**Software Test Plans**

**Description**

**for**

**잇 클래스**

**Release 1.0**

**by 2조**

**12/17/2021**

**Revision History**

| **Name** | **Date** | **Reason for Change** | **Version** |
| --- | --- | --- | --- |
| 유수현 | 12/17/2021 | 계획서 작성 | Revision 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**1.**](https://docs.google.com/document/d/1F9S-gwasqMv1jwLIk429WAHQWtLA5r_GPx5mwjB4r6U/edit#heading=h.3znysh7) **Introduction(소개) 3**

1.1 범위 3

1.2 테스트 항목 3

1.3 목적 3

1.4 내부 전제조건 4

**2.** **테스트 베이시스(Test basis) 4**

**3.** **테스트 전략 (Test Strategy) 5**

**4.** **일정 계획 (Schedule Plan) 5**

4.1 개발일정 5

**5.** **프로젝트 리스크(Project risk) 6**

**6.** **결함 관리(Defect management) 6**

1. **개요**

* 1. **범위**
* **로그인/로그아웃**

**✔**회원가입 후 사용자가 학생 또는 수업관리자로서 로그인.

**✔**로그인 후 사용자가 로그아웃

* **개인일정**

**✔**학생 사용자는 캘린더에 개인 일정을 작성

**✔**작성한 일정을 수정, 삭제

* **수업**

**✔**수업관리자가 수업을 추가, 삭제, 변경

**✔**추가된 수업에 학생을 추가

**✔**학생의 수업 목록에 추가된 수업 표시

* **출석체크**

**✔**수업관리자는 수업에 추가된 학생의 출결을 수정함

**✔**학생은 출결표에서 편집된 출결을 확인

* 1. **테스트 항목**

잇클라스.App 테스트의 항목은 다음과 같다.

* 회원가입 및 로그인/로그아웃의 원활한 동작 충족
* 사용자 유형(학생/수업관리자)에 따라서 다른 인터페이스 구현
* 요구사항에 명시된 기능 충족 확인
* 요구사항에 명시된 성능 충족 확인
* 데이터베이스 정확한 CRUD 확인

잇클라스.App 테스트에서 제외되는 항목은 다음과 같다.

* 다양한 입력 데이터의 허용성
* 신체 장애자 접근성

* 1. **목적**

잇클라스.App의 목적은 학생들은 출결을 확인하며 자신의 일정을 작성하고 수업관리자는 자신의 수업을 만들어 수업에 추가되어 있는 학생들의 출결을 체크하는 것이다.

* 기대결과 및 실제결과 간 차이점 보고
* 요구사항에 적합한지 결정
  1. **내부 전제조건**

잇클라스.App 테스트 수행을 위해서는 다음과 같은 전제 조건이 필요하다.

* 테스트 실행은 테스트 대상이 성공적으로 테스트 시작 조건을 만족할 때 시작함
* 주요 기능의 정상 동작 여부
* 요구사항의 테스트 용이성 평가
* 선행 테스트 단계에서 발견된 결함의 80%이상 조치 완료
* 테스트 시작 조건은 테스트 시작일 전까지 충족되어야 함
* 테스트 상의 변경은 테스트 관리자에게 즉시 전달되어야 함
* 테스트 중단 조건

✔ 테스트 대상 빌드에서 사전에 발견된 결함의 70% 이상이 조치되지 않음

✔ 결함으로 인해 App의 주요 기능이 정상 동작하지 않음

* 테스트 재개 조건

✔ 테스트 중단 사유가 해결됨

1. **테스트 베이시스(Test basis)**

* 소프트웨어 요구사항 명세서(Software Requirements Specification)
* 각 기능 및 비기능적 요구사항
* 소프트웨어 디자인 명세서(Software Design Description)
  + App의 시스템 아키텍처
  + App의 상세 설계
  + App의 사용자 및 소프트웨어 인터페이스

1. **테스트 전략(Test strategy)**

* Unit testing
  + 개발 과정에서 프로그램 각 부분을 고립 시켜서 각 부분이 정확하게 작동하고 있음을 확인한다.
  + 문제가 발생 시 정확하고 잘못된 부분을 빠르게 확인하여 수정한다.
  + 해당 테스트를 통해서 프로그램의 안정성을 높이고 디버깅 시간을 단축시켜 개발 시간을 최대한 단축시킨다.
* Performance testing

다양한 입력값을 주어 부하를 걸어 App이 부하에 견딜 수 있는지 테스트한다. 해당 과정에서 App에 오류가 없는지 혹은 정상 작동되는지 확인하고 오류가 발생할 경우 수정한다.

1. **일정 계획**

**4.1 개발 일정**

| **활동** | **12/4** | **12/5** | **12/6** | **12/7** | **12/8** | **12/9** | **12/10** | **12/11** | **12/12** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **회원가입** |  |  |  |  |  |  |  |  |  |
| **로그인/**  **로그아웃** |  |  |  |  |  |  |  |  |  |
| **서버구축** |  |  |  |  |  |  |  |  |  |
| **데이터베이스 구축** |  |  |  |  |  |  |  |  |  |
| **개인 일정** |  |  |  |  |  |  |  |  |  |
| **수업 추가** |  |  |  |  |  |  |  |  |  |
| **수업 삭제** |  |  |  |  |  |  |  |  |  |
| **출석체크** |  |  |  |  |  |  |  |  |  |

1. **프로젝트 리스크(Project risk)**

본 프로젝트에서는 다음과 같은 리스크가 식별되었다.

* 출석 체크, 개인일정 동작이 원활하지 않을 수 있다.
* Android OS간 호환성에서 문제가 발생할 수 있다.
* 인터넷이 잡히지 않는 경우에 발/수신할 경우의 한계성.

**6. 결함 관리(Defect management)**

**결함 관리를 위해, 결함 발생시 아래 항목들을 필수로 한다.**

- 결함내용

- 테스트 케이스 식별번호

- 결함유형

- 발견일

- 심각도

- 우선순위(결함 수정의 우선순위)

- 시정조치예정일

- 수정담당자

- 재 테스트 결과

- 종료일