**프로젝트 계획서**

<체어맨>

**- 변 경 이 력 -**

|  |  |  |  |
| --- | --- | --- | --- |
| **일자** | **버전** | **변경 내역** | **작 성 자** |
| 2021.04.12 | ver 1.0.0 | 회의를 토대로 중요 내용 작성 | 한정민 |
| 2021.04.13 | ver 1.0.1 | 1.3 <조직> 작성, 1.5 <도구> 작성 | 이기탁 |
| 2021.04.14 | ver 1.0.2 | 1.1 <목적> 1.2 <주요일정> 작성 | 김수환 |
| 2021.04.14 | ver 1.0.3 | 1.4< 생명주기 모델> 작성 | 이치구, 최윤제 |
| 2021.04.14 | ver 1.1.0 | 이전 버전을 통합하여 수정 및 작성  WBS 토의 및 작성  WBS 기반 일정 작성 | 팀원 전원 |
| 2021.04.15 | ver 1.1.1 | Gantt 차트 작성 | 김수환 |
| 2021.04.15 | ver 1.1.2 | PERT 차트 작성 | 이치구, 한정민 |
| 2021.04.15 | ver 1.1.3 | 일정 수정 | 최윤제 |
| 2021.04.16 | ver 1.2.0 | 산출물 관리 | 팀원 전원 |
|  |  |  |  |
|  |  |  |  |

- 목 차 -

[**1.**](#_2et92p0) **프로젝트 개요 - 3 -**

[1.1](#_2et92p0) 목적 - 3 -

[1.2](#_tyjcwt) 주요 일정 - 3 -

[1.3](#_3dy6vkm) 조직 - 4 -

[1.3.1](#_4d34og8) 조직도 - 4 -

[1.3.2](#_2s8eyo1) 역할 및 책임 - 4 -

[1.4](#_17dp8vu) 생명주기 모델 - 5 -

[1.5](#_3rdcrjn) 도구 - 5 -

[**2.**](#_26in1rg) **규모 산정 - 6 -**

[2.1](#_lnxbz9) WBS(Work Breakdown Structure) - 6 -

[**3.**](#_35nkun2) **일정 - 7 -**

[**4.**](#_1ksv4uv) **산출물 관리 - 8 -**

[**5.**](#_44sinio) **위험 관리 계획 - 9 -**

1. 프로젝트 개요
   1. 목적

|  |
| --- |
| 프로젝트에 대한 간단한 소개 및 수행 목적을 기술한다. |

|  |  |
| --- | --- |
| **제목** | **상세내용** |
| 기간 | 2021.04.05~2021.06.07 |
|
|
|
| 주제 | 캠핑장 예약 서비스를 제공하는 플랫폼 설계 |
| 목적 | 캠핑객들에게 캠핑 예약 및 캠핑 정보 종합서비스 제공  캠핑 사업자에게 캠핑객 중개 인터페이스 제공 |
| 팀 구성 | 한정민, 이기탁, 이치구, 최윤제, 김수환 총 5명 |
| 프로젝트 명 | 여기놀자 |
| 프로젝트 명 의미 | 주 의미 - 여기 어때와 야놀자를 합쳐 캠핑장 예약 어플로써의 대표 주자  보조 의미 - 캠핑이 여가 생활에 있어서 친근한 이미지를 가지게 되어 캠핑에 대한 접근성을 높인다. |
| 개발 플랫폼 | 앱을 기반으로 구동되는 모바일 어플리케이션  대상 운영체제 : 안드로이드 |

* 1. 주요 일정

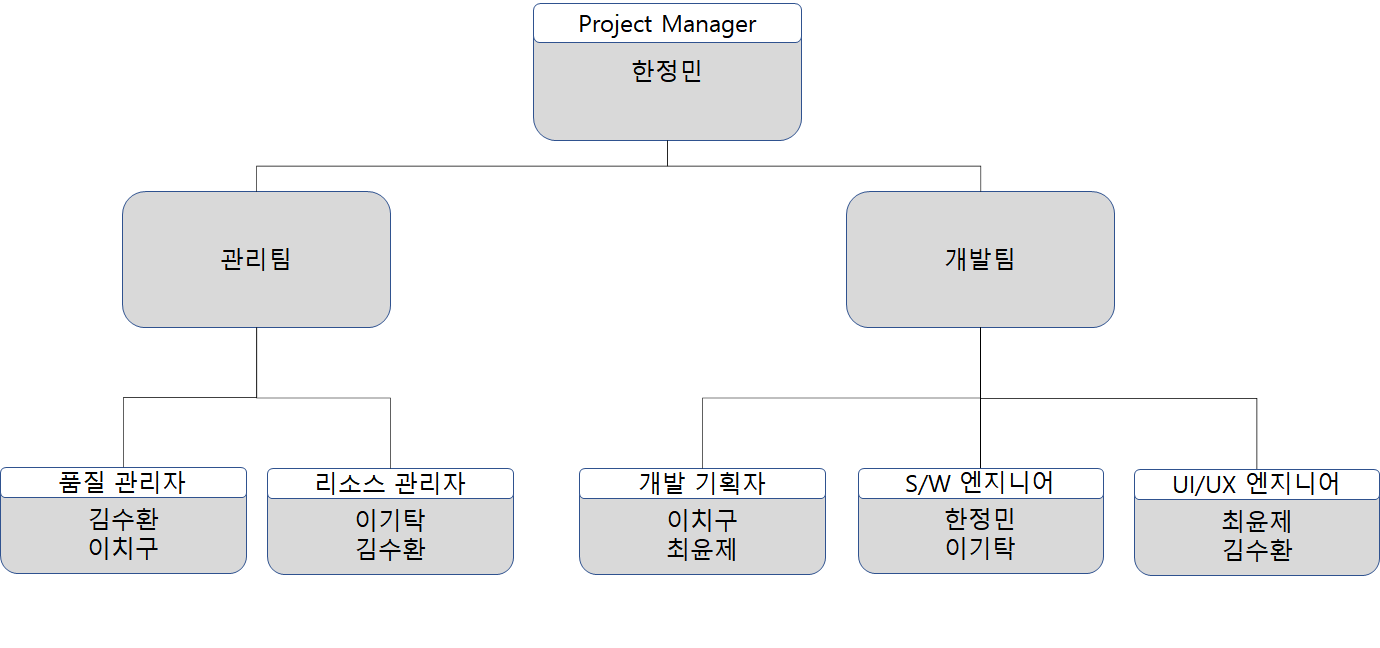
|  |
| --- |
| 프로젝트에서 주어진 주요 단계, 일정 별 산출물을 기술한다. |

|  |  |  |
| --- | --- | --- |
| **단계** | **일정** | **산출물** |
| 요구사항 분석 | 2021.04.05 ~ 2021.04.09 | 요구사항 명세서 |
| 프로젝트 계획 | 2021.04.12 ~ 2021.04.19 | 프로젝트 계획서 |
| 설계 및 디자인 | 2021.04.27 ~ 2021.05.17 | UI 디자인  Class 디자인 |
| 구현/QA | 2021.05.18 ~ 2021.05.31 | 테스트 보고서 |
| 프로젝트 중간발표 | 2021.04.20 ~ 2021.04.26 | 발표자료 |
| 프로젝트 종료 | 2021.06.01 ~ 2021.06.07 | 발표자료 |
| 전체 단계 | 2021.04.05 ~ 2021.06.07 | 회의록 |

