

# **Software Configuration Management Plan**

**Project:** Real-Time Queue & Appointment Optimizer System

## **1. Introduction**

This document defines the configuration management activities for the Real-Time Queue & Appointment Optimizer System. Configuration management is an important part of software development because it helps in organizing, controlling, and maintaining all project-related items in a structured manner. These items include source code, database files, design documents, test plans, and user manuals.

The main objective of this plan is to ensure that every change made during the development of the project is properly documented and approved before being applied. This reduces confusion, avoids accidental overwriting of files, and ensures that the team always works on the correct and latest version of the system. By following configuration management practices, the project can be developed in a disciplined and reliable way.

## **2. Configuration Identification**

Configuration identification involves selecting and labeling all important items related to the project. In this system, configuration items include program source code, database scripts, user interface files, requirement documents, design diagrams, and test case documents.

Each configuration item will be assigned a unique name and version number. Version numbers such as v1.0, v1.1, and v2.0 will be used to identify different releases of the system. This helps the team understand which version is currently in use and which version contains recent updates. Proper folder structure and file naming conventions will also be maintained so that files can be easily located and managed.

This approach helps avoid duplication of files and ensures that old versions can be restored if needed.

## **3. Configuration Control**

Configuration control ensures that all changes to the project are managed in an organized and secure way. Any change request related to the software or documentation will be first reviewed by the project team. The impact of the change will be analyzed in terms of time, cost, and system behavior before approval.

Only authorized team members will be allowed to modify the main system files. Once a change is approved, it will be implemented carefully and tested before being merged into the main version of the project. This process helps in preventing unauthorized changes and reduces the chances of introducing errors into the system.

By following this control process, the project remains stable and well-managed throughout its development life cycle.

#### **4. Configuration Status Accounting**

Configuration status accounting is used to track and report the status of configuration items. A change log will be maintained to record details such as what change was made, who made the change, and the date of modification. This provides a complete history of project updates and improvements.

Regular reports will be prepared to show the current version of the software, the list of completed changes, and pending updates. This information helps the team understand the progress of the project and identify any issues related to version management.

Maintaining proper records also helps in preparing documentation and presentations for project evaluation and future reference.

#### **5. Configuration Audits**

Configuration audits are performed to verify that all configuration items are correct and consistent with project requirements. These audits ensure that only approved and tested changes are included in the final system version.

During an audit, the team checks whether the correct files are stored in the repository, whether the version numbers are properly assigned, and whether documentation matches the actual system implementation. This process also helps in detecting missing files or unauthorized changes.

Periodic audits increase the reliability of the system and confirm that the project is being developed according to defined standards and procedures.