프로젝트 계획서

Campers'

- 변 경 이 력 -

|  |  |  |  |
| --- | --- | --- | --- |
| 일자 | 버전 | 변경 내역 | 작 성 자 |
| 2021.04.12 | ver 1.0 | 프로젝트 개요 및 산출물 관리 작성 | 이아현 |
| 2021.04.16 | ver 1.1 | 프로젝트 규모 산정 및 일정 작성 | 모두 |
| 2021.04.18 | ver 1.2 | 규모 산정 및 일정 변경 | 임지민 |
| 2021.04.27 | ver 1.3 | 위험 관리 계획 작성 | 이아현, 최아름 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

- 목 차 -

[1. 프로젝트 개요 - 3 -](#_Toc447537505)

[1.1 목적 - 3 -](#_Toc447537506)

[1.2 주요 일정 - 3 -](#_Toc447537507)

[1.3 조직 - 4 -](#_Toc447537508)

[1.3.1 조직도 - 4 -](#_Toc447537509)

[1.3.2 역할 및 책임 - 4 -](#_Toc447537510)

[1.4 생명주기 모델 - 5 -](#_Toc447537511)

[1.5 도구 - 5 -](#_Toc447537512)

[2. 규모 산정 - 6 -](#_Toc447537513)

[2.1 WBS(Work Break](#_Toc447537514)[down Structure) - 6 -](#_Toc447537514)

[3. 일정 - 7 -](#_Toc447537515)

[4. 산출물 관리 - 8 -](#_Toc447537516)

[5. 위험 관리 계획 - 9 -](#_Toc447537517)

# 프로젝트 개요

## 목적

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

고객은 편하게 캠핑장 예약이 가능하고, 반대로 캠핑장 주인은 쉽게 본인의 캠핑장을 등록할 수 있는 캠핑장 예약 웹 사이트이다. 사용 시 인터넷 연결은 필수이다. 또한 사용자나 캠핑장 정보 같은 데이터는 서버 내에 데이터베이스에서 관리한다.

개발 프레임워크 및 언어 : React, Node.js -> Javascript

팀 구성원 : 이아현(Full), 유서정(Front), 최아름(Front), 강세미(Back), 임지민(Back)

## 주요 일정

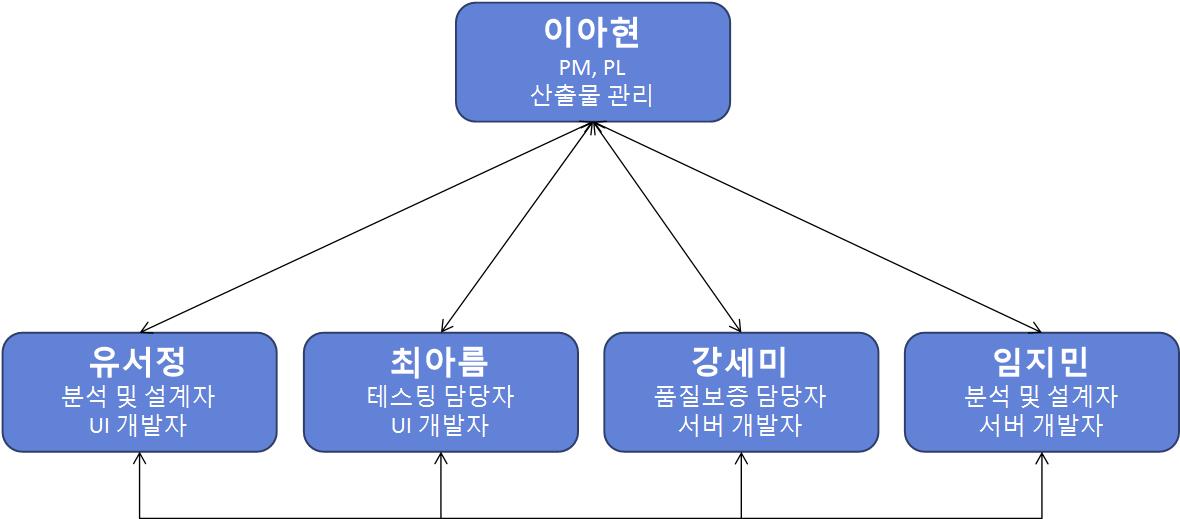
|  |
| --- |
| 프로젝트에서 주어진 주요 단계, 일정 별 산출물을 기술한다. |

