Test Plan

Orthanc PACS system

3/10/2022
Praveen Das
QA Engineer

Document History

Version	Date	Author	Description of Change
1	10/03/2022	Praveen Das	first version created

Approvers List

Name	Role	Approver / Reviewer	Approval / Review Date
Praveen Das	**QA Engineer**	Robert	

Reference Documents

Version	Date	Document Name
1.	10/03/2022	Software Requirement Specifications V1.0

Table of Contents

1. INTRODUCTION

1.1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the Orthanc PACS system. The document introduces:

- Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).
- Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
- Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster)

1.2. Project Overview

Radiologists have a 26.1% error rate of clinically relevant findings and 38% of these errors can be prevented through the collaboration of radiologists and AI. It is unacceptable that existing technology is not adopted in radiology. It is time to change that!

World-class ML models and their integration into doctors offices are worthless without a high reliability achieved through rigorous high-quality testing. It is your mission to bring the benefits of AI into medical settings.

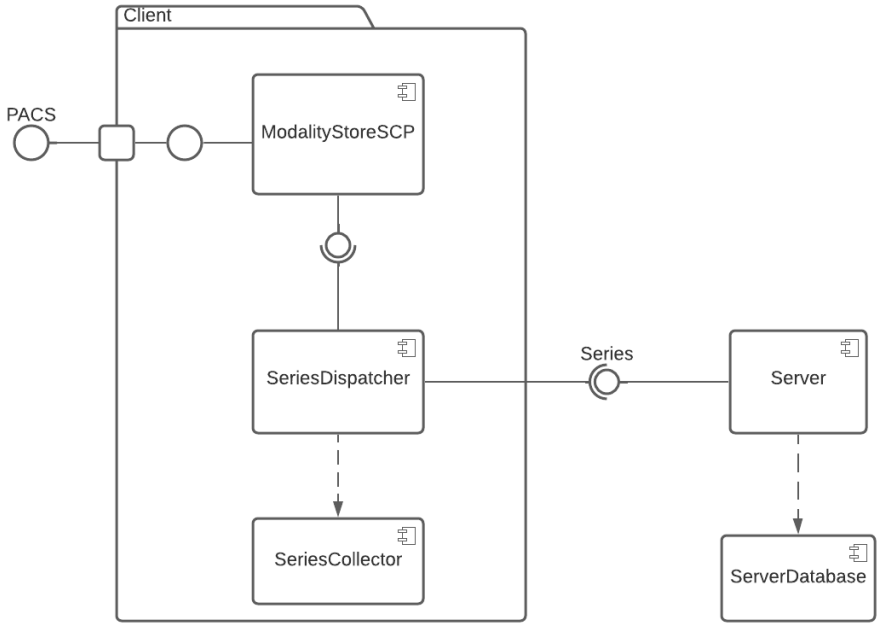

Fig 1: component diagram

When a patient enters the radiology office, he signs up at the front desk. Here, he receives legal forms and his visit is registered in the **radiological information system (RIS)**, which is the core management system of a radiology office.

Radiological images of the patient are then taken in different types of **modalities**, usually **magnetic resonance imaging machines (MRI)** or **computed tomography devices (CT)**. The images generated by these machines are stored in a **picture archiving and communication system (PACS)**. During the image capturing procedure, a **radiographer** observes the process using their **modality workstation**, a computer controlling the modality. Radiographers are responsible for verifying correct image capturing before sending them to the PACS. Once the images are stored in the PACS, the radiologist can create a diagnosis for the patient by viewing and evaluating the images.

