 번호 다음에 커밋 번호를 부여하고, n값이 증가할 때마다 개발자는 가장 최신 파일임을 인지합니다.**

**예시) Self\_Life ver1.9-2023.06.03**

|  |  |
| --- | --- |
|  | **Ver1. n : 애플리케이션 업무별 메인 페이지 인터페이스 구현, 계정업무 기능 구현**  **Ver2. n : 메인 페이지에서 파생되는 세부 페이지 인터페이스 구현, 업무별 우선순위에서 상위 기능 구현**  **Ver3. n : 반응형 인터페이스 구현, 업무별 우선순위에서 중위 기능 구현**  **Ver4. n : 사용자 인터페이스 최종 구현, 업무별 우선순위에서 하위 기능 구현(최종)** |

* **테스트 단계**

**요구분석서를 기반으로 테스트 계획서를 작성하며, 내용을 기반으로 테스트를 진행 후 문제점 및 이상 부분을 수정합니다. 수정을 할 때마다 알파벳 번호를 부여하며 커밋 메시지에 수정 내용 및 개선부분을 명세합니다.**

**예시)**

|  |  |
| --- | --- |
|  | **Self\_Life ver.A – 자금 통계 기능 DB연동 오류 해결**  **Self\_Life ver.B – 새로 고침 시, 페이지 재 요청 오류 해결** |

## 위험 관리

* **프로젝트 수행 중 개발 인원의 조기 취업**
* **요구사항 확정 이후에 계속되는 변경 요구**
* **계획한 개발 기간 초과**
* **소스코드 파일의 예상치 못한 삭제**

## 문제점 해결방안

* **조기 취업을 한 팀원은 퇴근 후 개인시간, 주말을 활용하여 주어진 역할을 수행하여 공백이 생기지 않게 진행**
* **요구사항 분석 단계에서 매시간마다 정의서, 명세서를 검토하여 최종적으로 논의하고 확정한 후 설계하여 예방**
* **각 단계별 일정을 최대한 투자할 수 있는 기간으로 선정하여 기간 초과를 사전에 예방**
* **백업을 주기적으로 실행하며, 팀장 통제 하에 버전 관리 일상화**

## 비용 관리 ( \15,427,520 )

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **단계** | **기능** | | | **개수** | **평균**  **가중치** | **기능점수** |
| **1** | **데이터**  **기능** | **ILF** | **로그인/로그아웃 기능,아이디, 비밀번호 찾기 기능,지출/수입 내역 수정 및 삭제 기능,생활용품 사용기간 예측 기능,쓰기/댓글 수정, 삭제 기능,관리자 권한 기능, 최적의 지출·수입 계획 모델 제공 ,**  **앱 서비스 이용 안내 기능** | **15** | **7.5** | **112.5** |
| **EIF** | **회원가입 기능,닉네임, 비밀번호 변경 기능,고정지출/수입계획 입력 기능,현재 지출/소득 입력 기능,생활용품 입력, 수정, 삭제 기능,게시판 글쓰기/댓글 기능,사진 업로드 기능 ,회원 탈퇴 기능** | **14** | **5.4** | **75.6** |
| **트랜잭션**  **기능** | **EI** | **지출/수입현황 게시판 기능,전년 대비 지출/수입 통계,예산 한도 비율 가이드 기능,지출·수입 위젯 기능,글/댓글 신고 기능,고객센터 기능** | **9** | **4.0** | **36** |
| **EO** | **알림 기능,캘린더 기능,게시물 목록 검색 기능,글 제목 검색 기능,지출·수입 내역 열람 기능,생활용품 재구매 안내 기능,** | **7** | **5.2** | **36.4** |
| **EQ** | **지출/수입 내역 조회,쓰기/댓글 조회,사진 조회** | **5** | **3.9** | **19.5** |
| **기능 점수 합** | **280** | | | | |
| **2** | **보정 전 개발 원가** | **280x6,0000원=16,800,000원** | | | | |
| **3** | **보정 계수** | **연계 복잡성 수준(0.88)x성능요구 수준(1.00)x운영 환경 호환성(1.00)x보안성 수준(1.03)** | | | | |
| **보정 후 개발 원가** | **16,800,000x0.88x1.00x1.00x1.03=15,227,520원** | | | | |

