

Laboratory Management

Software Requirement

Specification Version: 2.0

Approval Page

Prepared by:
Business Analyst

Signature: _____
Date: ____/____/____

Reviewed by:
Business Analyst

Signature: _____
Date: ____/____/____

Supported by: Signature:

Date: ____/____/____

Approved by: Signature:

Date: ____/____/____

Prepared By 0.7 1/40 Last modified on 10/09/2025 23:38:00
Laboratory Management

Revision History

Date	Version	Author	Change Description
04/03/2025	1.0	HauNK	Create new
19/06/2025	1.1	HauNK	Update privileges matrix
30/06/2025	1.2	HauNK	Rename the document, update Cleaner, add description for parameters list.
16/07/2025	2.0	HauNK	Update HL7 sync-up test results, health check. Update Warehouse service and Instrument Service.

Prepared By 0.7 2/40 Last modified on 10/09/2025 23:38:00
Laboratory Management

TABLE OF CONTENTS

1. Introduction.....5 1.1 Purpose

.....	5	1.2 Scope of
Project	5	
1.2.1 For Product Owners & Stakeholders:.....	5	
1.2.2 For Lab Users (Consultants):.....	5	
1.2.3 For the Development Team:.....	5	
2. Overall Description.....	5	2.1 Hematology
Analyzer	5	2.2 Actor
.....	6	2.3 Use
case diagram of the system	7	2.4
Configurations List.....	11	2.5
Reagents List.....	11	2.6
Parameters List	12	2.7
Privileges List	13	2.8
Event Table.....	14	
3. Software Requirements Specification.....	14	3.1 Identity and
Access Management Service.....	15	3.1.1 Users
Management.....	15	3.1.2 Roles
Management	17	3.1.3
Authentication/Authorize	18	3.2
Monitoring Service.....	19	3.2.1
Event Logs Management	19	3.2.2
Test Results Backup.....	20	3.2.3
Health Check	21	3.3
Warehouse Service.....	21	3.3.1
Instrument Management	21	3.3.2
Reagents History Tracing.....	23	3.3.3
Configurations Management.....	24	3.4
Patient Service	25	
Prepared By 0.7 3/40 Last modified on 10/09/2025 23:38:00		
Laboratory Management		
3.4.1 Patient Medical Record Management.....	25	3.5
Test Order Service	28	3.5.1

Test Order Management	28 3.5.2
Patient Test Order Results Management.....	29 3.5.3
Comment Management.....	31 3.5.4
Report.....	32 3.6
Instrument Service.....	33 3.6.1
Instrument Test Flow.....	33 3.6.2
Reagents Management.....	35 3.6.3
Configuration Management.....	36
4. Other Requirements	37
5. Integration	37
6. Non-functional Requirement	37
6.1 Reliability	37
6.2 Scalability.....	37
6.3 Supportability.....	38
6.4 Availability.....	38
6.5 Performance.....	38
6.6 Security & Privacy	38
6.7 Compatibility	38
6.8 Maintainability.....	38

Prepared By 0.7 4/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for the **Laboratory Management**, a platform designed to improve the efficiency, accuracy, and visibility of blood testing operations in clinical laboratories.

This document serves the following audiences:

- Product Owner (Business Representative): Understand how the system supports business goals such as reducing patient wait times, improving lab throughput, and maximizing the utilization of high-value hematology analyzers.
- Stakeholders/Sponsors: Gain clarity on the system's scope, expected outcomes, and alignment with strategic investments in laboratory automation and diagnostic quality.
- Consultants (Doctors, Lab Technicians): Ensure the system supports real-world workflows, including test order handling, instrument readiness, and reagent tracking, while integrating with commonly used hematology analyzers.
- Software Development Team: Use this document as a foundation for designing, developing, and testing the system according to clearly defined requirements and integration standards.

