**e\_Dialysis Clinical Systems**

**QA Test Plan**

**Version: <5.0>**

**Prepared By:**

**Vatsal Doshi** **002776613**

**Sri Sai Nikhil Enni** **002725894**

**Yuktaba Gohil** **002786959**

**Ananthakrishna Kalle**  **002772115**

**Niharika Santhoshini Karri** **002727629**

**For - Prof. Medi Servattalab**

**Software Quality Control and Management**

Table of Contents

[1 Document Acceptance and Sign-Off 3](#_Toc164865081)

[2 Revision History 3](#_Toc164865082)

[3 Introduction 4](#_Toc164865083)

[3.1 Purpose 4](#_Toc164865084)

[3.2 Project Overview 4](#_Toc164865085)

[4 Scope 6](#_Toc164865086)

[4.1 In-Scope 6](#_Toc164865087)

[4.2 Out-of-Scope 6](#_Toc164865088)

[5 Testing Strategy 7](#_Toc164865089)

[5.1 Test Objectives 7](#_Toc164865090)

[5.2 Risks & Assumptions 8](#_Toc164865091)

[5.3 Data Approach 8](#_Toc164865092)

[5.4 Types of Testing 9](#_Toc164865093)

[5.5 Unit Testing 10](#_Toc164865094)

[5.6 Functional Testing 11](#_Toc164865095)

[5.7 User Acceptance Testing 11](#_Toc164865096)

[5.8 Regression Testing 11](#_Toc164865097)

[5.9 Performance Testing 12](#_Toc164865098)

[6 Execution Strategy 13](#_Toc164865099)

[6.1 QA Entrance Criteria 13](#_Toc164865100)

[6.2 QA Exit criteria 13](#_Toc164865101)

[6.3 Defect Management 14](#_Toc164865102)

[7 Environment Requirements 16](#_Toc164865103)

[7.1 Test Environments 16](#_Toc164865104)

[8 Dependencies 18](#_Toc164865105)

9 Project Timelines…………………………………………………………………….……..20

# Document Acceptance and Sign-Off

By signing below, I acknowledge that I have read the entire contents of this document and accept the document in this form as reasonably fulfilling the goals described in the section titled Document Purpose. I further agree that this will constitute the document of record and cannot be changed without review and acknowledgement of the groups shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Group / Role** | **Approver Name** | **Approver Signature** | **Date Approved** |
| Group 3/QA | Vatsal Doshi | Vatsal Doshi | 4/24/2024 |
| Group 3/QA | Sri Sai Nikhil Enni | Sri Sai Nikhil Enni | 4/24/2024 |
| Group 3/QA | Yuktaba Gohil | Yuktaba Gohil | 4/24/2024 |
| Group 3/QA | Ananthakrishna Kalle | Ananthakrishna Kalle | 4/24/2024 |
| Group 3/QA | Niharika Santhoshini Karri | Niharika Santhoshini Karri | 4/24/2024 |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Document/Department Editor: | | | |
| **Date** | **Revision #** | **Editor** | **Description of Change** |
| 4/19/2024 | 1.0 | Vatsal Doshi | Initial Test Plan, Introduction |
| 4/20/2024 | 2.0 | Sri Sai Nikhil Enni | Scope, Project Timelines |
| 4/21/2024 | 3.0 | Ananthakrishna Kalle | Testing Strategy, Risks and Assumptions, Data Approach |
| 4/22/2024 | 4.0 | Niharika Santhoshini Karri | Execution Strategy, QA Entry and Exit Criteria, Defect Management |
| 4/23/2024 | 5.0 | Yuktaba Gohil | Environment Requirements, Dependencies |

# Introduction

## Purpose

***Test Strategy***

* The test strategy for the e\_Dialysis Clinical System focuses on ensuring the quality and reliability of the application. It includes a comprehensive approach to testing that covers functional, integration, and regression testing. The strategy also emphasizes the importance of early testing and continuous integration to identify and address issues as early as possible. The strategy is to use testing techniques to achieve thorough test coverage.

