# 1. Introduction

Overview of the purpose of the system analysis and design phase.

Importance of defining system requirements and architecture for successful implementation.

# 2. Problem Statement & Objectives

Problem Statement:

Manual website testing is time-consuming and prone to human error, leading to delays in deployment and potential quality issues.

Objectives:

To develop an automated testing tool that enhances testing efficiency, accuracy, and integration with development workflows.

# 3. Use Case Diagram & Descriptions

Use Case Diagram:

Include a diagram illustrating the interactions between users (QA Engineers, Developers, Product Owners) and the system.

Use Case Descriptions:

Use Case 1: Create Test Case

Actors: QA Engineer

Description: The QA Engineer creates a new test case in the system.

Use Case 2: Run Test Suite

Actors: QA Engineer

Description: The QA Engineer executes a set of automated tests.

Use Case 3: View Test Reports

Actors: Product Owner

Description: The Product Owner reviews the results of the executed tests.

# 4. Software Architecture

Architecture Overview:

High-level design outlining the main components of the system, including:

User Interface (UI)

Test Execution Engine

Reporting Module

Integration Layer (for CI/CD)

Architecture Style:

Describe the chosen architecture style (e.g., Microservices, MVC) and its benefits for the project.

# 5. Database Design & Data Modeling

Entity-Relationship Diagram (ERD):

Include an ERD showcasing the database structure, including entities such as Users, Test Cases, Test Results, and Reports.

Logical & Physical Schema:

Define tables, attributes, keys, and normalization considerations for the database.

# 6. Data Flow & System Behavior

Data Flow Diagram (DFD):

Context-level and detailed DFD showing how data moves through the system.

Sequence Diagrams:

Illustrate key interactions between components during test execution and reporting.

Activity Diagram:

Visualize the workflow of processes, such as creating and executing test cases.

State Diagram:

Represent different states of a test case (e.g., Created, Running, Completed, Failed).

Class Diagram:

Define the structure of the system by showing classes, attributes, methods, and relationships.

# 7. UI/UX Design & Prototyping

Wireframes & Mockups:

Include visual representations of the user interface for key functionalities (e.g., test case creation, report viewing).

UI/UX Guidelines:

Outline design principles, color schemes, typography, and accessibility considerations.

# 8. Conclusion

Summary of the system analysis and design process.

Emphasis on how the defined architecture and requirements will guide the development of the Website Testing Automation Tool.