목차

[8. 테스트 계획 - 2 -](#_Toc71891781)

[8.1. 목적 - 2 -](#_Toc71891782)

[8.2. 테스트 정책 - 2 -](#_Toc71891783)

[8.2.1. 컴포넌트 테스팅(component testing) - 2 -](#_Toc71891784)

[8.2.2. 시스템 테스팅(system testing) - 2 -](#_Toc71891785)

[8.2.3. 인수 테스팅(acceptance testing) - 2 -](#_Toc71891786)

[8.2.4. 릴리즈 테스팅(release testing) - 3 -](#_Toc71891787)

[9. 개발 계획 - 3 -](#_Toc71891788)

[9.1. 목적 - 3 -](#_Toc71891789)

[9.2. 프론트엔드 개발환경 - 3 -](#_Toc71891790)

[9.2.1. 플러터(Flutter) - 3 -](#_Toc71891791)

[9.2.2. 어도비 포토샵(Adobe Photoshop) - 4 -](#_Toc71891792)

[9.2.3. 어도비 Xd(Adobe Xd) - 4 -](#_Toc71891793)

[9.3. 백엔드 개발환경 - 5 -](#_Toc71891794)

[9.3.1. 파이어베이스(Firebase) - 5 -](#_Toc71891795)

[9.3.2 깃허브(Github) - 5 -](#_Toc71891796)

[9.3.3 도커(Docker) - 6 -](#_Toc71891797)

[9.4. 제약 - 6 -](#_Toc71891798)

[9.5. 설계에서의 가정(assumptions)과 의존(dependencies) - 6 -](#_Toc71891799)

[10. 부가 정보 - 7 -](#_Toc71891800)

[10.1. 디자인 명세서 - 7 -](#_Toc71891801)

[10.2. 문서 히스토리 - 7 -](#_Toc71891802)

# 8. 테스트 계획

## 8.1. 목적

이 문단에서는 소프트웨어 개발 및 배포 단계에서 필요한 여러 테스트에 대해 설명한다. 수행하는 테스트는 크게 4개로, **컴포넌트 테스팅, 시스템 테스팅, 인수 테스팅, 릴리즈 테스팅**이다. 이 테스트들은 공통적으로 소프트웨어의 에러나 결점을 미리 발견하여 수정할 수 있게 만들어 준다. 그래서 궁극적으로, 안정적이고 믿을만한 소프트웨어를 시장 및 소비자에게 배포할 수 있게 해준다.

## 8.2. 테스트 정책

이 문단에서는 문단 8.1에서 언급한 4가지 테스트에 대해 설명하고자 한다. 이 중에서 컴포넌트 테스팅, 시스템 테스팅, 인수 테스팅은 개발 단계에서 이루어진다.

### 8.2.1. 컴포넌트 테스팅(component testing)

컴포넌트라는 것은 소프트웨어의 기본 단위로, 서로 관련이 있는 함수 혹은 객체들을 모아놓은 것을 말한다. 컴포넌트 테스팅이란 개별 컴포넌트를 다른 컴포넌트와 합치지 않고 독립적으로 테스트를 수행하는 것을 말한다. 이 테스트를 수행하면 소프트웨어가 더 복잡해지기 전에 버그를 빠르게 찾을 수 있어 개발 비용을 줄이는데 큰 도움이 된다.

### 8.2.2. 시스템 테스팅(system testing)

시스템 테스팅은 마지막 컴포넌트까지 통합하고 나서 만들어진 최종적인 시스템을 테스트하는 것이다. 이때 신뢰성(reliability), 성능(performance), 가용성(availability) 등 창발적 속성(emergent property)이 잘 드러나는지 테스트하는 것이 특히 중요하다. 또한 시스템의 기능을 꼼꼼하게 테스트할 수 있도록, 실제로 시스템에서 처리될 것으로 예상되는 데이터를 토대로 테스트 케이스가 작성되어야 한다.

### 8.2.3. 인수 테스팅(acceptance testing)

인수 테스팅은 시스템이 요구사항 명세서에서 제시된 사용자의 요구를 만족하는지 테스트하는 과정이다. 이 테스트는 실제 사용자의 데이터를 가지고 진행되어야 한다.

### 8.2.4. 릴리즈 테스팅(release testing)

많은 시간과 비용을 들여서 만든 소프트웨어를 실제로 시장에 배포하고, 소비자가 사용할 수 있게 하는 것은 전체 프로세스 중에서도 꽤 중요한 단계라고 할 수 있다. 지금까지의 테스트에서 문제가 없었더라도, 실제로 소프트웨어를 배포했을 때 예상치 못한 문제점이 생길 수도 있다. 릴리즈 테스팅은 새로운 버전의 소프트웨어가 출시되기 전에 문제점이나 결점이 없는지 테스트를 하는 것을 의미한다. 그리하여 소프트웨어가 출시되었을 때, 문제점이 없도록 하는 것이 목적이 된다.

또한 소프트웨어 사용자들에게 주기적으로 설문조사 혹은 리뷰를 요청해야 한다. 실제로 소프트웨어를 사용하는 유저들이 개발 단계에서는 몰랐던 문제점이나 아이디어를 발견할 수 있기 때문이다.

# 9. 개발 계획

## 9.1. 목적

이 문단에서는 소프트웨어 개발에 사용되는 여러 기술 및 개발환경에 대해 설명한다.