1.2 Scope of Project

The **Laboratory Management** is a centralized software solution that supports the end-to-end workflow of blood testing in clinical laboratories. It is designed to:

1.2.1 For Product Owners & Stakeholders:

- **Optimize lab operations** by reducing manual coordination and improving visibility into instrument status.
- **Protect capital investment** by ensuring optimal usage and maintenance of high-value hematology analyzers such as:
 - **Sysmex XN-Series**
 - **Beckman Coulter DxH Series**
 - **Abbott CELL-DYN Series**
 - **Mindray BC-Series**
 - **Erba H-Series**
- **Enhance patient experience** by minimizing delays and ensuring timely test processing.
- **Support compliance and traceability** through audit logs and reagent usage tracking.

1.2.2 For Lab Users (Consultants):

- Test Order Management: **Create, track, and manage blood test orders when patients arrive.**
- Instrument Availability Monitoring: **View real-time status of hematology analyzers to estimate wait times and plan workloads.**
- Instrument Readiness Verification: **Check if instruments are properly configured, have sufficient reagents, and are not in error or maintenance states.**

1.2.3 For the Development Team:

- The system will be deployed within the lab's internal network.
- It will integrate with existing Laboratory Information Systems (LIS) and support communication protocols such as HL7, ASTM, and TCP/IP.
- It must support user roles, audit logging, and real-time data synchronization with supported analyzers.

2. Overall Description

2.1 Hematology Analyzer

Link: [SMT30 | SMT30 3-Part Auto Hematology Analyzer | #SMT30 Operation Guide](#)

Prepared By 0.7 5/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

2.2 Actor

The table below describes all the actors involved in Laboratory Management. Each actor has a distinct role within the system, with access controlled by privileges. Additionally, actors can be **assigned custom roles**, allowing **flexibility** in defining **permissions** based on operational needs. The system includes **Administrator, Service, Lab Manager, and Lab Users**, each with specific access rights to ensure secure and efficient functionality.

Actor	Description
Adminstrator	Who have right to access all features in system.
Laboratory Manager	Who manages thelab, lab users, service users, have right to view and monitor the system.
Service	Who are individuals authorized to interact with a system for operational and maintenance purposes. Their main responsibilities involve monitoring, managing, and maintaining the system to ensure optimal performance and reliability.
Lab users	Who work within a laboratory setting and are responsible for conducting tests, analysing samples, and managing various laboratory processes. Their roles are integral to the effective operation of clinical, research, or industrial laboratories.
Normal user	The patient wants to view their test results.