***Test Approach***

* The test approach involves a combination of testing techniques. Testing will be used for functional testing, where test cases will be executed to validate the behavior of the application. Testing will be used for regression testing, where test scripts will be developed to test the application after each change. This approach ensures that the application is thoroughly tested, and any issues are identified and addressed promptly.

***Execution Strategy***

* The execution strategy for testing involves setting up a dedicated test environment that mirrors the production environment as closely as possible. Test data will be carefully managed to ensure that it is representative of real-world scenarios. Test cases will be executed according to a predefined test plan, and any identified issues will be documented and tracked using a defect tracking system. The execution strategy also includes regular communication and reporting to stakeholders to keep them informed of the testing progress and any issues that arise.

***Test Management***

* Test management involves planning, organizing, and controlling the testing activities. This includes defining the scope of testing, developing test plans and test cases, and allocating resources for testing. It also involves managing the test environment, test data, and test execution. Test management also includes tracking and reporting on testing progress, including the number of test cases executed, the number of defects identified, and the overall quality of the application.

## Project Overview

The e\_Dialysis Clinical System is a new application designed to enhance the efficiency of nurses in providing patient care. This system streamlines the process of admitting and treating patients, enabling healthcare professionals to deliver better care.

The system offers a wide range of functionalities to support patient care management. It allows for the admission of both new and existing patients, with different processes for each. For new patients, their demographic information is sent to the e\_Finance System, e\_Storage System, and Pfizer System Lab, creating new admission folders in these systems. For existing patients, their existing folders are updated with the latest demographic information. Insurance information is also sent to the appropriate insurance companies for registration purposes.

Treatment options are available for patients at home or in a clinical setting, with specific modalities for each. Treatment and lab charges are stored in the system's database after each treatment session, and lab orders are sent to the Pfizer System Lab for processing. The lab results are then sent back to the e\_Dialysis Clinical System for storage. All treatment and lab charges are sent to the e\_Finance system for billing purposes.

The e\_Finance System plays a crucial role in processing claims for treatment. Claims are submitted to insurance companies, including Medicare/Medicaid, Tufts, or United Health, for payment. Payments received from insurance companies are directed to the Recon\_Trust company based on the claims received. Weekly reconciliations are conducted between the e\_Finance System and the Recon\_trust company for auditing purposes.

# Scope

## In-Scope

1. Admitting new and existing patients.
2. Sending patient demographics to e\_Finance, e\_Storage, and Pfizer System Lab.
3. Retrieving and updating patient folders in e\_Finance, e\_Storage, and Pfizer System Lab.
4. Validating the "Dialysis\_Flag."
5. Examining treatment location, modality, and drugs.
6. Verifying that the e\_Dialysis Clinical System sends treatment and lab charges to the e\_Finance System.
7. Checking if the end treatment is sent to the e\_Dialysis Clinical System.
8. Verifying that patient insurance is sent to the insurance company and Recon\_Trust Company.
9. Verifying that the Recon\_Trust Company sends payments to the e\_Finance System.

**Interfaces, Process, Workflow, Vendor Integration:**

* Interfaces: e\_Finance, e\_Storage, Pfizer System Lab, e\_Dialysis Clinical System, insurance companies, Recon\_Trust Company.
* Process: Admission, treatment, lab orders and results, charges collection, billing, reconciliation.
* Workflow: Patient admission, treatment decision, lab processes, charges collection, billing, payment reconciliation.
* Vendor Integration: Integration of systems like e\_Finance, e\_Storage, and Pfizer System Lab with the e\_Dialysis Clinical System for data exchange and processing.

## Out-of-Scope

* The Recon\_Trust Company failing to do a weekly reconciliation is out of scope as it falls outside the boundaries of the e\_Dialysis Clinical System's functionalities.
* The scenario of at-home patients needing In-Center treatment is also out of scope as it is not a functionality requirement of the system being tested.