* 1. 조직

|  |
| --- |
| 프로젝트 내 구성원들의 조직도 및 역할을 기술한다. |

* + 1. 조직도



* + 1. 프로젝트 구성원 정보

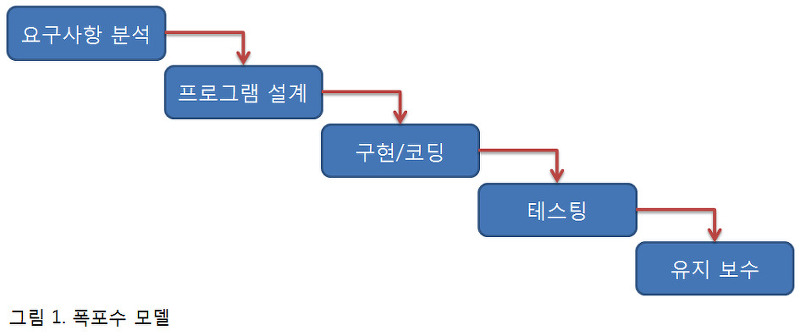
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **이름** | **학과** | **학번** | **메일** | **연락처** |
| 김수환 | 컴퓨터과학과 | 201710904 | tnghks5335@gmail.com | 010-2932-4314 |
| 이기탁 | 컴퓨터과학과 | 201710949 | gitak0127@gmail.com | 010-2992-5240 |
| 이치구 | 컴퓨터과학과 | 201710965 | ttee2347@nate.com | 010-5124-2230 |
| 최윤제 | 컴퓨터과학과 | 201911017 | dbswp703@gmail.com | 010-2169-3069 |
| 한정민 | 컴퓨터과학과 | 201710998 | han51361@gmail.com | 010-5279-3774 |

* + 1. 역할 및 책임

|  |  |  |
| --- | --- | --- |
| **역할** | **팀원** | **책임** |
| Project Manager | 한정민 | 프로젝트 계획과 실행의 종합 책임  프로젝트 이슈 및 리스크 관리 책임  프로젝트 일정 조율 책임  구성원의 역할 분담 책임  비상 상황 대비 매뉴얼 확립 책임 |
| 품질관리자 | 김수환, 이치구 | 품질 개선 및 결함 재발 방지 책임  테스트 계획 수립 책임  테스트 케이스 작성 책임  모든 기능의 테스트 실시 책임  소비자 피드백 반영 책임 |
| 리소스 관리자 | 이기탁, 김수환 | 산출물 저장의 책임  산출물 호환성 무결점에 대한 책임  기술문서 및 메뉴얼 작성 책임  번역 프로세스 정립 책임  모든 리소스의 보안 책임 |
| 개발 기획자 | 이치구, 최윤제 | 요구사항 수집 및 기획 책임  개발 시나리오 설정 책임  기본 설계의 무결성에 대한 책임  Class 설계 책임  소프트웨어 보안에 대한 책임 |
| SW 엔지니어 | 한정민, 이기탁 | 상세 설계 책임  구현 일정 설계 책임  class 구체화 책임  소프트웨어 무결성에 대한 책임  코드 표준화에 대한 책임 |
| UI/UX 엔지니어 | 최윤제, 김수환 | 구조 및 화면 설계 책임  스토리보드 설계 책임  앱 디자인 및 개발 책임  디자인 파일 보관의 책임 |

* 1. 생명주기 모델 : Water fall

폭포수 모델은 순차적으로 소프트웨어를 개발하기 위한 과정으로 각 단계별로 체계적인 문서화가 가능해 요구사항의 변화가 적은 프로젝트에 적합하다. 단계별 상세 내용은 아래 그림과 표와 같다.



|  |  |
| --- | --- |
| **단계** | **조직상세 내용** |
| 요구사항 분석 | 새로운 제품을 개발하는 과정에서 고객의 요구사항을 충족하는 기능, 성능, 인터페이스들을 파악하고 문서화하여 요구사항 명세서를 제출한다. |
| 프로그램 설계 | 프로그램을 구현하기 전 전체적인 시스템의 구조를 설계하여 프로그램의 데이터 구조, 소프트웨어 구조, 인터페이스 구조, 알고리즘을 포함해 설계서를 제출한다. |
| 구현/코딩 | 설계와 디자인 단계에서 제출한 보고서를 프로그래밍 언어를 이용해 컴퓨터 프로그램으로 변환 시킨다. |
| 테스팅 | 프로그램이 일정 수준의 품질을 가질 수 있도록 성능 테스트를 체계적으로 수행하여 테스트 보고서를 제출한다. |
| 유지보수 | 유지보수는 이미 개발된 소프트웨어의 결함을 수정하고 성능이나 기타 특성을 개선하기 위해 수정한다.. |