Prepared By 0.7 6/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

2.3 Use case diagram of the system

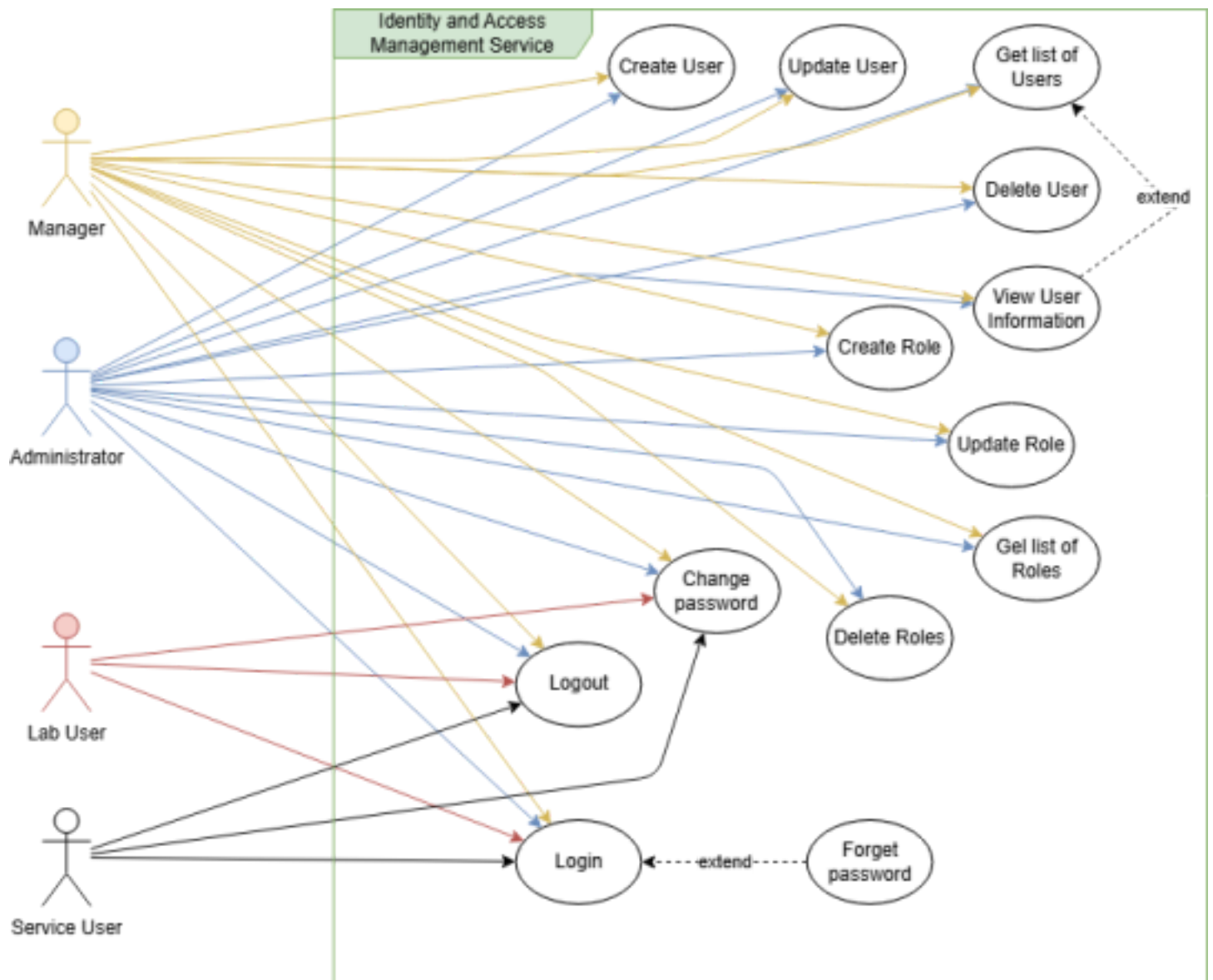


Figure 1: IAM use case diagram

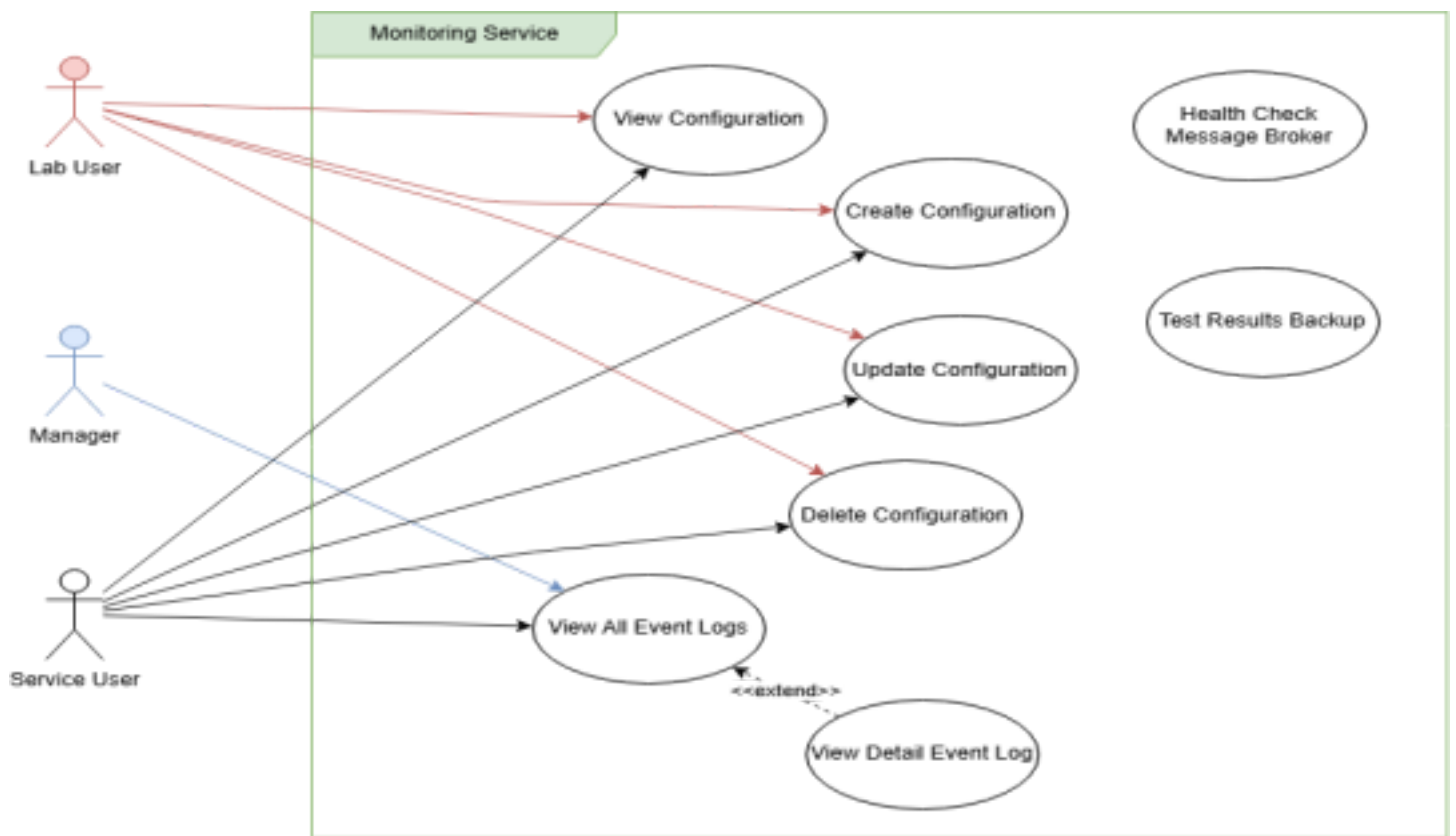


Figure 2: Monitoring use case diagram