|  |  |  |
| --- | --- | --- |
| 단계 | 일정 | 산출물 |
| 요구사항 분석 | 2021.04.02 ~ 2021.04.09 | 요구사항 명세서 |
| 프로젝트 계획 | 2021.04.12 ~ 2021.04.18 | 프로젝트 계획서 |
| 설계 | 2021.04.19 ~ 2021.05.03 | 설계 명세서 |
| 구현 | 2021.05.03 ~ 2021.05.30 | 소스코드 및 프로그램 |
| 테스팅 | 2021.05.31 ~ 2021.06.07 | 테스트 계획서, 테스트 보고서 |
| 유지보수 | 2021.06.07 ~ 2021.06.14 |  |

## 조직

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

### 조직도



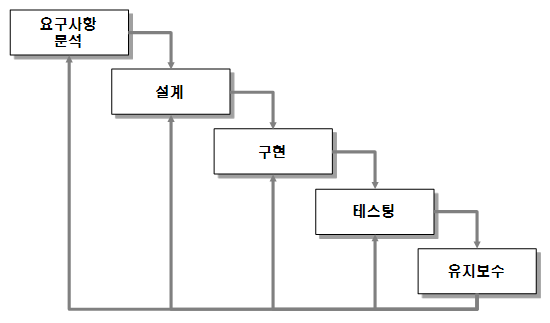
### 역할 및 책임

|  |  |  |
| --- | --- | --- |
| 팀원 | 역할 | 책임 |
| 이아현 | PM, PL, 산출물 관리 | - 프로젝트의 전반적인 계획 및 실행  - 프로젝트 산출물 관리 |
| 유서정 | 분석 및 설계자  UI 개발자 | - 요구사항에 맞게 프로그램 설계  - UI 개발 |
| 최아름 | 테스팅 담당자  UI 개발자 | - 테스트 기간동안 오류 관리  - UI 개발 |
| 강세미 | 품질보증 담당자  서버 개발자 | - 프로그램의 품질이 일정 수준 이상을 유지할 수 있도록 관리  - 서버 개발 |
| 임지민 | 분석 및 설계자  서버 개발자 | - 요구사항에 맞게 프로그램 설계  - 서버 개발 |

## 생명주기 모델

|  |
| --- |
| 프로젝트에 적용할 생명주기에 대한 설명을 기술한다. |

**폭포수 모델(Waterfall Model)**



소프트웨어 개발의 전 과정을 요구사항 분석, 설계, 구현, 테스팅, 유지보수로 나누어 체계적이고 순차적으로 접근하는 모델

|  |  |
| --- | --- |
| **단계** | **설명** |
| 요구사항 분석 | 개발하고자 하는 소프트웨어에 대한 요구사항 수집, 문제 이해 및 분석 |
| 설계 | 프로그램의 모든 시스템의 구조 결정 |
| 구현 | 설계 명세서를 시스템의 실제 모습으로 변환 |
| 테스팅 | 프로그램이 입력에 따라 요구되는 결과대로 작동하는지, 내부적 이상 여부 및 오류 발견을 위해 수행 |
| 유지보수 | 개발된 소프트웨어의 변경사항 수정 |

