3.5. Development Methodology

3.6. Software Quality Assurance Plans (include plans for testing Functional Requirements, Non-Functional Requirements, User-Acceptance Test, and System Stress Test)

3.6.1. Unit Test (Functional Requirements and Non-Functional Requirements)

In this section, we will delve into the unit testing phase of our project. Unit testing is a fundamental component of the software development process that focuses on verifying the individual units or components of the system in isolation. The purpose of this phase is to ensure that each unit of the system performs as expected and meets the defined requirements. In this section, we will outline the criteria and scenarios that will be used to conduct comprehensive unit testing for our system.

3.6.1.1. Unit Test Criteria

In this section, we will outline the unit test criteria for our system. These criteria will guide the development of effective unit tests to thoroughly evaluate the functionality and behavior of individual units within our system. The following unit test criteria will be employed:

Unit Test Criteria	Description
Isolation	Each unit test will isolate the specific unit under test to ensure that it can be evaluated independently of other components or dependencies.
Repeatability	Each unit test will be designed to produce consistent results when executed multiple times. This repeatability ensures that the behavior of the units remains stable and predictable over time.
Readability and Maintainability	We will prioritize writing unit tests that are easy to read, understand, and maintain. Following best practices for test organization, using meaningful test case names, providing clear documentation, and adhering to coding standards will enhance the readability and

	maintainability of our unit
Assertions	Each unit test will include assertions that validate the expected behavior and outcomes of the units being tested. These assertions will check if the actual results match the expected results, ensuring that the functional requirements are met.

3.6.1.2. Unit Test Scenario

In this section, we will define several unit test scenarios to validate the behavior and functionality of individual units within our system. These unit test scenarios aim to cover various aspects of the system and simulate different user interactions to ensure its correctness and robustness.

Scenario	Description	Test Steps
Natural Language Understanding	Input a user query and assess if the virtual service desk accurately understands the intent and extracts relevant information.	● Provide a user query containing specific keywords or phrases related to a service request. ● Verify that the virtual service desk correctly identifies the intent and extracts the necessary details from the query.
Response Generation	Evaluate the virtual service desk's ability to generate relevant and accurate responses based on user queries.	● Input different user queries covering various topics or service-related questions. ● Verify that the virtual service desk generates appropriate responses that address the user's query accurately and provide relevant information.

Request Identification and Classification	Test the system's capability to identify and classify user requests according to service operation management processes.	● Input different user requests representing various service categories or operations. ● Validate that the virtual service desk correctly identifies and classifies each request into the corresponding service operation management process.
Question Probing	Assess the system's capability to probe and clarify user questions when necessary	● Input user queries that require further clarification or additional information. ● Evaluate if the virtual service desk recognizes the need for clarification and responds with probing questions to gather the necessary details. ● Verify that the system's probing questions are clear, contextually relevant, and help the user provide the required information for accurate assistance.
Reporting and Analytics	Verify the system's ability to generate clear and concise reporting and analytics on user interactions, ticket resolution times, and other key performance indicators.	● Monitor and record user interactions and system responses. ● Assess the accuracy and completeness of the generated reports and analytics, ensuring they provide valuable insights for improving service operation processes.