# Testing Strategy

## Test Objectives

**Patient Admission Testing:**

* Test approval process for admitting new and existing patients based on the "FinAdmit\_Flag" and financial clearance status.
* Verify demographic information sent to relevant systems upon admission.
* Confirm creation of folders in the e\_Finance System as needed.

**Patient Treatment Testing:**

* Verify determination of patient treatment based on the "Dialysis Flag".
* Test functionality for patients to choose At-Home or In-Center treatment, with different modalities and drugs based on insurance provider.

**Treatment and Lab Charges Testing:**

* Test storage of patient treatments and lab results in the system's database.
* Verify charges are collected and sent to e\_Finance System for billing.
* Financial Billing and Reconciliation Testing:
* Test submission of claims for treatment and lab charges to insurance companies.
* Verify correct processing of payments received from insurance companies.
* Test weekly reconciliation process with Recon\_Trust Company for auditing.

**Interfaces Testing:**

* **Admission Interface:** Test interface with e\_Finance System, e\_Storage System, Pfizer System Lab, and Financial admit flag for managing patient admission and claims.
* **Insurance Interface:** Test interface between e\_Storage System and insurance companies for verifying patient insurance.
* **Treatment Interface:** Test interface for determining patient treatment based on Dialysis Flag, including location, drugs, and insurance coverage.
* **Payment Interface:** Test interface between e\_Dialysis Clinical System and e\_Finance System for processing charges.
* **Audit Interface:** Test interface between Recon\_Trust Company and e\_Finance System for reconciliation and auditing.

**Tasks and Responsibilities**

* The e\_Storage system determines patient admission based on the FinAdmit\_Flag.
* The FinAdmit\_Flag in the e\_Storage system determines if the patient will be treated by the specified insurance company.
* The Dialysis\_flag determines whether the patient will receive treatment.
* The e\_Finance system collects lab charges and submits claims to the insurance company.
* Treatment Location determines the treatment modality and drug usage for the patient.
* The Pfizer System Lab collects and sends lab orders to the e\_Dialysis Clinical System for storage.
* The Recon\_Trust company collaborates with the e\_Finance System and conducts weekly audits.

## Risks & Assumptions

|  |  |
| --- | --- |
| **Risks/Assumptions** | **Mitigation** |
| The testing environment may not accurately reflect the production environment, leading to differences in behaviour. | Validate the testing environment to ensure it accurately reflects the production environment. |
| Assumption that necessary resources like test servers, databases are available for testing activities. | Evaluate and select testing tools carefully, ensuring they meet the testing requirements and are compatible with the environment. |
| Insufficient or inaccurate test data may hinder the ability to simulate real-world scenarios effectively. | Manage test data carefully, ensuring it is accurate, relevant, and available for testing purpose |

## Data Approach

**Functional Testing:**

1. **Identify Test Scenarios:** Define functional test scenarios covering patient admission, treatment administration, billing processes, etc.
2. **Data Source:** Test data can be sourced from various sources including synthetic data generators, production-like datasets, or extracted from existing databases.
3. **Purpose:** The test data is used to simulate real-world scenarios and validate the functionality of the e\_Dialysis system under different conditions.

**User Acceptance Testing (UAT):**

1. **Collaborate with Stakeholders:** Gather input from end-users and business representatives to understand UAT data requirements.
2. **Data Source:** UAT test data may come from production-like datasets, anonymized customer data, or synthesized data representing typical user interactions.
3. **Purpose:** The test data mirrors real-world usage scenarios and is used to validate the system's compliance with user requirements and expectations during UAT.