**※추가비용 – 로고 이미지 외주(200,000원)**

# 검토회의

|  |  |
| --- | --- |
|  | **주최자 : 장기태(PM) 참석인원 : 최우혁, 김현동 , 최현도, 손동인, 유환일** |

* **주간회의 : 한 주 동안 진행한 업무를 보고하고 인수인계 내용을 전달, 다음 주간 계획 수립**
* **월간회의 : 한 달 동안 진행한 모든 업무를 검토하여 피드백하고, 다음 월간 계획 수립**
* **검증회의 : 한 업무를 마치고 다음 업무를 수행 하기 전, 수행했던 업무를 최종적으로 검증**

# 유지보수

**유지보수 방법**

**예방적 유지보수: 프로젝트의 안정적인 운영을 위해 주기적인 예방점검과 정기적인 업데이트를 실시합니다. 이를 통해 잠재적인 문제를 조기에 발견하고 수정할 수 있습니다.**

**수정적 유지보수: 사용자의 피드백이나 발생한 오류에 대한 수정 작업을 수행합니다. 이를 통해 소프트웨어의 기능을 개선하고 사용자 요구에 더욱 부합하도록 유연하게 대응할 수 있습니다.**

**확장적 유지보수: 프로젝트에 새로운 기능이나 모듈을 추가할 경우, 관련 기능을 유지하고 기존 기능과의 호환성을 유지하면서 확장 작업을 수행합니다. 이를 통해 프로젝트의 기능적인 발전과 함께 변화하는 요구사항에 대응할 수 있습니다.**

**기술적 유지보수: 기술적인 트렌드나 플랫폼의 변화에 따라 시스템을 최신 기술에 맞추기 위한 유지보수 작업을 수행합니다. 이를 통해 보안 강화, 성능 최적화, 호환성 유지 등의 목표를 달성할 수 있습니다.**

**유지보수 절차**

**문제 식별: 사용자 피드백, 오류 보고 또는 모니터링 시스템 등을 통해 발생한 문제를 식별합니다. 문제의 성격과 심각도를 파악하여 우선순위를 결정합니다.**

**문제 분석: 문제의 원인을 분석하고 해당 문제를 해결하기 위한 방안을 도출합니다. 문제의 근본 원인을 파악하여 유사한 문제의 발생을 예방할 수 있는 방법을 모색합니다.**

**문제 해결: 도출된 방안에 따라 문제를 해결합니다. 코드 수정, 데이터베이스 업데이트, 구성 변경 등의 작업을 수행하여 문제를 해결하고 시스템을 정상적으로 복구합니다.**

**테스트 및 검증: 문제 해결 후 시스템을 테스트하여 문제가 완전히 해결되었는지 검증합니다. 기능 및 성능 테스트, 회귀 테스트 등을 수행하여 문제의 재발을 방지합니다.**

**변경 사항 관리: 유지보수 작업을 통해 수정된 코드, 업데이트된 데이터베이스 등 변경된 내용을 관리합니다. 변경 이력을 추적하고 문제 발생 시 변경 내역을 파악하여 원인 분석에 활용합니다.**

**문서화: 유지보수 작업의 결과물을 문서화하여 관리합니다. 수정된 코드, 데이터베이스 스키마, 테스트 결과 등을 문서로 정리하여 추후 유지보수 작업에 활용하고 지식 공유를 위해 활용합니다.**

# 성능 시험 방법

**소프트웨어 결함의 문제를 식별하고 품질 평가 및 개선을 통해 품질과 성능을 향상하고 소프트웨어의 보안성과 안정성 확보를 목표로 합니다. 테스트는 기능 단위 테스트, 통합 서비스 테스트로 구분하여 실행합니다.**

**기능 단위 테스트**

* **확인 검사**
* **검증 검사**

**통합 서비스 테스트**

* **애플리케이션 통합 검사**
* **비 기능 명세 내용과 동일하게 실행하는지 검사**
* **서버 및 DB 연동 검사**
* **보안 검사**

# 인수계획

**인수 기준 및 조건: 시스템의 모든 기능이 정상적으로 동작해야 하며, 사용자 매뉴얼에 명시된 요구사항을 충족해야 합니다.**

**인수 검증을 위한 테스트 방법과 절차: 사용자 시나리오를 작성하여 시스템의 전반적인 기능을 테스트하며, 이후 버그 및 오류 사항에 대한 보고서를 작성합니다.**

**인수 이전에 수행되어야 할 작업과 준비 사항: 인수 전 사용자 매뉴얼과 트레이닝 자료 제작, 시스템 백업 및 복원 테스트 등을 수행합니다.**