## 도구

|  |
| --- |
| 프로젝트에 적용할 도구에 대한 설명을 기술한다.  Ex) 일정관리 도구, 형상관리 도구 등.. |

1. Git & Github: 분산 버전 관리 시스템 & 원격 저장소

- 프로젝트 문서와 프로그램 소스코드 저장 및 관리

2. Visual Studio Code: 개발 툴

- 구현 과정에서 웹 환경 기준으로 VS code를 사용해서 개발

- React와 Node.js 사용

- Github와 연동하여 소스코드 저장

3. Notion: 회의록 작성 공간

- 회의를 가질 때마다 회의 내용을 정리

4. StarUML: 유스케이스 다이어그램 생성 프로그램

5. Oven: 웹 UI 테스트 및 설계 확인 가능한 프로그램

- 요구사항 명세서 작성 시나 설계 단계에서 어떤 식으로 구현할지를 미리 테스트해볼 수 있는 환경

6. MySQL workbench : DBMS

- 데이터를 모아서 편하게 정리해주는 DBMS

7. SmartDraw: Gantt Chart 작성 프로그램

8. Word: 워드 프로세서

- 산출물 문서 작성

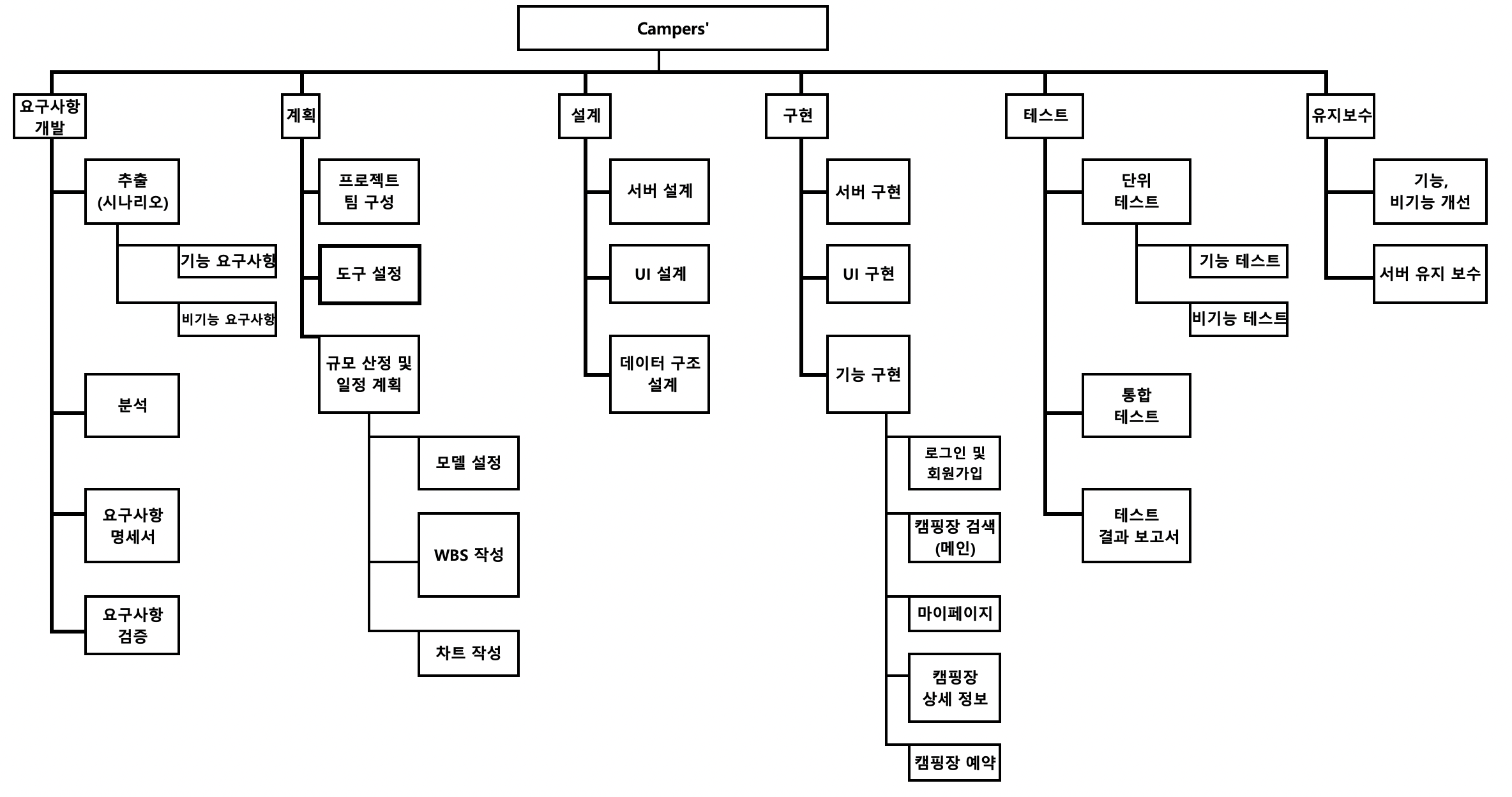
9. Excel: 엑셀

- WBS 작성

# 규모 산정

|  |
| --- |
| WBS를 작성하고, 각 작업에 소요되는 기간을 계산한다. |

## WBS(Work Breakdown Structure)

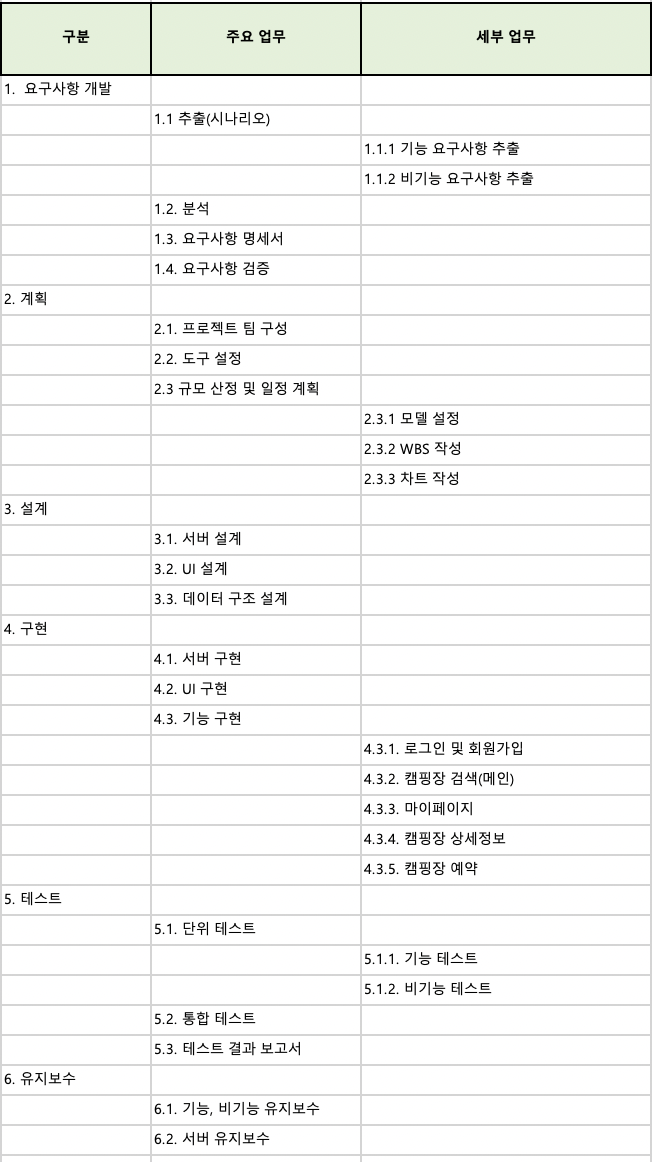


|  |  |  |
| --- | --- | --- |
| 단계 | 일정 | 기간 |
| 요구사항 분석 | 2021.04.02 ~ 2021.04.09 | 8일 |
| 프로젝트 계획 | 2021.04.12 ~ 2021.04.18 | 7일 |
| 설계 | 2021.04.19 ~ 2021.05.03 | 15일 |
| 구현 | 2021.05.03 ~ 2021.05.30 | 28일 |
| 테스팅 | 2021.05.31 ~ 2021.06.07 | 8일 |
| 유지보수 | 2021.06.07 ~ 2021.06.14 | 8일 |
| 총합 |  | 74일 |

