



California  
**DEPARTMENT OF TECHNOLOGY**  
*Consulting & Planning Division*

# NON- FUNCTIONAL REQUIREMENTS TEMPLATE

**VERSION 1.0**

This template was created to enable departments to more easily develop their project plans. The Department of Technology, Consulting and Planning Division, created this template based on its experiences. The template relies on industry best practices combined with decades of experience on California state information technology projects. The way it was structured is to enable a department to complete the information related to its project without having to write background information related to the discipline. A department may use as much or as little of the template as it wishes.

#### **Template Instructions:**

- ***Instructions for completing*** this template – written for the author of the project plan - are encased in [ ] and the text is ***italicized and bolded***.
- *Examples* are provided as a guideline to the type of sample information presented in each section and the text is *italicized*.
- Boilerplate standard language for each section is written in the document font and may be used or modified, as necessary.
- A department's project specific information goes within the brackets << >>.
- *Informational text is italicized* within square brackets [ ] for informational purposes to the person who has to create the plan and includes background information, explanation, rationale, etc.

# DOCUMENT HISTORY

DOCUMENT REVISION HISTORY			
DATE	DOCUMENT VERSION	REVISION DESCRIPTION	AUTHOR
04/01/2026	1.0	INITIAL VERSION OF NON- FUNCTIONAL REQUIREMENTS DOCUMENT FOR THE ATTENDANCE MANAGEMENT SYSTEM	PROJECT TEAM

## TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>2. OVERVIEW .....</b>	<b>1</b>
<b>3. REFERENCED DOCUMENTS.....</b>	<b>2</b>
<b>4. NON-FUNCTIONAL REQUIREMENTS .....</b>	<b>2</b>
4.1. PERFORMANCE.....	3
4.1.1. USER RESPONSIVENESS .....	3
4.1.2. PEAK PERIOD LOAD .....	3
4.1.3. BATCH PROCESSING.....	4
4.2. SECURITY.....	4
4.2.1. ACCESS CONTROL & AUDITING .....	4
4.2.2. AUTHENTICATION SECURITY.....	5
4.3. COMPATIBILITY.....	5
4.3.1. CROSS-BROWSER COMPATIBILITY .....	5
4.4. MAINTAINABILITY .....	5
4.4.1. ERROR REPORTING.....	5
4.5. OPERATIONAL.....	6
4.5.1. SEMESTER ROLLOVER STABILITY.....	6
 <b>APPENDIX A. CONSIDERATIONS.....</b>	 <b>1</b>

## 1. INTRODUCTION

Non-functional requirements define the quality attributes, constraints, and operational conditions under which the **University Staff Attendance Management System** must operate. These requirements do not describe system functionality; instead, they specify how well the system must perform and under what conditions it must operate.

This document specifies the non-functional requirements for the University Staff Attendance Management System, as derived from project stakeholders and the approved Business Requirements Document (BRD). The requirements are structured using a standardized scenario-based format to ensure clarity, measurability, and verifiability.

The system is a custom-developed web application used to manage teacher and staff attendance, leave requests, and semester-based schedules in a university environment.

## 2. OVERVIEW

This document defines the non-functional requirements for the University Staff Attendance Management System using a structured, scenario-based approach. The purpose of this approach is to ensure that each requirement is clearly contextualized, measurable, verifiable, and aligned with real operational conditions within a university environment.

The non-functional requirements in this document address quality attributes such as performance, security, compatibility, maintainability, and operational stability. These requirements place constraints on how the system is designed, implemented, and operated over multiple academic semesters and years.

This system is intentionally limited in scope. It does not include student attendance, payroll, biometric devices, mobile applications, external system integrations, or notification services. The non-functional requirements in this document apply only to the attendance, leave management, scheduling, and reporting features described in the approved Business Requirements Document (BRD).

Each non-functional requirement in this document is defined using the following sections:

- **Scenario** – Describes the operational situation or system condition under which the requirement applies
- **Requirement** – Defines the measurable quality attribute the system must satisfy
- **Constraints** – Identifies boundaries, limitations, or assumptions that affect the requirement
- **Verification Method** – Specifies how the requirement will be tested or validated
- **Notes** – Provides additional clarification to support consistent understanding

This structure ensures that all non-functional requirements in this document are testable, traceable, and suitable for formal review.