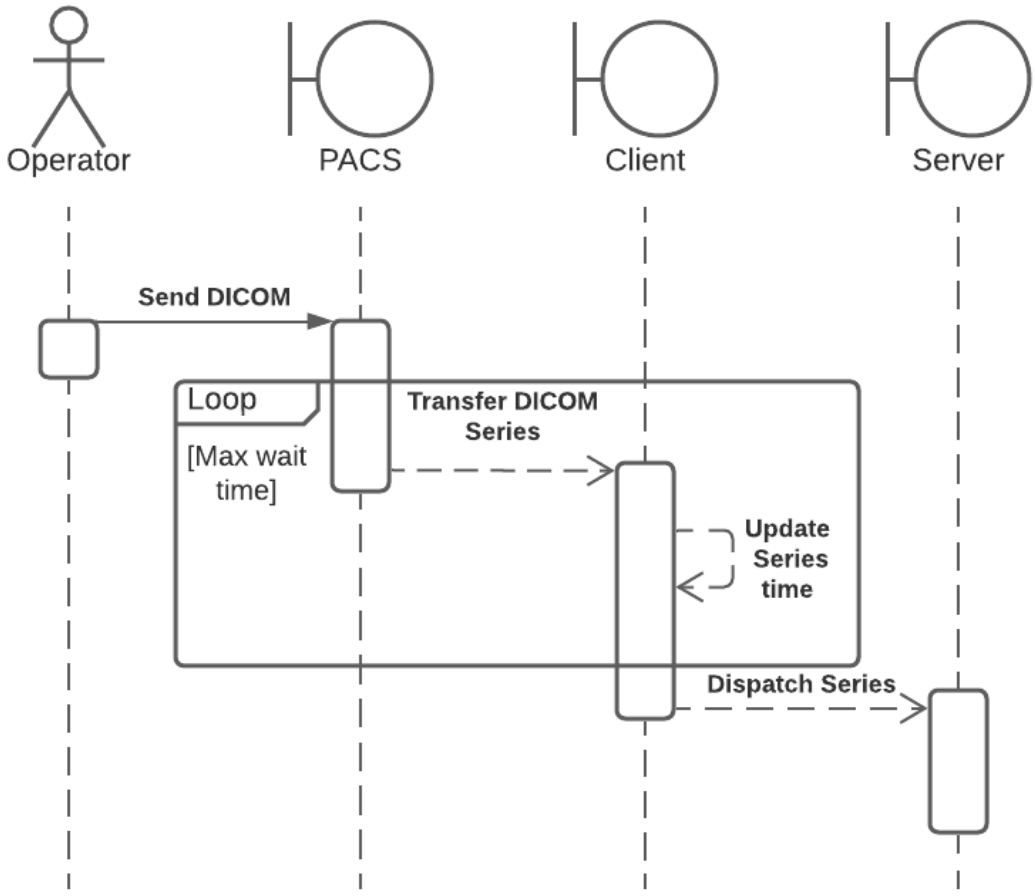

Fig 2: sequence diagram

The communication between the software systems (RIS, PACS, and the modality software) is standardized by the **DICOM (Digital Imaging and Communications in Medicine) standard**. DICOM uses a hierarchical data model which is oriented on the real world shown in *Fig. 2*. The top level of data organization is formed by the **Patient**. Each Patient subsequently has one or more **Study**, where one Study is usually generated during one visit. A Study consists of one or more **Series**, and a Series is a “stack” of single images, the so-called **Instances**. For more information on DICOM take a look around the DICOM is Easy blog.

1.3. Audience

- Project team members perform tasks specified in this document, and provide input and recommendations on this document.
- Project Manager Plans for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results.
- The stakeholders' representatives and participants (individuals as identified by the PMO Leads) may take part in the UAT test to ensure the business is aligned with the results of the test.
- Technical Team ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.
- Business analysts will provide their inputs on functional changes.

2. TEST STRATEGY

2.1. Test Objectives

The objective of the test is to verify that the functionality of Orthanc PACS system works according to the specifications.

The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via CR.

The final product of the test is twofold:

- A production-ready software;
- A set of stable test scripts that can be reused for Functional and UAT test execution.

2.2. Test Assumptions

Key Assumptions

- Production like data required and be available in the system prior to start of Functional Testing

General

- Exploratory Testing would be carried out once the build is ready for testing
- Performance testing is not considered for this estimation.

Functional Testing

- During Functional testing, testing team will use preloaded data which is available on the system at the time of execution
- The Test Team will be perform Functional testing only on Orthanc PACS system

2.3. Test Principles

- Testing will be focused on meeting the business objectives, cost efficiency, and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
- Testing environment and data will emulate a production environment as much as possible.
- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.
- There will be entrance and exit criteria.

2.4. Data Approach

- In functional testing, Orthanc PACS system will be loaded with the provided sample data sets.

2.5. Scope and Levels of Testing

2.5.1. Exploratory

PURPOSE: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.

TESTERS: Testing team.

METHOD: this exploratory testing is carried out in the application without any test scripts and documentation and is carried out manually.

TIMING: at the beginning of each cycle.

2.5.2. Functional Test

PURPOSE: Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

TESTERS: Testing Team.

METHOD: The test will be performed according to software requirement specifications.

TIMING: after Exploratory test is completed.

TEST ACCEPTANCE CRITERIA

1. All the functional requirements mention in the SRS are fulfilled.
2. Test cases approved and signed-off prior to start of Test execution
3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
4. Test environment with application installed, configured and ready to use state

User	Sub Levels	Complexity	No. of Test cases	Negative Test Cases	Expecting Additional Test Cases
Radiographer	capture MRI image	Medium	1	1	
Doctors	View the MRI/CT image	medium	10	1	1
	view diagnostic report from radiologist	Complex	10	1	
Radiologist	View the MRI/CT image	Complex	5	1	1
	update patient's diagnosis	Complex	5	1	
		Medium	3		
Receptionist	create patient account	Medium	1		
	create appointment	medium	1	1	3

Milestone list

MIlestone list is tentative and may change during project implementation depending on different scenarios.

Functional	Start Date	End Date	Available time
Test Execution	Monday, March 15, 2022	Tuesday, June 25, 2022	(3 months)
test of client side software	Monday, March 15, 2022	Friday, March 30, 2022	
test of server side software	Monday, April 3, 2022	Monday, April 17, 2022	
data protection tests	Tuesday, April 18, 2022	Tuesday, May 18, 2022	
date load testing	Wednesday, May 19, 2022	Friday, June 25, 2022	

2.5.3. Unit Test

PURPOSE: this test focuses on validating the function of each module. It aims to make sure that each module functions as expected.

