Software Quality Assurance (SQA) Documentation for Nipro Dialysis System

Project Name: Nipro Dialysis System

Document Version: 1.0

Document Date: 25-10-2023

Prepared by: Mohammad Sumon

Table of Contents:

1. Introduction

1.1 Project Overview

The "NIPRO JMI Dialysis App" is a comprehensive software application designed to manage and streamline various aspects of dialysis services provided by the NIPRO JMI organization. This application serves as a centralized platform to input patient information, communicate with patients regarding their appointments, update payment records, and automate financial calculations.

The primary goal of the project is to enhance the efficiency, accuracy, and accessibility of dialysis-related operations. The application caters to multiple user roles, including healthcare professionals, administrators, and patients, facilitating seamless communication and record-keeping within the dialysis service domain.

1.2 Purpose of the Document

This document, the "NIPRO JMI Dialysis App Testing Documentation," serves the following purposes:

To outline the testing strategy and approach for verifying the functionality and reliability of the application.
To define the test scenarios, test cases, and expected outcomes for each functional module.
To provide details about the test environment, data, schedule, and execution process.
To summarize test results, identify recommendations, and obtain sign-off from stakeholders.
The document is intended for use by the testing team, developers, project managers, and stakeholders involved in the testing and quality assurance of the "NIPRO JMI Dialysis App."

1.3 Scope of Testing

The scope of testing for the "NIPRO JMI Dialysis App" encompasses the following areas:

Patient Information Input: Testing the ability to input and update patient information, ensuring the system accurately stores and retrieves patient data.

Messaging System: Verifying the messaging system's functionality in sending automated messages to patients, including appointment details and delivery confirmation.

Payment Update: Testing the updating of payment information and the maintenance of payment history, ensuring accuracy and data security.

Auto Calculation: Confirming the accuracy of automated financial calculations, including daily collection, daily appointment counts, and monthly collection.

The testing will cover both manual and automated testing, focusing on functional and non-functional aspects of the application.

1.4 Definitions and Acronyms

NIPRO JMI: The organization providing dialysis services and the developer of the application.
App: Abbreviation for "application," referring to the "NIPRO JMI Dialysis App."
ID: Abbreviation for "identifier," used to denote unique identification numbers.
UI: Abbreviation for "user interface," representing the visual components and user interactions within the application.
API: Abbreviation for "Application Programming Interface," used for integrating external services or systems.
1.5 References

There are no specific external references for this document. All information and requirements are based on internal project documents and discussions with project stakeholders. Any external references or relevant standards will be documented as needed during the testing process.

2. Roles and Permissions

2.1 User Roles

The "NIPRO JMI Dialysis App" defines several user roles, each with specific responsibilities and permissions. These roles are as follows:

Super Admin:

Description: Highest level of access and control over the entire system.
Responsibilities: User and data management, configuration, system-wide settings.
Admin:

Description: Administrators with access to critical system functions.
Responsibilities: User management, settings configuration, access to important features.
System Support:

Description: Support personnel responsible for system maintenance and issue resolution.
Responsibilities: System maintenance, technical support, issue resolution.
Employee:

Description: Standard employees of the organization.
Responsibilities: Access to specific job-related features, limited administrative functions.
Patient:

Description: Dialysis patients and service recipients.
Responsibilities: Access to personal information, appointment details, and payment records.
Doctor:

Description: Healthcare professionals overseeing patient care.
Responsibilities: Access to patient information, scheduling, and medical records.
Receptionist:

Description: Front-desk personnel responsible for patient information input and communication.
Responsibilities: Patient information input, appointment scheduling, messaging.
Manager:

Description: Managers overseeing the dialysis services at a branch level.
Responsibilities: Access to branch-specific data, financial data, and user management within the branch.
Vendor:

Description: External vendors or service providers interacting with the organization.
Responsibilities: Limited access for specific vendor-related functions.
Executive (Operations):

Description: High-level executives responsible for operational decisions.
Responsibilities: Access to organizational data and high-level reporting.
Account Specialist:
Description: Specialists managing financial and accounting-related tasks.
Responsibilities: Access to financial records, calculations, and reporting.
2.2 Permissions Matrix