ID#	UNIQUE	"BRIEF DESCRIPTIVE NAME FOR THE REQUIREMENT"
Scenario:		<ul style="list-style-type: none"> <li>• Who or what initiates the scenario</li> <li>• The event that initiates the scenario</li> <li>• The system or environmental conditions (e.g., normal operations, shutting down)</li> <li>• Which part of system, or whole, is involved</li> <li>• How is the system being stressed</li> </ul>
Requirement:		What noticeable event happens as a result of the scenario
Constraints:		Limitations, boundaries, other conditions that must be considered
Verification Method:		Describe how the Requirement can be tested and verified
Notes:		Additional information to help communicate the needs.

### 3. REFERENCED DOCUMENTS

The following documents were referenced and used in creating this deliverable; they also provide definitions and background information related to non-functional requirements and attendance management systems.

In addition to the academic references listed below, this document is primarily derived from the approved Business Requirements Document (BRD) of the University Staff Attendance Management System.

- Obansola, A. A., Makinde, O. S., Adeshina, S. A., & Adebayo, A. A. (2016). Development of staff attendance management system using fingerprint biometric identification technique. International Journal of Computer Applications, 139(11), 10–15.
- Narangamma, S., & Shamila, F. (2025). QR code based attendance management system. Journal of Information Systems and Technology, 8(2), 45–52.
- Lateef, O. M., & Kamil, A. A. (2023). Facial recognition technology-based attendance management system application in smart classroom. International Journal of Smart Education and Learning, 5(1), 1–12.

### 4. NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements for this project are grouped into the following categories:

- Performance
- Security
- Compatibility
- Maintainability
- Operational

## 4.1. PERFORMANCE

### 4.1.1. USER RESPONSIVENESS

ID#	NF-PU-01	NORMAL ACADEMIC-DAY LOAD
Scenario:	<ul style="list-style-type: none"> <li>During a normal academic day, teachers, class moderators, HR assistants, and Heads of Department access the system to mark attendance, view schedules, submit leave requests, and review reports. The system is operating under normal conditions with no batch processing in progress.</li> </ul>	
Requirement:	The system shall respond to 95% of user-initiated requests within 500 milliseconds and no request shall exceed 2 seconds.	
Constraints:	<ul style="list-style-type: none"> <li>Network latency outside the university infrastructure is excluded from response-time measurement.</li> <li>Measurements apply from API request receipt to response dispatch.</li> </ul>	
Verification Method:	<ul style="list-style-type: none"> <li>Simulate concurrent users using load-testing tools.</li> <li>Measure response times at the API gateway.</li> <li>Verify compliance with defined thresholds.</li> </ul>	
Notes:	This requirement ensures usability during peak teaching hours.	

### 4.1.2. PEAK PERIOD LOAD (CLASS START TIMES)

ID#	NF-PU-02	PEAK PERIOD LOAD
Scenario:	<ul style="list-style-type: none"> <li>At class start times, multiple class moderators and HR assistants mark attendance concurrently. The system experiences short-duration peak load.</li> </ul>	
Requirement:	The system shall maintain response times under 1 second for attendance marking operations during peak periods.	
Constraints:	<ul style="list-style-type: none"> <li>Peak duration assumed to be less than 30 minutes.</li> </ul>	
Verification Method:	<ul style="list-style-type: none"> <li>Stress testing with simulated peak concurrency.</li> </ul>	

ID#	NF-PU-02	PEAK PERIOD LOAD
	Notes:	<ul style="list-style-type: none"> <li>Attendance marking is a critical, time-sensitive operation.</li> </ul>

#### 4.1.3. BATCH PROCESSING

ID#	NF-PB-01	BATCH PROCESSING
	Scenario:	<ul style="list-style-type: none"> <li>At 23:59 daily, the system executes an automated batch process to mark all unrecorded attendance entries as Absent.</li> </ul>
	Requirement:	The batch process shall complete within <b>2 minutes</b> without impacting user access.
	Constraints:	<ul style="list-style-type: none"> <li>Batch processing runs once per day.</li> </ul>
	Verification Method:	<ul style="list-style-type: none"> <li>Execute batch job on a dataset representing one semester.</li> <li>Measure execution duration.</li> </ul>
	Notes:	<ul style="list-style-type: none"> <li>This batch process implements the system rule that all unmarked attendance records become Absent at 23:59.</li> </ul>

## 4.2. SECURITY

### 4.2.1. ACCESS CONTROL & AUDITING

ID#	NF-SA-01	ACCESS CONTROL & AUDITING
	Scenario:	Users with different roles (Admin, HoD, HR, Class Moderator, Teacher, Staff) access the system to perform authorized actions.
	Requirement:	The system shall enforce role-based access control (RBAC) and log all critical operations including leave approvals, attendance marking, and semester creation.
	Constraints:	<ul style="list-style-type: none"> <li>Each user must have a unique identifier.</li> <li>Logs must be immutable.</li> </ul>
	Verification Method:	<ul style="list-style-type: none"> <li>Attempt authorized and unauthorized actions per role.</li> <li>Review audit logs for correctness and completeness.</li> </ul>
	Notes:	Ensures accountability and prevents privilege misuse.