## Types of Testing

|  |  |  |
| --- | --- | --- |
| **Test Type** | **Description** | **Responsible Parties** |
| Unit Testing | Verify individual units or components of the e\_Dialysis Clinical System in isolation to ensure they function correctly | Developer, QA Tester |
| Integration Testing | Validate the interaction between individual units or components of the e\_Dialysis Clinical System to ensure they integrate seamlessly. | Developer, QA Tester |
| System Testing | Evaluate the entire System to confirm it meets the specified requirements and functions as intended. | QA Tester |
| User Acceptance Testing | Assess the e\_Dialysis Clinical System from the user's perspective to ensure it meets user needs and requirements. | QA Tester, Business End Users, SME, Business Analyst |
| Regression Testing | Confirm that modifications to the e\_Dialysis Clinical System do not adversely affect existing functionalities. | QA Tester |
| Performance Testing | Measure the performance of the e\_Dialysis Clinical System under various conditions (e.g., load, stress) to verify it meets performance criteria. | QA Tester |
| Functional Testing | Validate the functionality of the e\_Dialysis Clinical System against defined requirements. | QA Tester |
| Stress Testing | Evaluate the e\_Dialysis Clinical System's ability to withstand extreme conditions (e.g., high load, high traffic) while remaining stable and responsive. | QA Tester |

## Unit Testing

Unit testing is the process of testing individual units or components of a software application in isolation to ensure they perform as expected. It involves verifying that each unit behaves correctly according to its design specifications, typically through automated tests written by developers. The goal is to identify and fix defects early in the development cycle, improving the overall quality and reliability of the software.

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Ananthakrishna Kalle | Quality Assurance | QA Tester |
| Niharika Santhoshini Karri | Development | Developer |
| Vatsal Doshi | Development | Developer |

## Functional Testing

Functional testing assesses a software application's features against requirements to ensure it functions as intended. It verifies user interfaces, APIs, and integrations, aiming to validate the software's functionality, usability, and performance, ensuring it meets user needs and expectations.

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Yuktaba Gohil | Quality Assurance | Test Manager |
| Sri Sai Nikhil Enni | Quality Assurance | Test Lead |
| Vatsal Doshi | Quality Assurance | Test Analyst |

## User Acceptance Testing

UAT, or User Acceptance Testing, involves testing the software's functionality from the end-user's perspective to ensure it meets their requirements and expectations. It's the final phase of testing before deployment, where users validate the system's usability, functionality, and compliance with business needs.

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Niharika Santhoshini Karri | Quality Assurance | QA Tester |
| Yuktaba Gohil | Product | Subject Matter Expert (SME) |
| Ananthakrishna Kalle | Business | Business Analyst |
| Sri Sai Nikhil Enni | Customer | Business End User |

## Regression Testing

Regression testing verifies that recent code changes haven't adversely affected existing functionality. It ensures that previously developed and tested software still performs as expected after modifications or enhancements.

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Sri Sai Nikhil Enni | Quality Assurance | Test Manager |
| Vatsal Doshi | Quality Assurance | Test Lead |
| Yuktaba Gohil | Quality Assurance | Test Analyst |

## Performance Testing

Performance testing evaluates software's responsiveness, stability, and scalability under varying conditions to ensure it meets performance requirements. It identifies potential bottlenecks and assesses system behavior, such as response time and resource utilization, to optimize performance and enhance user experience.

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Niharika Santhoshini Karri | Quality Assurance | Test Manager |
| Ananthakrishna Kalle | Quality Assurance | Test Lead |
| Yuktaba Gohil | Quality Assurance | Test Analyst |

# Execution Strategy

## QA Entrance Criteria

* QA entrance criteria ensures that the software is ready for testing and meets predefined standards before quality assurance activities begin.

|  |  |  |  |
| --- | --- | --- | --- |
| **QA Entrance Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *Test environment(s) is available* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | *Check if the test environment is ready.* |
| *Test data is available* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | *Ensure required test data is accessible.* |
| *The code has been merged successfully* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | *Confirm code integration is done.* |
| *Development has completed unit testing* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | *Verify unit tests are passed.* |
| *Test scripts are completed, reviewed, and approved by the Project Team* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | *Confirm code integration is done.* |