* 1. 도구

<표 1 - 연락 도구>

|  |  |  |
| --- | --- | --- |
| **도구 명** | **내용 및 규칙** | |
| Slack | 기본적으로 사용하는 개발 도구  목적에 맞게 채널을 여러 개 생성하여 활용함.  채널의 이름과 목적은 아래와 같음. | |
|
|
|
| 채널 이름 | 채널 목적 |
| # -5 | 처음 개발을 시작하면서 팀원들과 친목 도모를 위한 채널. |
| # 공지사항 | 공동 개발 할 날짜와 그 날짜에 할 일을 공지하기 위한 채널 |
| # 과제 | github 나 e-캠퍼스에 올라온 과제 내용을 올리는 채널 |
| # 액터-유스케이스-다이어그램 | 프로젝트 관련한 최신화된 내역들을 바로 확인해가며 자신의 작업에 적용 해야 된다고 생각하여 각 파트별로 생성된 작업물을 업로드 할 채널을 만듦. |
| # 요구사항명세서 |
| # 프로젝트계획서 |
| # 회의록 |
| 한정민, 이기탁, 이치구, 최윤제, 김수환 총 5명 | |
| 구글 docs | 실시간으로 최신화된 문서 내용에 대하여 댓글과 채팅을 통한 피드백 제공 수단 | |
| 카카오톡 | 회의 날짜를 정하거나, slack을 이용하지 못하는 경우(ex.서버다운) 긴급 상황이 발생하였을 때를 대비하여 카카오톡 단체방을 이용 | |

<표 2 - 디자인 및 개발 도구>

|  |  |
| --- | --- |
| **도구 명** | **내용 및 규칙** |
| Oven | 유스케이스 화면 기술과 앱 스크린 디자인을 위한 도구로 사용. |
|
|
|
| Git | 상호간 개발 자료를 공유하기 위한 도구로 사용. |
| Putty | 서버에 접속하여 파일 읽기/쓰기를 하기 위한 도구로 사용. |
| Visual studio code | 안드로이드 앱 개발(코틀린)을 위한 코딩 도구로 사용. |
| Adobe illustrator | 앱의 디자인 요소를 제작하는 도구로 사용. |
| Adobe xd | 디자인 프로토타입을 만들기 위한 도구로 사용. |
| 먼데이 닷컴 | 간트 차트를 만들기 위한 도구로 사용. |

< 표 3 - 자료 저장 도구>

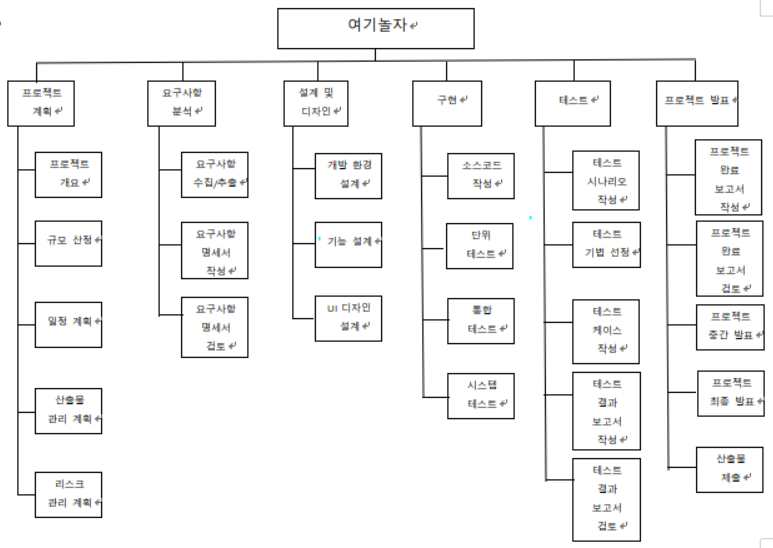
|  |  |
| --- | --- |
| **도구 명** | **내용 및 규칙** |
| Slack Cloud | 개발에 필요한 자료들을 공유할 때 사용. |
|
|
|
| AWS | 앱 서비스를 구현할 때 사용. |
| 드롭 박스 | 중요 자료를 백업해 둘 때 사용. |