# 일정

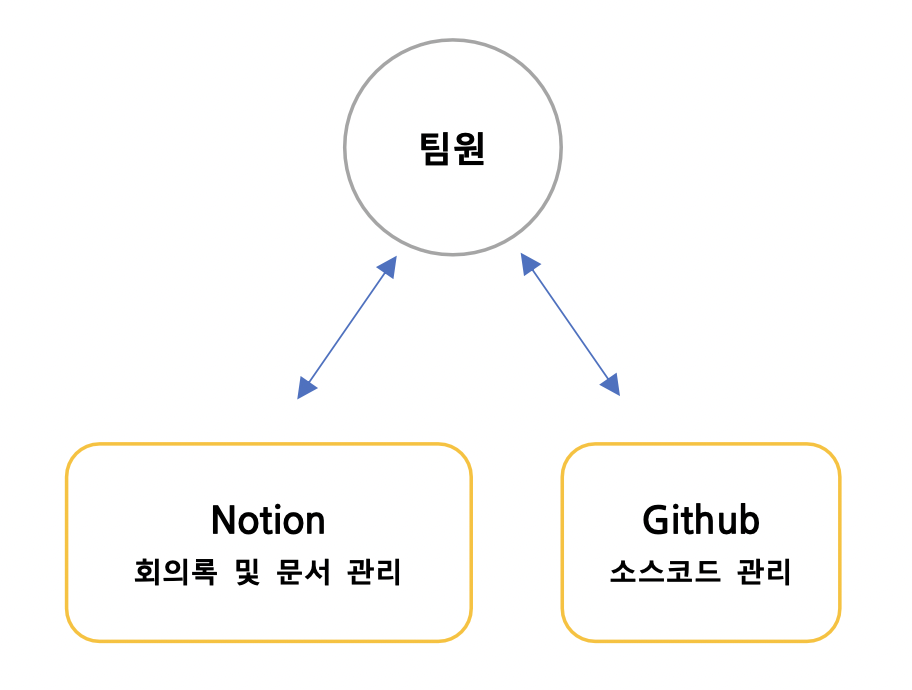
|  |
| --- |
| 주요 일정을 Gantt 차트와 PERT 차트를 이용하여 작성한다. |





# 산출물 관리

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



본인 역할에 맞는 산출물을 작성한 후, 팀원들 간 조율과 피드백을 거쳐 완성된 최종본을 정리하여 Github에 올린다. 후에 산출물 버전이 업데이트 될 때마다 깃허브 문서를 업데이트한다.

프로그램 구현 시 소스코드 또한 Github와 연동하여 올린다. 소스코드를 저장하는 폴더는 따로 제작하여 문서와 소스코드를 따로 관리하도록 한다.

회의록, 작업물의 경우 Notion을 통해 관리해서 지속적인 피드백과 체계적인 관리가 이루어지도록 한다.

[산출물 규칙]

1. 문서 이름은 문서명\_팀명\_버전으로 통일한다.

(ex) 요구사항 명세서\_Campers'\_ver1.2.doc

2. 소스코드 파일의 이름은 무슨 기능을 수행하는지 알기 쉽도록 기능명으로 짓도록 한다.

(ex) Login.js

# 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 |
| 요구사항 변경 | 중 | 상 | 요구사항이 변경됨에 따라 즉각 여러 산출물들을 수정한다. |
| 실력 부족 | 상 | 중 | 모르는 부분이 있을 때는 1차적으로는 검색을 한 뒤, 검색 내용이 이해되지 않거나 찾지 못했을 시 팀원들에게 도움을 요청하는 방식으로 진행한다. |
| 소통 부족 | 중 | 하 | 의견 불일치 시 서로 간의 의견 나눔 후 의견을 통일하거나 대안책을 내놓는다. |
| 구현 기간 부족 | 상 | 중 | 주요 기능 중심으로 구현하고, 부가 기능 구현 과정에서 시간 부족 시 품질을 낮추거나 미완성본을 제출한다. |
| 데이터 충돌 및 손실 | 하 | 중 | 서로 데이터가 충돌 또는 손실되지 않도록 파일 수정 전 팀원들에게 이 사실을 알리는 것을 의무로 한다. 또한 수정 완료 시에도 깃허브에 수정사항이 제대로 올라갔는지를 확인한 뒤, 팀원들에게 이 사실을 알린다. |
| 팀원들의 프로젝트에 대한 무관심 | 하 | 중 | 각자 어려운 부분이 있으면 즉각 소통하고, 의견 충돌에 있어서도 서운함을 갖지 않도록 각자의 의견을 존중하여 결과를 이끌어낸다. |
| PM의 프로젝트 관리 부족 | 중 | 상 | PM은 일정을 생각하며 각자에게 일을 잘 분배할 수 있도록 한다. 팀원들 또한 PM을 믿고 따른다. |