#### 4.2.2. AUTHENTICATION SECURITY

ID#	NF-SA-02	AUTHENTICATION SECURITY
Scenario:	<i>Users authenticate to access the system.</i>	
Requirement:	<i>All passwords shall be securely hashed and no plaintext credentials shall be stored or transmitted. The system uses standard web authentication mechanisms and role-based authorization.</i>	
Constraints:	<ul style="list-style-type: none"><li>• <i>Industry-standard hashing algorithms must be used.</i></li></ul>	
Verification Method:	<ul style="list-style-type: none"><li>• <i>Inspect database records.</i></li><li>• <i>Review authentication source code.</i></li></ul>	
Notes:	<i>Aligns with standard security best practices.</i>	

#### 4.3. COMPATIBILITY

##### 4.3.1. CROSS-BROWSER COMPATIBILITY

ID#	NF-CB-01	CROSS-BROWSER COMPATIBILITY
Scenario:	<i>Users access the system via modern web browsers on desktop devices.</i>	
Requirement:	<i>The system shall not attempt or rely on the ability to download “add-in” that will be required to execute of the public user’s client machine.</i>	
Constraints:	<ul style="list-style-type: none"><li>• <i>No browser plugins or extensions may be required.</i></li></ul>	
Verification Method:	<ul style="list-style-type: none"><li>• <i>Manual and automated browser testing.</i></li></ul>	
Notes:	<i>The system is desktop-first.</i>	

#### 4.4. MAINTAINABILITY

##### 4.4.1. ERROR REPORTING

ID#	NF-MA-01	ERROR REPORTING
Scenario:	<i>Application or database errors occur during system operation.</i>	
Requirement:	<i>The system shall log errors with sufficient detail to support debugging while presenting user-friendly error messages.</i>	
Constraints:	<ul style="list-style-type: none"><li>• <i>Internal error details must not be exposed to end users.</i></li></ul>	

ID#	NF-MA-01	ERROR REPORTING
Verification Method:	<ul style="list-style-type: none"> <li>Induce controlled faults and inspect logs and UI messages.</li> </ul>	
Notes:	Supports long-term maintainability.	

## 4.5. OPERATIONAL

### 4.5.1. SEMESTER ROLLOVER STABILITY

ID#	NF-OP-01	SEMESTER ROLLOVER STABILITY
Scenario:	At the start of a new semester, administrators configure new schedules and classes.	
Requirement:	The system shall allow new semester creation without modifying or deleting historical data.	
Constraints:	<ul style="list-style-type: none"> <li>Historical attendance records must be immutable.</li> </ul>	
Verification Method:	<ul style="list-style-type: none"> <li>Create a new semester and validate historical data integrity.</li> </ul>	
Notes:	Critical for multi-year university operation.	

## APPENDIX A. CONSIDERATIONS

CONSIDERATIONS FOR COTS, MOTS, AND CUSTOM IMPLEMENTATION	
<b>COTS</b>	<ul style="list-style-type: none"><li>For COTS products, many non-functional requirements are inherently built into the product and cannot be modified. Only non-functional requirements related to the hosting environment (such as availability, performance, and certain security controls) would typically apply. Internal qualities such as code maintainability and internal security mechanisms are largely outside the project's control.</li></ul>
<b>MOTS</b>	<ul style="list-style-type: none"><li>For MOTS solutions, non-functional requirements apply only to the modified components of the system. Unmodified components follow the same limitations as COTS products. Maintainability, security, and performance requirements may be specified only where the project has direct influence over the modified code or configuration.</li></ul>
<b>CUSTOM</b>	<ul style="list-style-type: none"><li>This Attendance Management System is a <b>fully custom-developed application</b>. Therefore, <b>all non-functional requirements specified in this document apply in full</b>. Performance, security, maintainability, compatibility, and operational requirements are directly addressed through system architecture, implementation choices, and development practices. These requirements are defined based on stakeholder needs and long-term university operational constraints.</li></ul>