< 표 4 - 문서 작성 도구>

|  |  |
| --- | --- |
| **도구 명** | **내용 및 규칙** |
| 워드 | 회의록 및 타 문서 작성을 하기 위해서 사용. |
|
|
|
| 엑셀 | 작성한 차트를 문서화 시키기 위해서 사용. |
| 파워포인트 | 기초적인 표 이미지 작업과 발표 자료를 만들기 위해 사용. |
| 구글 문서 | 실시간으로 프로젝트 계획서, 요구사항 명세서 작성을 하기 위해서 사용. |

1. 규모 산정

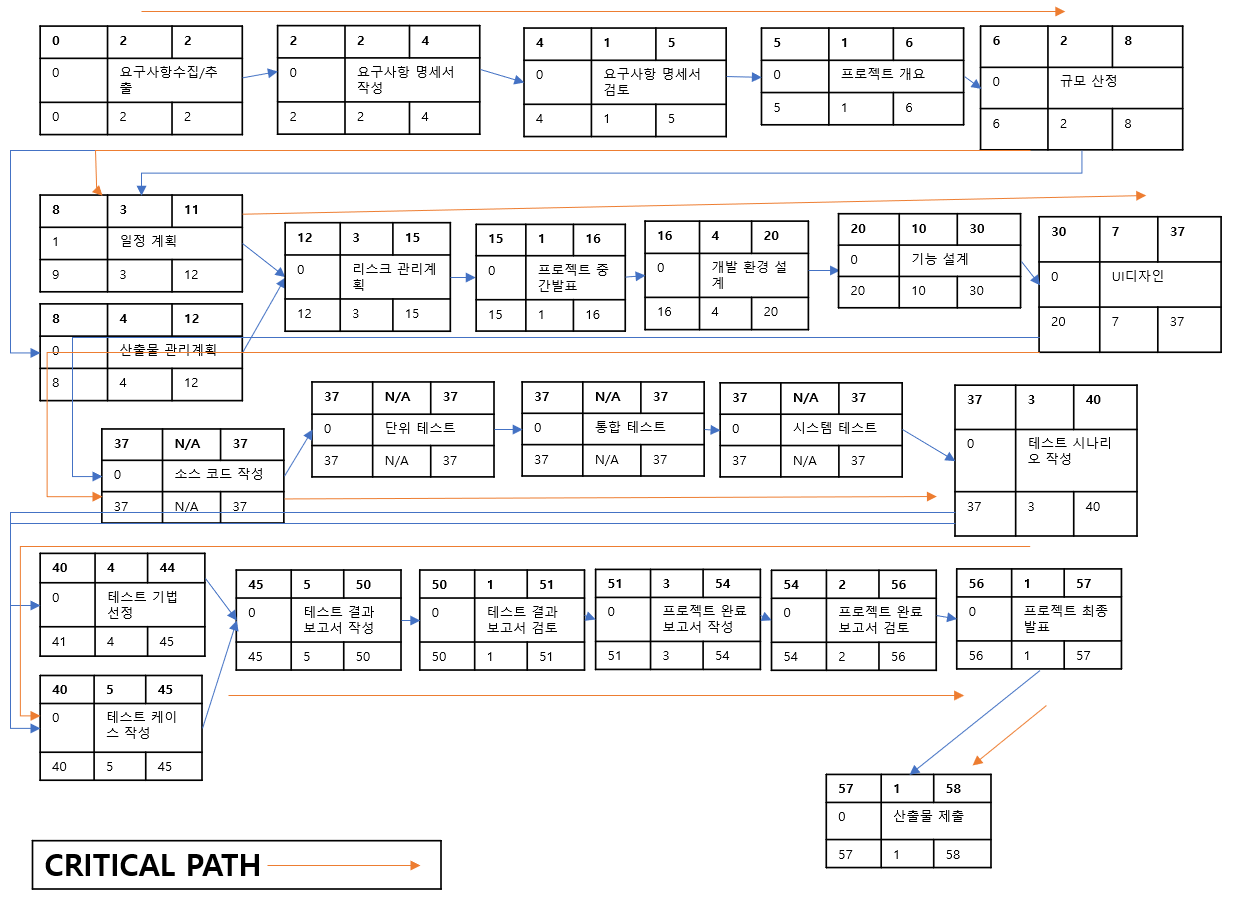
WBS(Work Breakdown Structure)



