**<Virtual Room Reservation>**

**Software Architecture Document**

**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).]*

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Contents

[1.                  Introduction 2](#_Toc26565130)

[1.1               Purpose 2](#_Toc26565131)

[1.2               Scope 2](#_Toc26565132)

[1.3Definitions, Acronyms and Abbreviations 2](#_Toc26565133)

[1.4               References 3](#_Toc26565134)

[1.5               Overview 3](#_Toc26565135)

[2.Architectural Representation 3](#_Toc26565136)

[3.Architectural Goals and Constraints 3](#_Toc26565137)

[4.                  Use-Case View 3](#_Toc26565138)

[4.1Use-Case Realizations 3](#_Toc26565139)

[5.                  Logical View 3](#_Toc26565140)

[5.1               Overview 4](#_Toc26565141)

[5.2Architecturally Significant Design Packages 4](#_Toc26565142)

[6.                  Process View 4](#_Toc26565143)

[7.                  Deployment View 4](#_Toc26565144)

[8.                  Implementation View 4](#_Toc26565145)

[8.1               Overview 4](#_Toc26565146)

[8.2               Layers 4](#_Toc26565147)

[9.                  Data View (optional) 5](#_Toc26565148)

[10.Size and Performance 5](#_Toc26565149)

[11.             Quality 5](#_Toc26565150)

**Software Architecture Document**

# 1.                  Introduction

*[The introduction of the****Software Architecture Document****should provide an overview of the entire****Software Architecture Document****. It should include the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the****Software Architecture Document****.]*

## Purpose

This document provides a comprehensive architectural overview of the Virtual Room Reservation system. It captures and conveys significant architectural decisions and views of the system to facilitate understanding and guide further development.

*[This section defines the 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.]*

## Scop

This document covers the software architecture of the Virtual Room Reservation system, detailing the design and structural layout. It applies to all components of the system, including user interfaces, data management, and server-side processing.

*[A brief description of what the Software Architecture Document applies to; what is affected or influenced by this document.]*

## Definitions, Acronyms and Abbreviations

* VRR: Virtual Room Reservation
* UI: User Interface
* DB: Database

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

## References

Software engineering textbook.

*[This subsection should provide a complete list of all documents referenced elsewhere in the****Software Architecture Document****.  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 document is organized to provide a detailed view of the system's architecture, including use-case realizations, logical view, process view, deployment view, and implementation view.

*[This subsection should describe what the rest of the****Software Architecture Document****contains and explain how the****Software Architecture Document****is organized.]*

# 2.Architectural Representation

The architecture of the VRR system is represented through several views: Use-Case, Logical, Process, Deployment, and Implementation. Each view focuses on specific aspects of the system, such as user interactions, system design, data flow, physical deployment, and code structure.

*[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.]*

# 3.Architectural Goals and Constraints

The architecture is designed with a focus on modularity, scalability, and security. Constraints include the use of specific technologies (e.g., a particular DBMS), adherence to security standards, and compatibility with existing institutional systems.

*[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.]*

# 4.                  Use-Case View

Key use cases include room reservation, user management, and schedule visualization. These represent central functionalities of the system, involving several architectural components.

*[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.]*

## 4.1Use-Case Realizations

Realizations demonstrate how the system fulfills use cases like user registration, room booking, and conflict resolution.

*[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.]*

# 5.                  Logical View

The design model is decomposed into subsystems such as User Management, Reservation Processing, and Data Handling. Each subsystem contains packages and classes with defined responsibilities.一張含有 圖表, 行, 白色 的圖片

自動產生的描述

*[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.]*

## 5.1               Overview

The system is organized into layers, each containing packages related to specific functionalities like UI, business logic, and data access.

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

## 5.2Architecturally Significant Design Packages

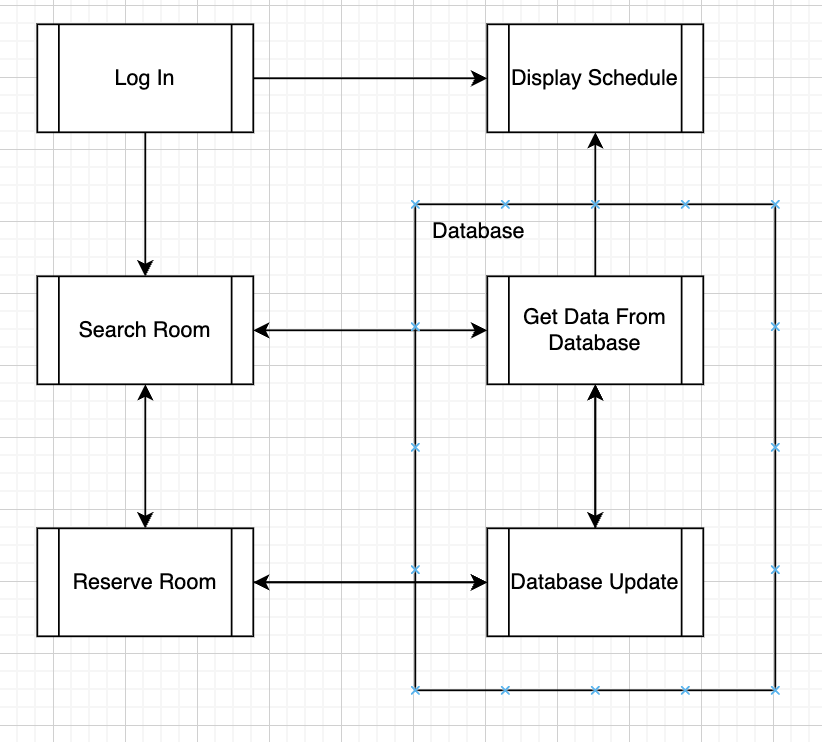
User Management: Handles user authentication, registration, and profile management.

Reservation Processing: Manages room booking, scheduling, and conflict checking.

*[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.]*

# 6.                  Process View

Describes the system's decomposition into processes and threads, focusing on communication between these elements, such as through APIs or database queries.

*[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.]*

# 7.                  Deployment View

Illustrates the physical deployment of the system on servers and clients, including the network configuration and hardware specifications.

*[This section describes one or more physical network (hardware) configurations on which the software is deployed and run. It is a view of the Deployment Model. 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.]*

# 8.                  Implementation View

Details the system's implementation structure, including the separation into layers such as the presentation layer, business logic layer, and data access layer.

*[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.]*

## 8.1               Overview

Defines the contents of each layer and the rules governing their interaction.

*[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. ]*

## 8.2               Layers

Each layer is described, detailing the subsystems and components it contains, along with their relationships.

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

# 9.                  Data View (optional)

Covers the persistent data storage aspect, detailing the database schema and data flow between the system and the database.

*[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.]*

# 10.Size and Performance

Specifies the software's performance criteria, such as response times, and outlines the scalability requirements.

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

# 11.             Quality

Describes how the architecture supports system qualities like reliability, security, and maintainability, highlighting any critical areas such as data protection or user privacy.

*[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.]*