## QA Exit criteria

* QA exit criteria signify desirable conditions that must be met before concluding the QA process. If any of these exit criteria are not met, the QA team will assess the associated risks, identify mitigation actions, and provide a recommendation regarding the readiness of the system for implementation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *100% Test Scripts executed* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | Ensure all test scripts are executed. |
| *90% pass rate of Test Scripts* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | Aim for a high pass rate to ensure the system meets quality standards. |
| *No open Critical and High severity defects* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | Critical and high severity defects must be resolved before proceeding with implementation. |
| *All remaining defects are either canceled or documented as Change Requests for a future release* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | Document any remaining defects for future releases or address them appropriately. |
| *All expected and actual results are captured and documented with the test script* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | Ensure all test results are documented for future reference and analysis. |
| *All test metrics collected based on reports from daily and weekly status reports* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | *Collect and analyze test metrics to assess the overall quality of the system.* |
| *All defects logged in the defect tracker/spreadsheet* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | Ensure that all identified defects are logged in the defect tracker or spreadsheet. |
| *Test environment cleanup completed and a new backup of the environment* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png | The test environment should be cleaned up to remove any artifacts and backed up for future use. |

## Defect Management

* *Validate test cases by executing them against the software under test.*
* *Compare the expected results with the actual results to confirm the software behaves as intended.*
* *Log any inconsistencies as defects in the defect tracker.*
* *It is expected that the testers execute all the scripts in each of the cycles described above.*
* *The defects will be tracked through the Defect Tracker or Spreadsheet.*
* *It is the responsibility of the tester to open the defects, retest, and close the defect.*

**Possible defects that can arise :**

* **Functional Defects:** Problems related to the system's functionality, such as incorrect treatment decisions or faulty charge calculations.
* **Interface Defects:** Issues with integrating different modules or systems, leading to data transfer errors between systems like e\_Finance or Pfizer System Lab.
* **Data Defects:** Errors in stored or processed data, such as incorrect patient information or lab results.
* **Usability Defects:** Problems affecting the system's usability, like confusing interfaces or unclear error messages.
* **Performance Defects:** Issues with system performance, such as slow response times or failures under high loads.
* **Compatibility Defects:** Problems with system compatibility, including issues with different browsers, devices, or operating systems.
* **Security Defects:** Vulnerabilities that could be exploited to gain unauthorized access or compromise patient data.
* **Regression Defects:** Problems arise when a previously working feature stops functioning correctly after a system change or update.

Defects found during the Testing should be categorized as below:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| *1 (Critical)* | * *Functionality is blocked and no testing can proceed* * *Application/program/feature is unusable in the current state* |
| *2 (High)* | * *Functionality is not usable and there is no workaround but testing can proceed* |
| *3 (Medium)* | * *Functionality issues but there is a workaround for achieving the desired functionality* |
| *4 (Low)* | * *Unclear error message or cosmetic error which has minimum impact on product use.* |

# Environment Requirements

## Test Environments

**Test Environment Requirements:**

1. **Hardware Specifications:**
   * Hardware specifications should closely match those of the production environment, including CPU, RAM, and storage.
   * Diverse client machine configurations are necessary to ensure compatibility across various systems.
2. **Software Specifications:**
   * A range of operating system (OS) versions should be available to cover different user scenarios.
   * Compatibility testing with various web browsers and their versions is required.
   * Integration with all third-party software components utilized by the application should be tested.
3. **Network Environment:**
   * Network settings should replicate those of the production environment to simulate real user conditions.
   * VPN access should be provided for testers who need to conduct tests remotely.
4. **Access and Permissions:**
   * Testers should be provided with appropriate access rights to simulate different user roles.
   * Configuration management tools should be utilized to maintain consistency across testing environments.

**Security Requirements:**

1. **Data Protection:**
   * Sensitive data should be encrypted to ensure confidentiality.
   * Secure channels such as HTTPS, SSL, and TLS should be used for data transmission.
2. **Access Control:**
   * Role-based access controls should be implemented within the system.
   * Multi-factor authentication and strong password policies should be enforced for enhanced security.
3. **Network Security:**
   * Firewalls and intrusion detection systems should be employed to protect network integrity.
   * Regular security scanning and vulnerability assessments should be conducted to identify and mitigate risks.
4. **Regular Updates and Patches:**
   * All systems should be regularly updated with the latest software and security patches to address vulnerabilities.