TESTERS: the unit test is normally performed by developers while they are developing the software.

METHOD: The developers create unit tests as and when they are developing the software.

TIMING: Unit testing is done during software development.

TEST DELIVERABLES

S.No.	Deliverable Name	Author	Reviewer
1.	Unit Test Cases	Developer	QA Engineer

2.6. Test Effort Estimate

The total test activity takes about 100 man hours to complete all scenarios.

3. EXECUTION STRATEGY

3.1. Entry and Exit Criteria

Following are some examples of entry and exit criterias for the software

Exit Criteria
100% Test Scripts executed
95% pass rate of Test Scripts
No open Critical and High severity defects
95% of Medium severity defects have been closed
All remaining defects are either canceled or documented as Change Requests for a future release
All expected and actual results are captured and documented with the test script

3.2. Validation and Defect Management

- It is expected that the testers and developers execute all the scripts in their respective cycles. However it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts.
- The defects will be tracked through Jira. The technical team will gather information on a daily basis from Jira, and request additional details from the Defect reporter. The technical team will work on fixes.
- It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Defect Manager to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to review HP ALM on a daily basis, ask for details if necessary, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

Defects found during the Testing will be categorized according to the bug-reporting tool Jira and the categories are:

Severity	Impact
1 (Critical)	▪ This bug is critical enough to crash the system, cause file corruption, or cause potential data loss ▪ It causes an abnormal return to the operating system (crash or a system failure message appears). ▪ It causes the application to hang and requires rebooting the system.
2 (High)	▪ It causes a lack of vital program functionality with workaround.
3 (Medium)	▪ This Bug will degrade the quality of the System. However there is an intelligent workaround for achieving the desired functionality - for example through another screen. ▪ This bug prevents other areas of the product from being tested. However other areas can be independently tested.
4 (Low)	▪ There is an insufficient or unclear error message, which has minimum impact on product use.
5(Cosmetic)	▪ This type of issue is created just for changes which improves the UI and UX of the software.

3.3. Test Metrics

Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. The below are some of the metrics

Report	Description	Frequency
Test preparation & Execution Status	To report on % complete, %WIP, % Pass, % Fail Defects severity wise Status – Open, closed, any other Status	Daily
Daily regression test	To report on Pass, Fail, Total defects, highlight Showstopper/ Critical defects for running all functional tests in main branch	Daily
Project Weekly Status report	Project driven reporting (As requested by PM)	Weekly

4. TEST MANAGEMENT PROCESS

4.1. Test Design Process

- The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
- Each Test case will be mapped to Use cases to Requirements as part of Traceability matrix.
- Each of the Test cases will undergo review by the BUSINESS ANALYST and the review defects are captured and shared to the Test team. The testers will rework on the review defects and finally obtain approval and sign-off.
- During the preparation phase, tester will use the prototype, use case and functional specification to write step by step test cases.
- Testers will maintain a clarification Tracker sheet and the same will be shared periodically with the Requirements team and accordingly the test case will be updated. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements.
- For unit tests, they are created by developers during software development.

4.2. Test Execution Process

- Once all Test cases are approved and the test environment is ready for testing, tester will start a exploratory test of the application to ensure the application is stable for testing.
- If any showstopper during exploratory testing will be escalated to the respective development team for fixes

- Each tester performs step by step execution and updates the executions status. The tester enters Pass or Fail Status for each of the step directly in test management tool.
- Tester will prepare a Run chart with day-wise execution details
- If any failures, defect will be raised as per severity guidelines in Jira tool detailing steps to simulate along with screenshots and logs if appropriate.
- Daily Test execution status as well as Defect status will be reported to all stakeholders.
- Testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.
- This process is repeated until all test cases are executed fully with Pass/Fail status.

As per Process, final sign-off or project completion process will be followed

4.3. Test Risks and Mitigation Factors

Risk	Prob.	Impact	Mitigation Plan
SCHEDULE Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the scheduled start date.	High	High	• The testing team can control the preparation tasks (in advance) and the early communication with involved parties. • Some buffer will be added to the schedule for contingencies, although not as much as best practices advise.
RESOURCES Not enough resources, resources on boarding too late (process takes around 15 days.	Medium	High	Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing.
DEFECTS Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve.	Medium	High	Defect management plan is in place to ensure prompt communication and fixing of issues.
SCOPE Scope completely defined	Medium	Medium	Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing.
Non-availability of Independent Test environment and accessibility	Medium	High	Due to non availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution.
Delayed Testing Due To new Issues	Medium	High	During testing, there is a good chance that some “new” defects may be identified and

Risk	Prob.	Impact	Mitigation Plan
			may become an issue that will take time to resolve. There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved. If these issues become showstoppers, it will greatly impact the overall project schedule. If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution.

TEST ENVIRONMENT

- Docker image of the Orthanc PACS system
- Linux computer to run the docker image

APPROVALS

The Names and titles of all persons who must approve this plan.

Signature:	
Name:	**Praveen Das**
Role:	**QA Engineer**
Date:	

Signature:	
Name:	**Robert**
Role:	**CTO**
Date:	