1. 일정

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **구 분** | **작 업** | **시작일** | **기 간(일)** | **종료일** |
| 요구사항 분석 | 요구사항 수집/추출 | 2021.04.05 | 2 | 2021.04.06 |
| 요구사항 명세서 작성 | 2021.04.07 | 2 | 2021.04.08 |
| 요구사항 명세서 검토 | 2021.04.09 | 1 | 2021.04.09 |
| 프로젝트 계획 | 프로젝트 개요 | 2021.04.11 | 2 | 2021.04.12 |
|
|
|
| 규모 산정 | 2021.04.13 | 2 | 2021.04.14 |
| 일정 계획 | 2021.04.15 | 3 | 2021.04.17 |
| 산출물 관리 계획 | 2021.04.15 | 4 | 2021.04.18 |
| 리스크 관리 계획 | 2021.04.19 | 3 | 미정(2021.04.21) |
| 시스템 디자인  및 설계 | 개발 환경 설계 | 2021.04.27 | 4 | 2021.04.30 |
| 기능 설계 | 2021.05.01 | 10 | 2021.05.10 |
| UI 디자인 | 2021.05.11 | 7 | 2021.05.17 |
| 시스템 구현 | 소스 코드 작성 | N/A | | |
| 단위 테스트 |
| 통합 테스트 |
| 시스템 테스트 |
| 테스트 | 테스트 시나리오 작성 | 2021.05.18 | 3 | 2021.05.20 |
| 테스트 기법 선정 | 2021.05.21 | 4 | 2021.05.24 |
| 테스트 케이스 작성 | 2021.05.21 | 6 | 2021.05.26 |
| 테스트 결과 보고서 작성 | 2021.05.26 | 5 | 2021.05.30 |
| 테스트 결과 보고서 검토 | 2021.05.31 | 1 | 2021.05.31 |
| 프로젝트 발표 | 프로젝트 중간 발표 | 2021.04.26 | 1 | 2021.04.26 |
| 프로젝트 완료 보고서 작성 | 2021.06.01 | 3 | 2021.06.03 |
| 프로젝트 완료 보고서 검토 | 2021.06.03 | 3 | 2021.06.05 |
| 프로젝트 최종 발표 | 2021.06.06 | 1 | 2021.06.06 |
| 산출물 제출 | 2021.06.07 | 1 | 2021.06.07 |