**Additional Considerations:**

* **Test Server Setup:** Establish separate test servers as needed for testing requirements.
* **Network Setup:** Include configurations for internet, LAN, and WiFi settings.
* **Test PC Setup:** Prepare test PCs with necessary software and hardware configurations.
* **Test Data Creation:** Generate appropriate test data for various testing scenarios.
* **Bug Reporting Tools:** Ensure availability of bug reporting and tracking tools for the testing team.

**Test Environment Pathway:**

* Development -> QA (Quality Assurance) -> Staging -> Production

# Dependencies

**Dependencies and Their Comments:**

• **Financial Clearance Flag:** This flag must be set to ‘Yes’ to indicate successful financial clearance, allowing patients to be admitted into the system for treatment. Patients not financially cleared will be rejected.

• **Dialysis Flag:** This flag must be set to ‘Yes’ to ensure that new or existing patients will receive treatment through the e\_Dialysis Clinical System based on the patient's dialysis eligibility.

• **Integration with e\_Finance System, e\_Storage System, and Pfizer System Lab:** Successful integration with these systems is crucial for patient admission, updating of patient folders, and transmission of demographic information and lab orders.

• **Insurance Verification:** Verification of patient insurance is essential for admission approval. Patients with insurance types other than Medicare/Medicaid, Tufts, or United Health will be rejected.

**Test Types and Tool Requirements:**

* **UX Testing:** Employing Usability Hub or similar tools to evaluate user experience and interface design for the e\_Dialysis system.
* **Unit Testing:** Utilizing JUnit or similar frameworks for testing individual units or components of the e\_Dialysis software.
* **Automation Testing:** Implementing Selenium WebDriver for automated script execution and testing efficiency, especially for repetitive tasks.
* **Performance & Load Testing:** Using Apache JMeter to assess system performance under varying load conditions, ensuring scalability and reliability.
* **Database Testing:** Employing SQL Server Management Studio or similar tools for comprehensive database validation and verification, ensuring data integrity and consistency.
* **Back End Testing:** Utilizing Postman or similar tools for testing server-side APIs and functionalities of the e\_Dialysis system.
* **Reporting Tool:** Utilizing Microsoft Excel or similar tools for documenting test results and findings, facilitating clear communication and analysis.
* **Bug/Defect Tracking:** Employing Bugzilla or Jira for efficient tracking and management of identified bugs and defects throughout the testing process.

**Other Critical Dependencies:**

* **Availability of Consistent Test Data:** Reliable test data that accurately reflects real-world patient demographics, insurance information, and treatment scenarios is essential for thorough testing.
* **Availability of Testing Resources:** The necessary testing tools and resources required for reliable testing should be readily available and accessible to the testing team.
* **Early Defect Detection and Management:** Dependencies on prompt identification and resolution of testing-related issues, allowing for retesting and closure within project schedules, are crucial for maintaining project timelines and quality standards.

**9 Project Timelines**

|  |  |  |
| --- | --- | --- |
| **Task** | **Start Date** | **End Date** |
| Design and Functional Requirements Review | May 01, 2024 | June 2,2024 |
| User Story Reviews | June 12, 2024 | June19, 2024 |
| Test Plan Review | June 10, 2024 | June 10, 2024 |
| Test Cases Review, Before each Sprint | June 30, 2024 | June 30, 2024 |
| Sprint 1 to Sprint 5 (Two Week Sprints) | July 10, 2024 | August 28, 2024 |
| Performance & Load Testing | September 1, 2024 | September 15, 2024 |
| Pilot Release | October 1, 2024 | October 1, 2024 |
| Full Deployment | November 1, 2024 | November 1, 2024 |
| Project Retrospective | December 1, 2024 | December 1, 2024 |