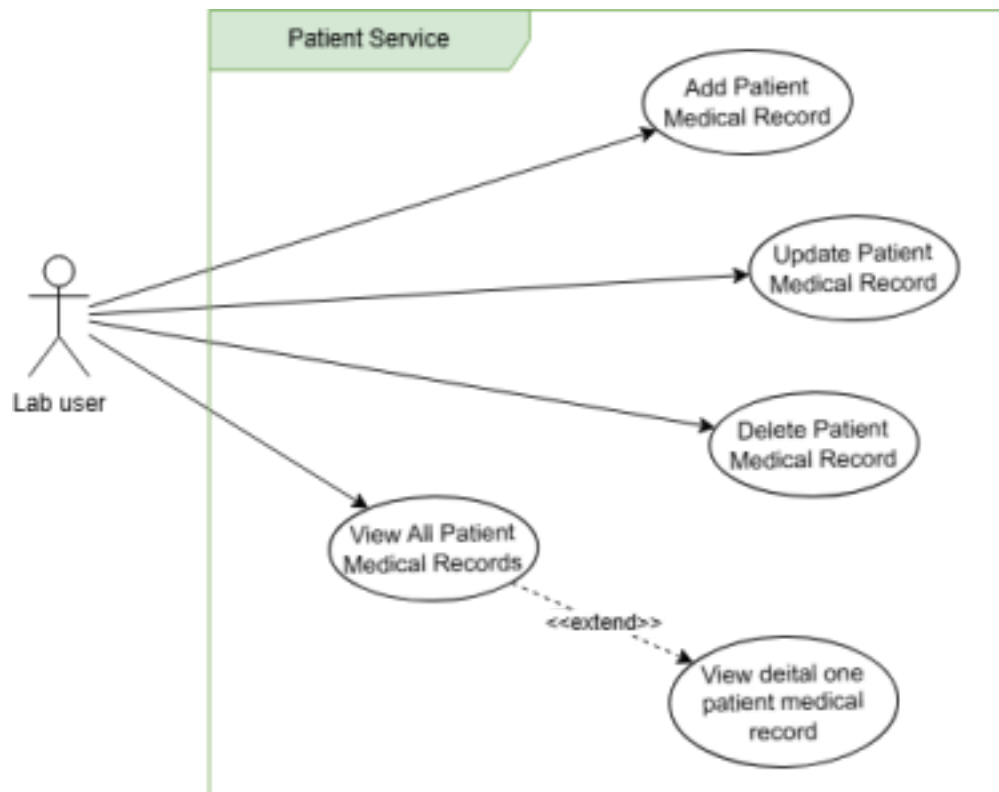


Figure 3: Patient use case diagram

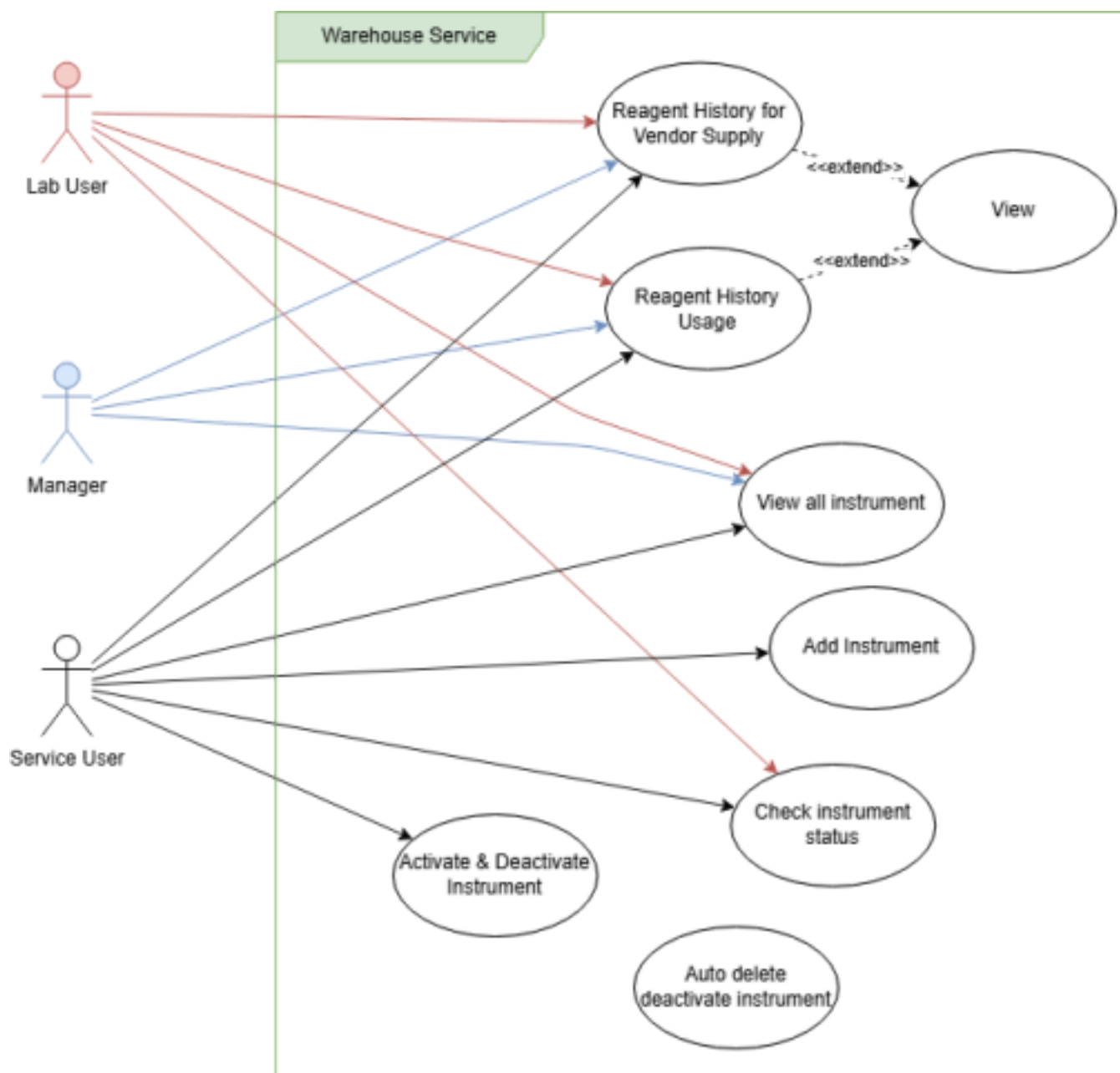