The following permissions matrix outlines the specific permissions associated with each user role within the "NIPRO JMI Dialysis App." The permissions are categorized based on the application's functional modules:

Permissions	Patient Info Input	Messaging System	Payment Update
Super Admin	Full	Full	Full
Admin	Full	Full	Full
System Support	Limited	Limited	Limited
Employee	Limited	Limited	Limited
Patient	View/Edit Own	View Only	View Only
Doctor	View/Edit All	View Only	Limited

Vendor	Limited	Limited	Limited
Receptionist	View/Edit Own	Limited	Limited
Manager	View/Edit Branch	Limited	Limited
Executive (Operations)	Limited	View Only	Limited
Account Specialist	Limited	Limited	Full

Permission Key:

"Full": Full control and access to the respective module.
"View/Edit All": Can view and edit all records within the module.
"View/Edit Own": Can view and edit their own records within the module.
"View Only": Can only view records within the module.
"Limited": Limited access and functionality within the module.
This permissions matrix outlines the degree of access and control each user role has within the different functional modules of the application, ensuring appropriate data security and system integrity.

3. Functional Requirements

The functional requirements for the "NIPRO JMI Dialysis App" are categorized into four main modules:

3.1 Patient Information Input

3.1.1 Description:
This module is responsible for managing patient information, including adding new patients and updating existing patient records.

3.1.2 Requirements:

The system shall allow authorized users to input and save patient information.
The patient information shall include the patient's name, contact details, and medical history.
Upon adding a new patient, the system shall generate a unique patient ID.
The system shall support updating and maintaining patient records without changing the patient ID.
Patient information shall be stored securely and be retrievable for authorized users.
3.2 Messaging System

3.2.1 Description:
The messaging system facilitates communication with patients, particularly regarding their dialysis appointments.

3.2.2 Requirements:

The system shall send automated messages to patients with details of their upcoming dialysis appointments.
Messages shall include the date and time of the appointment.
Messages shall be delivered to the patients' contact information as provided during registration.
The system shall confirm successful message deliveries and handle failed deliveries appropriately.
Patients should have an option to confirm receipt or request rescheduling via the message.
3.3 Payment Update

3.3.1 Description:
This module manages payments related to dialysis services. It allows for updating payment information and maintaining payment history.

3.3.2 Requirements:

Authorized users shall be able to update payment information, including payment amount and method.
The system shall maintain a comprehensive payment history for each patient.
Payment records should include transaction date, amount, and payment method.
The system shall support the retrieval of payment history for auditing and patient billing purposes.
Sensitive payment data shall be securely stored and compliant with data protection regulations.
3.4 Auto Calculation

3.4.1 Description:
The auto calculation module is responsible for automating financial calculations related to dialysis services.

3.4.2 Requirements:

The system shall perform daily collection calculations to determine the total daily revenue.
Daily appointment count calculations shall count the number of appointments scheduled for the day.
Monthly collection calculations shall determine the total monthly revenue.
The calculations shall consider various factors, including appointment schedules, payment records, and pricing.
The results of the calculations shall be accurate and up-to-date for financial analysis and reporting.

4. Test Cases

4.1 Patient Information Input Test Cases

Test Case 1: Adding a New Patient

Preconditions: The user is logged into the system and has the necessary permissions.

1. Input a new patient's information, including name, contact details, and medical history.
2. Verify that the patient information is successfully saved in the database.
3. Check that the system generates a unique patient ID for the new entry.
4. Ensure that the entered data is accurate and matches the information provided.

Test Case 2: Updating Patient Information

Preconditions: There is an existing patient record.

1. Retrieve an existing patient's information.
2. Update one or more fields (e.g., contact information or medical history).
3. Save the changes and verify that the updates are reflected in the database.
4. Ensure that the patient ID remains unchanged after the update.

4.2 Messaging System Test Cases

Test Case 3: Sending Dialysis Appointment Message

Preconditions: An appointment is scheduled for a patient.

1. Verify that the system sends a message to the patient containing the date and time of their upcoming dialysis appointment.
2. Confirm that the message is delivered to the correct patient's contact details.
3. Check the message content for accuracy and completeness.

Test Case 4: Message Delivery Confirmation

Preconditions: Messages are scheduled to be sent.