## 9.2. 프론트엔드 개발환경

### 9.2.1. 플러터(Flutter)



[Figure ] 플러터 로고

플러터는 크로스 플랫폼 앱개발을 지원하는 UI 프레임워크로, 이것으로 컴파일된 프로그램은 안드로이드와 IOS 기기에서 동시에 실행이 가능하다. 플러터에 사용되는 언어는 구글에 의해 만들어진 다트(Dart)라는 언어다. 플러터는 UI를 자체 렌더엔진인 스키아(Skia)로 직접 렌더링하여 뛰어난 성능을 보여준다. 여기서 플러터의 목적은 앱의 전체적인 인터페이스를 제작하는 것으로, 모든 유저의 입출력이 플러터를 거쳐가게 된다.

### 9.2.2. 어도비 포토샵(Adobe Photoshop)



[Figure ] 어도비 포토샵 로고

어도비 포토샵은 이미지의 합성과 편집에 특화된 프로그램으로, 픽셀을 기본 단위로 하는 비트맵 방식을 사용한다. 포토샵을 이용하여, 소프트웨어에 사용되는 직관적이면서도 아름다운 로고 및 아이콘을 디자인한다.

### 9.2.3. 어도비 Xd(Adobe Xd)



[Figure ] 어도비 Xd 로고

어도비 Xd는 UI/UX 디자인에 쓰이는 프로그램으로, 디자인에서 바로 프로토타입을 만들고 공유할 수 있어 원활한 커뮤니케이션을 돕고 빠른 소프트웨어 생산에 기여한다. 어도비 Xd를 이용하여, 유저가 어렵지 않게 프로그램을 사용할 수 있도록 직관적이고 깔끔한 UI/UX를 디자인한다.

## 9.3. 백엔드 개발환경

### 9.3.1. 파이어베이스(Firebase)



[Figure ] 파이어베이스 로고

파이어베이스는 고품질의 모바일 앱 및 웹사이트를 쉽게 만들 수 있게 도와주는 개발 플랫폼으로, 매우 다양한 기능을 제공한다. 대표적으로 로그인 인증, 클라우드 스토리지, 실시간 처리가 가능한 데이터베이스, 호스팅 등 백엔드 개발에 필요한 기능을 제공한다. 특히 실시간 데이터베이스를 통해, 모든 클라이언트에서 데이터가 동기화되어 안정적인 기능 제공이 가능하다.

### 9.3.2 깃허브(Github)



[Figure ] 깃허브 로고

깃허브는 코드 버전 관리 및 협업을 지원하는 웹호스팅 서비스다. 즉, 하나의 프로젝트를 여러 팀원과 수행할 때 안정적인 협업을 도움으로써 서로 다른 소프트웨어 컴포넌트 간의 원활한 통합을 돕는다.

### 9.3.3 도커(Docker)



[Figure ] 도커 로고

도커는 컨테이너를 기반으로 한 오픈소스 가상화 플랫폼이다. 도커는 다양한 프로그램 및 실행 환경을 하나의 컨테이너로 추상화한다. 대부분의 소프트웨어 서비스 환경은 컨테이너로 추상화될 수 있으며, 이들은 AWS, Google Cloud 등 어디에서든 도커만 설치되어 있다면 실행이 가능하다.

## 9.4. 제약

이 시스템은 이 문서에 언급된 컨텐츠를 바탕으로 디자인, 구현된다. 세부적인 디자인, 구현은 개발자에 의해 정해진다. 다음의 사항들은 세부적인 디자인, 구현에서의 제약 사항들이다.  
• 경로 탐색의 경우 5초 이상 걸리면 안된다.  
• 비용(저작권 등)을 지불해야 하는 소프트웨어는 가능하다면 사용을 피한다.  
• 가능하다면 오픈소스 소프트웨어를 사용한다.  
• 시스템 비용과 유지 비용을 고려한다.

• 유저 친화적인 소프트웨어를 개발한다.  
• 미래의 기술 발전을 고려하여 확장성을 높인 시스템을 개발한다.  
• 개발 환경 OS는 윈도우 10 이다.  
• 개발 툴은 안드로이드 스튜디오 4.1.3 버전과 Xcode 12.5 버전이다.  
• 구동을 위해 필요한 최소 안드로이드 버전은 Android version 6.0 (API 23), iOS 버전은 iOS 10이다.  
• 테스트를 위한 에뮬레이터 버전은 Android version 10 (API 29)이다.

## 9.5. 설계에서의 가정(assumptions)과 의존(dependencies)

이 문서는 모든 시스템이 안드로이드, ios 디바이스 및 오픈 소스를 이용하여 디자인, 구현된다는 가정을 토대로 작성되었다. 구동을 위해 필요한 버전은 안드로이드의 경우 Android version 6.0(API 버전 23) 이상, ios의 경우 ios 10 이상이다. 테스트에 쓰이는 에뮬레이터의 버전은 Android version 10(API 버전 29)다. 이 외의 다른 운영체제나 버전에서는 정상적으로 작동하지 않을 수 있다.

# 10. 부가 정보

## 10.1. 디자인 명세서

이 소프트웨어 디자인 명세서는 IEEE Recommendation (IEEE Recommended Practice for Software Design Description, IEEE-Std-1016)에서 제시하는 기준을 준수하여 작성되었다.

## 10.2. 문서 히스토리

[표 1] 문서 히스토리

|  |  |  |  |
| --- | --- | --- | --- |
| 일자 | 버전 | 설명 | 작성자 |
| 2021.05.13 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |