

**변 경 이 력**

| **버전** | **일자** | **변경 내역** | **작 성 자** |
| --- | --- | --- | --- |
| 1.0.0 | 2024.12.08 | 1차 작성 | 일조매 |
| 1.0.1 | 2024.12.14 | Class Diagram 수정 | 일조매 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

***1.*** ***High level architecture.................. 4***

***2.*** ***Class Diagrams for static view .....*** 5

***3.*** ***Sequence for dynamic view ..........*** 6

# 

# **1.** **High level architecture**

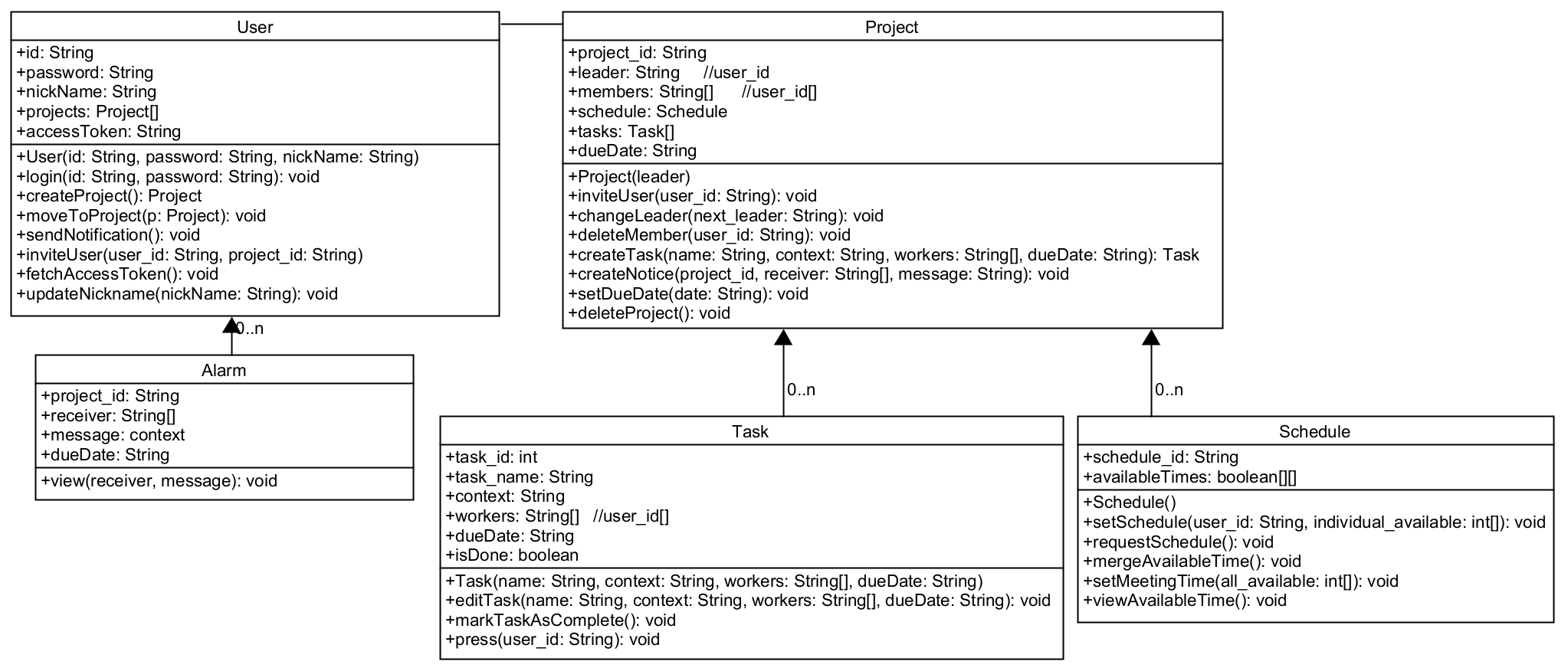
# 본 프로젝트는 High-level architecture를 낮은 비용으로 쉽게 구현하기 위해 single client-server 구조를 사용한다.

# 클라이언트에는 javascript 기반의 React, 서버에는 javascript 기반의 node js를 사용한다.

# 데이터베이스 관리에는 ORACLE 기반의 SQL developer를 사용한다.

# **2.** **Class Diagrams for static view**

* Class diagram을 만들 때 유의한 점은 ‘재사용성’이다. 특히 Data Base가 어떤 데이터를 가지고 있을지 가장 고려하였고, 사용하는 DB table을 중점으로 class를 만들었다.
* Actor의 경우, 팀원 actor와 팀장 actor가 분리되어 있으나, 프로젝트 설계상 로그인 시 user 이미 객체가 만들어지고 moveToProject() 함수가 실행된 후 팀장 혹은 팀원이 결정되므로 팀장 class, 팀원 class를 분리하지 않았다.



# **3.** **Sequence for dynamic view**

* 많은 use case 중, 구현이 복잡한 SCHE01(스케쥴 시간대 입력), SCHE02(회의시간 설정)을 하나의 sequence diagram으로 표현하였고, 본 프로젝트의 핵심 기능인 CAL01(Task 목록 및 독촉 알림 전송)을 하나의 sequence diagram으로 표현하였다.

| ID | 유스케이스 명 | 설명 | 우선 순위 |
| --- | --- | --- | --- |
| SCHE01 | 스케쥴  시간대 입력 | 팀원은 주간 스케줄에서 여유로운 시간대를 입력할 수 있다.  다른 팀원이 입력한 여유로운 시간대를 확인할 수 있다. | 1 |
| SCHE02 | 회의시간 설정 | 팀장은 여유로운 시간대를 바탕으로 회의시간을 설정할 수 있다.  설정한 회의시간은 캘린더에 기록된다. | 2 |
| CAL01 | Task 목록 및 독촉 알림 전송 | 팀원은 날짜를 선택하여 세부 Task를 확인, 특정 Task 독촉 알림을 전송할 수 있다. | 1 |

* High-level architecture로 single client-server 구조를 고려하고 있기에, actor가 시스템, 데이터베이스와의 상호작용하는 과정과 use case의 실패대안을 중심으로 diagram을 표현하였다.