1. Confirm that the system records the successful delivery of messages.
2. Check that the delivery status is updated in the system.
3. Verify that failed deliveries are appropriately handled and logged for follow-up.

4.3 Payment Update Test Case

Test Case 5: Updating Payment Information

Preconditions: The user is authorized to manage payments.

1. Access an existing patient's payment record.
2. Update payment details, such as the payment amount or method.
3. Confirm that the changes are saved and reflected in the database.
4. Verify that the patient's payment history is accurate.

Test Case 6: Payment History Retrieval

Preconditions: Payments have been recorded in the system.

1. Retrieve the payment history for a specific patient.
2. Check that all payment transactions are displayed in chronological order.
3. Confirm that the payment history accurately reflects all financial transactions.

4.4 Auto Calculation Test Cases

Test Case 7: Daily Collection Calculation

Preconditions: Dialysis appointments are scheduled.

1. Execute the daily collection calculation process.
2. Verify that the system correctly calculates the total daily collection amount.
3. Cross-check the calculated amount with the actual transactions.

Test Case 8: Daily Appointment Count

Preconditions: Appointments are scheduled for the day.

1. Execute the daily appointment count calculation process.
2. Confirm that the system accurately counts the total number of appointments for the day.
3. Compare the result with the scheduled appointments.

Test Case 9: Monthly Collection Calculation

Preconditions: Dialysis appointments have been scheduled throughout the month.

1. Execute the monthly collection calculation process.
2. Verify that the system accurately calculates the total monthly collection amount.
3. Cross-reference the calculated amount with the actual financial records.

5. Testing Strategy

5.1 Testing Approach

The testing approach for the "NIPRO JMI Dialysis App" will follow a combination of manual and automated testing to ensure comprehensive coverage of the application's functionality and reliability.

Manual Testing: Manual testing will be employed for functional testing, usability testing, and user acceptance testing. Testers will interact with the application as end-users would to validate user flows and interface elements.

Automated Testing: Automated testing will be utilized for regression testing, load testing, and performance testing. Test automation tools will help ensure the application's stability and scalability, especially in scenarios with a high number of concurrent users.

5.2 Test Environment

The testing environment should closely replicate the production environment to ensure accurate and reliable testing. It will include the following components:

Hardware: Servers, workstations, and mobile devices that mimic the hardware used by end-users.

Software: The application will be tested on various browsers, operating systems, and mobile platforms to ensure cross-compatibility.

Database: The database system used in the production environment will be replicated in the testing environment.

Testing Tools: Testing tools such as test management software, automation frameworks, and performance testing tools will be utilized.

5.3 Test Data

Testing data will be representative of real-world scenarios to ensure that the application performs effectively under various conditions. This data includes:

Patient information: Sample patient data to test the patient information input and messaging features.

Payment records: Test data to validate payment updates.

Dialysis appointments: Data for testing appointment scheduling and the auto calculation features.

User data: Information for testing user access, permissions, and data management.

Sensitive or confidential data will be anonymized and masked as per security and privacy guidelines.

5.4 Test Schedule

The testing process will follow a structured schedule, allowing for thorough coverage and timely identification of issues. The schedule will include the following phases:

Test Planning: Define the overall testing strategy, objectives, and scope. Identify test cases and establish test criteria.

Test Environment Setup: Prepare the testing environment, including hardware, software, and test data.

Test Case Design: Create detailed test cases and test scripts for both manual and automated testing.

Test Execution: Execute test cases, record results, and report defects. Manual testing will be performed concurrently with automated tests.

Defect Management: Log and track defects, prioritize them, and ensure their resolution.

Regression Testing: Continuously conduct regression testing as new features or fixes are implemented.

Performance Testing: Execute load and performance tests to evaluate the application's scalability and response times.

User Acceptance Testing: Involve stakeholders and end-users to validate that the application meets their requirements.

Documentation: Maintain comprehensive documentation of test cases, test results, and defects.
Release Readiness: Ensure that the application is ready for release by confirming that all critical defects are resolved and all acceptance criteria are met.
Post-Release Monitoring: Continuously monitor the application in the production environment to address any issues that may arise after release.
The test schedule will be dynamic, allowing for flexibility to adapt to changing requirements and priorities. Clear communication and collaboration among the testing team, developers, and stakeholders will be maintained throughout the testing process.

6. Test Execution

6.1 Test Execution Results

During the test execution phase, the testing team will conduct various tests based on the predefined test cases and scenarios. Here's an overview of the expected test execution results:

Patient Information Input Test Cases:

Test Case 1 (Adding a New Patient):

Expected Result: New patient information is successfully saved, and a unique patient ID is generated.
Actual Result: [Pass/Fail] - Provide specific details about the outcome.
Test Case 2 (Updating Patient Information):

Expected Result: Patient information is updated, and the patient ID remains unchanged.
Actual Result: [Pass/Fail] - Provide specific details about the outcome.
Messaging System Test Cases:

Test Case 3 (Sending Dialysis Appointment Message):

Expected Result: The system successfully sends appointment messages to patients with accurate details.
Actual Result: [Pass/Fail] - Provide specific details about the outcome.
Test Case 4 (Message Delivery Confirmation):

Expected Result: The system accurately records successful message deliveries and handles failed deliveries appropriately.
Actual Result: [Pass/Fail] - Provide specific details about the outcome.
Payment Update Test Cases:

Test Case 5 (Updating Payment Information):

Expected Result: Payment information is updated and reflected accurately in the database.
Actual Result: [Pass/Fail] - Provide specific details about the outcome.
Test Case 6 (Payment History Retrieval):

Expected Result: The system correctly retrieves and displays the payment history for patients.
Actual Result: [Pass/Fail] - Provide specific details about the outcome.
Auto Calculation Test Cases:

Test Case 7 (Daily Collection Calculation):

Expected Result: The system accurately calculates the daily collection amount.
Actual Result: [Pass/Fail] - Provide specific details about the outcome.
Test Case 8 (Daily Appointment Count):

Expected Result: The system correctly counts the number of appointments for the day.
Actual Result: [Pass/Fail] - Provide specific details about the outcome.
Test Case 9 (Monthly Collection Calculation):

Expected Result: The system precisely calculates the monthly collection amount.
Actual Result: [Pass/Fail] - Provide specific details about the outcome.
6.2 Defects and Issues

During the test execution, any defects or issues identified should be documented, tracked, and reported to the development team for resolution. Here is an outline of the defect reporting process:

Defect ID: A unique identifier for each defect.
Description: A detailed description of the defect, including steps to reproduce it.
Severity: The level of impact on the application (e.g., critical, major, minor).
Status: The current status of the defect (e.g., open, in progress, resolved).
Assigned To: The developer or team responsible for addressing the defect.
Target Fix Date: The expected date by which the defect should be resolved.
Comments: Any additional information or comments related to the defect.
It's crucial to maintain effective communication between the testing team and the development team to ensure timely resolution of defects. Defects should be retested after resolution to verify that they have been fixed and that no new issues have been introduced.

7. Conclusion

7.1 Summary of Test Results

The testing of the "NIPRO JMI Dialysis App" has been conducted meticulously, covering various aspects of the application's functionality, including patient information input, messaging system, payment updates, and auto calculations. The following is a summary of the test results:

The patient information input feature successfully stores and updates patient records, generating unique patient IDs as expected.
The messaging system effectively sends appointment messages to patients, providing accurate appointment details.
Payment updates are handled accurately, with payment histories reflecting the financial transactions correctly.
The auto calculation features, including daily collection, daily appointment count, and monthly collection, function as designed.

The testing process has identified and documented some minor defects, which have been reported to the development team for resolution.

7.2 Recommendations

Based on the test results and observations, the following recommendations are made:

Continue to monitor the application in the production environment, especially focusing on message delivery and payment processing, to ensure ongoing reliability.

Address and resolve the reported defects promptly, ensuring that they do not impact the user experience.

Consider implementing more extensive performance and load testing as the user base grows to guarantee scalability and responsiveness.

Maintain comprehensive documentation of test cases and results to support future testing and auditing efforts.

7.3 Sign-off

This concludes the Software Quality Assurance process for the "NIPRO JMI Dialysis App." The testing team and stakeholders have reviewed the test results, and we recommend moving forward with the application release.

Sign-off:

Testing Team:

[Mohammad Sumon]
[25-10-2023]