3.1 PERT 차트



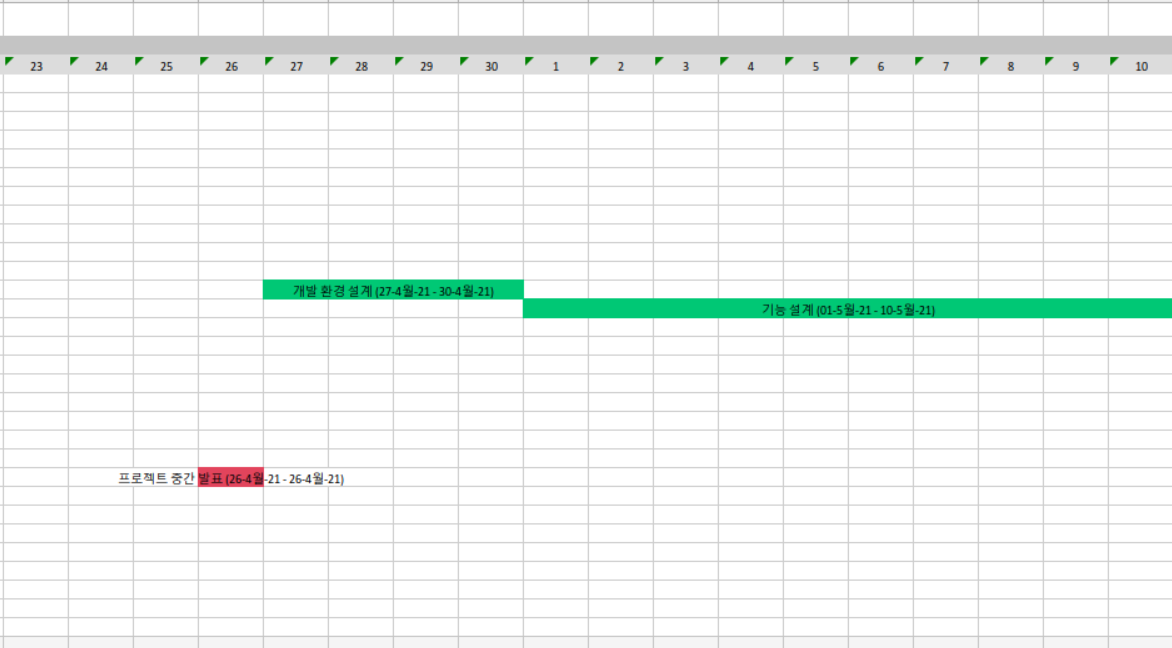
3.2 Gantt 차트

(1)

테이블이(가) 표시된 사진

자동 생성된 설명

(2)



(3)

테이블이(가) 표시된 사진

자동 생성된 설명

1. 산출물 관리

|  |
| --- |
| 산출물의 관리 방안을 기술한다.  산출물을 어디에 저장하고, 산출물 명을 어떤 규칙으로 할지 등을 기술한다. |

|  |  |  |
| --- | --- | --- |
| **계획** | **내용 및 규칙** | |
| 산출물 저장 | 작업물 파일은 협업 도구인 Slack을 이용하여 채널별 목적에 맞게 업로드 한다. | |
|
|
|
| 중요 파일의 백업은 서버나 드롭박스를 이용하여 모든 인원이 접속 가능 한 환경에 보관한다. 백업 과정에서 각 파일을 분류해서 작업별로 나눠둔다. | |
| 모든 산출물은 리소스 관리자가 다운받아서 추가적으로 보관하고 있도록 한다. | |
| 산출물 명 설정 | 작업 산출물 명은 <프로젝트 버전\_날짜시간-이름>로 정한다. | |
| 버전은 Waterfall 모델을 기준으로 버전을 1씩 증가시키며 수정 될 시 에는 0.01을 더한다.  개인 수정이 아닌, 회의를 통하여 팀원모두가 함께 수정 또는 기준이 변경 되었을 때 소수점 2번째 자리를 올림 한다. | |
| 최종 완성본은 지정된 제출명에 따라 저장 후 제출. | |
| 산출물은 google 문서에서 제공하는 바탕 폰트를 사용한다. | |
| 버전 별 수정 사항을 별도의 문서 혹은 작업물 문서에 기입하여 작업시에 차질이 없도록 한다. | |
|
|
| 단계별 산출물  관리 세부사항 | 요구사항 명세서 | 요구사항 명세서는 요구사항을 기능에 따라 구분을 지어 작성한다.. |
| 프로젝트 계획서 | 프로젝트 계획서는 위의 방법들을 토대로 작성한다. |
|
| 소프트웨어 설계문서 | 설계문서는 기능을 수행하는 모듈을 단위로 구분하여 작성한다. 설계 단계에서 알고리즘을 같이 작성해서 각 기능들이 연동되고 작동되는 것을 명시해 구현을 할 떄 이해하기 편하도록 한다. |
|
| 테스트 보고서 | 테스트 보고서는 각 기능들이 잘 작동하는지 확인하고, 모듈별로 구분하여 테스트 결과를 작성한다. 또한, 구분한 모듈마다 테스트 성공과 실패의 경우를 각각 구분하여 작성한다. |
| 회의록 | 회의록은 팀 회의에 대하여 당일 회의 했던 안건을 정리하여 작성 후 slack에 올린다. |

1. 위험 관리 계획

|  |
| --- |
| 프로젝트 수행 중에 발생할 위험을 식별하고, 이를 대처하기 위한 방안을 기술한다.  아직 위험 관리를 배우지 않은 상태에서는 작성하지 않는다. |

|  |  |  |  |
| --- | --- | --- | --- |
| 위험 요소 | 가능성 | 영향도 | 대처 방안 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |