<약속관리 어플>

For <Subsystem or Feature>

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **<25/03/2025>** | **<0.1>** | **<문서 작성>** | **<조용무>** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Overall Description 3

3. Specific Requirements 3

3.1 Functionality 3

3.1.1 <Functional Requirement One> 3

3.2 Usability 3

3.2.1 <Usability Requirement One> 3

3.3 Reliability 3

3.3.1 <Reliability Requirement One> 3

3.4 Performance 3

3.4.1 <Performance Requirement One> 3

3.5 Supportability 3

3.5.1 <Supportability Requirement One> 3

3.6 Design Constraints 3

3.6.1 <Design Constraint One> 3

3.7 Online User Documentation and Help System Requirements 3

3.8 Purchased Components 3

3.9 Interfaces 3

3.9.1 User Interfaces 3

3.9.2 Hardware Interfaces 3

3.9.3 Software Interfaces 3

3.9.4 Communications Interfaces 3

3.10 Licensing Requirements 3

3.11 Legal, Copyright and Other Notices 3

3.12 Applicable Standards 3

4. Supporting Information 3

# Introduction

이 문서는 약속관리 어플리케이션에 대한 소프트웨어 요구사항 명세서입니다. 약속관리 어플리케이션은 사용자가 개인 및 그룹 일정을 효율적으로 관리하며, 상대방과 또는 그룹 모임에서 늦는 상황 방지, 또는 상대방의 위치를 모른 체 하염없이 기다리는 일을 방지하는데 도움을 주는 어플리케이션입니다.

주요 기능:

* 일정 등록 및 수정
* 알림 기능
* 지도 및 상대방과 사용자의 위치 확인 기능
* (기능 추가 예정)

## Purpose

이 어플리케이션의 개발 사용자와 상대방 간의 약속을 효율적으로 조율하고, 상대방의 위치를 실시간으로 확인할 수 있는 기능을 제공하는 것입니다. 사용자가 약속을 잡고, 상대방의 위치를 파악함으로써, 약속에 늦거나 상대방이 지각할 경우, 기다리는 시간을 더욱 효율적으로 활용할 수 있습니다. 이 어플리케이션은 약속의 정확한 시간 관리와 효율적인 일정 조율을 돕는 동시에, 사용자에게 불필요한 대기 시간을 최소화하고 일정의 유연성을 높이는 기능을 제공합니다.

## Scope

이 문서에서 다루는 약속 관리 어플리케이션은 개인 사용자와 그룹 사용자를 대상으로 하며, 모바일 플랫폼에서 실행됩니다. 사용자는 본 어플리케이션을 통해 상대방과의 약속을 설정하고, 약속 시간에 대한 알림을 받으며, 상대방의 실시간 위치를 추적할 수 있습니다. 또한, 약속이 지각이 되었을 때, 기다리는 시간을 효율적으로 활용할 수 있는 기능이 포함됩니다.

## Definitions, Acronyms and Abbreviations

[This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the **SRS**.  This information may be provided by reference to the project Glossary.]

## References

[This subsection should provide a complete list of all documents referenced elsewhere in the **SRS**. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

## Overview

[This subsection should describe what the rest of the **SRS** contains and explain how the document is organized.]

# Overall Description

[This section of the **SRS** should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

• product perspective

• product functions

• user characteristics

• constraints

• assumptions and dependencies

• requirements subsets]

# Specific Requirements

[This section of the **SRS** should contain all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.   When using use-case modeling, these requirements are captured in the Use-Cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section, as shown below.]

## Functionality

[This section describes the functional requirements of the system for those requirements which are expressed in the natural language style. For many applications, this may constitute the bulk of the **SRS** Package and thought should be given to the organization of this section. This section is typically organized by feature, but alternative organization methods may also be appropriate, for example, organization by user or organization by subsystem. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modeling tools, etc., are employed to capture the functionality, this section document will refer to the availability of that data, indicating the location and name of the tool that is used to capture the data.]

### <Functional Requirement One>

[The requirement description.]

## Usability

[This section should include all of those requirements that affect usability. For example,

• specify the required training time for a normal users and a power user to become productive at particular operations

• specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like

• specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]

### <Usability Requirement One>

[The requirement description goes here.]

## Reliability

[Requirements for reliability of the system should be specified here. Some suggestions follow:

• Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, etc.

• Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.

• Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?

• Accuracy—specify precision (resolution) and accuracy (by some known standard) that is required in the system’s output.

• Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand of lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).

• Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]

### <Reliability Requirement One>

[The requirement description.]

## Performance

[The system’s performance characteristics should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

• response time for a transaction (average, maximum)

• throughput, for example, transactions per second

• capacity, for example, the number of customers or transactions the system can accommodate

• degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)

• resource utilization, such as memory, disk, communications, etc.

### <Performance Requirement One>

[The requirement description goes here.]

## Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.]

### <Supportability Requirement One>

[The requirement description goes here.]

## Design Constraints

[This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, etc.]

### <Design Constraint One>

[The requirement description goes here.]

## On-line User Documentation and Help System Requirements

[Describes the requirements, if any, for on-line user documentation, help systems, help about notices, etc.]

## Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc. so that the software can be developed and verified against the interface requirements.]

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, etc. ]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this **SRS** but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, etc.]

## Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notice, wordmark, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, etc.]

# Supporting Information

[The supporting information makes the **SRS** easier to use. It includes:

• Table of contents

• Index

• Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the **SRS** should explicitly state whether or not the appendices are to be considered part of the requirements.]