<School Security Linkage System>

<School Police Officer System>

Software Architecture Document

Version <1.0>

[SPOS : school Police officer system은 학교전담 경찰관이 학교 내에서 근무하면서 사용할 시스템이다. 웹 페이지와 하이브리드 앱을 사용할 시스템이며 PC로는 웹 페이지를 사용하고 모바일은 어플리케이션을 사용하는 시스템이다. 이 시스템 사용자는 학교전담 경찰관 뿐만 아니라 학교 선생님, 학교 학생, 학부모가 사용하는 시스템이다.한 명의 전담 경찰관이 5일 동안 5개의 학교에 랜덤으로 출근을 하며 출근한 해당 학교 학생 및 선생님의 정보만을 열람할 수 있다. (A학교 출근 시 B학교 정보 열람에 제약)]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <2025/03/29> | <1.0> | <학교전담 경찰관 전용 페이지 디자인> | <김예슬> |
|  |  | <학교 전담 경찰관 – 교육부 DB연결?> |  |
|  |  |  |  |
|  |  |  |  |

목차

1. 개요 2

1.1 목적 2

1.2 범위 2

1.3 내용 설명

1.4 상세요구 2

1.5 참고 문헌 2

2. Architectural Representation 2

3. Architectural Goals and Constraints 2

4. Use-Case View 2

4.1 Use-Case Realizations 2

5. Logical View 2

5.1 Overview 2

5.2 Architecturally Significant Design Packages 2

6. Process View 2

7. Deployment View 2

8. Implementation View 2

8.1 Overview 2

8.2 Layers 2

9. Data View (optional) 2

10. Size and Performance 2

11. Quality 2

소프트웨어 구조 설계

# 개요

[학교전담 경찰관이 학교 상주근무 할 때에 근무 기록을 용이 하게 도우며 학교내외에서 일어나는 사건을 빠르게 해결하거나 관할 경찰서로 인계하는 시스템이다. 이외에도 교권침해나 가정 폭력 등]

## 목적

SPOS의 목적은 학교 전단 경찰관마다 5개의 학교를 전담하며 해당 학교에서 상주하며 학교 내외에서 일어나는 사건에 즉각 대응이 쉽게 하며

[This section defines the role or purpose of the **Software Architecture Document**, in the overall project documentation, and briefly describes the structure of the document. The specific audiences for the document should be identified, with an indication of how they are expected to use the document.]

## 범위

[

## 내용 설명

[

## 지원 정보

[

## 참고 문헌

[]

# Architectural Representation

[This section describes what software architecture is for the current system, and how it is represented. Of the **Use-Case**, **Logical**, **Process**, **Deployment**, and **Implementation Views**, it enumerates the views that are necessary, and for each view, explains what types of model elements it contains.]

# Architectural Goals and Constraints

[This section describes the software requirements and objectives that have some significant impact on the architecture, for example, safety, security, privacy, use of an off-the-shelf product, portability, distribution, and reuse. It also captures the special constraints that may apply: design and implementation strategy, development tools, team structure, schedule, legacy code, and so on.]

# Use-Case View

[This section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the final system, or if they have a large architectural coverage - they exercise many architectural elements, or if they stress or illustrate a specific, delicate point of the architecture.]

## Use-Case Realizations

[This section illustrates how the software actually works by giving a few selected use-case (or scenario) realizations, and explains how the various design model elements contribute to their functionality.]

# Logical View

[This section describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages. And for each significant package, its decomposition into classes and class utilities. You should introduce architecturally significant classes and describe their responsibilities, as well as a few very important relationships, operations, and attributes.]

## Overview

[This subsection describes the overall decomposition of the design model in terms of its package hierarchy and layers.]

## Architecturally Significant Design Packages

[For each significant package, include a subsection with its name, its brief description, and a diagram with all significant classes and packages contained within the package.

For each significant class in the package, include its name, brief description, and, optionally a description of some of its major responsibilities, operations and attributes.]

# Process View

[This section describes the system's decomposition into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). Organize the section by groups of processes that communicate or interact. Describe the main modes of communication between processes, such as message passing, interrupts, and rendezvous.]

# Deployment View

[This section describes one or more physical network (hardware) configurations on which the software is deployed and run. At a minimum for each configuration it should indicate the physical nodes (computers, CPUs) that execute the software, and their interconnections (bus, LAN, point-to-point, and so on.) Also include a mapping of the processes of the **Process View** onto the physical nodes.]

# Implementation View

[This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.]

## Overview

[This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers. ]

## Layers

[For each layer, include a subsection with its name, an enumeration of the subsystems located in the layer, and a component diagram.]

# Data View (optional)

[A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data, or the translation between the Design Model and the Data Model is trivial.]

# Size and Performance

[A description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.]

# Quality

[A description of how the software architecture contributes to all capabilities (other than functionality) of the system: extensibility, reliability, portability, and so on. If these characteristics have special significance, for example safety, security or privacy implications, they should be clearly delineated.]