Figure 4: Warehouse use case diagram

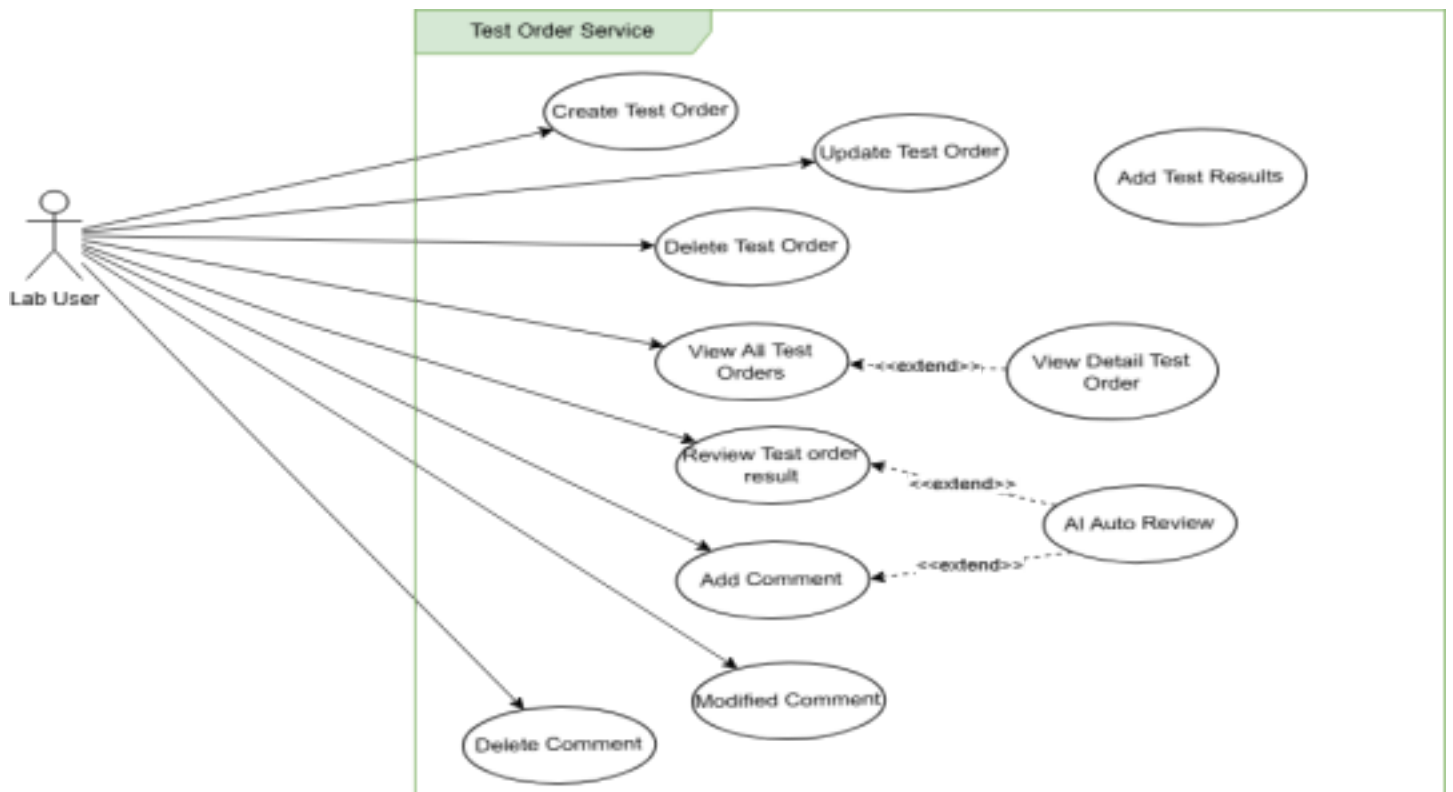


Figure 5: Test Order use case diagram

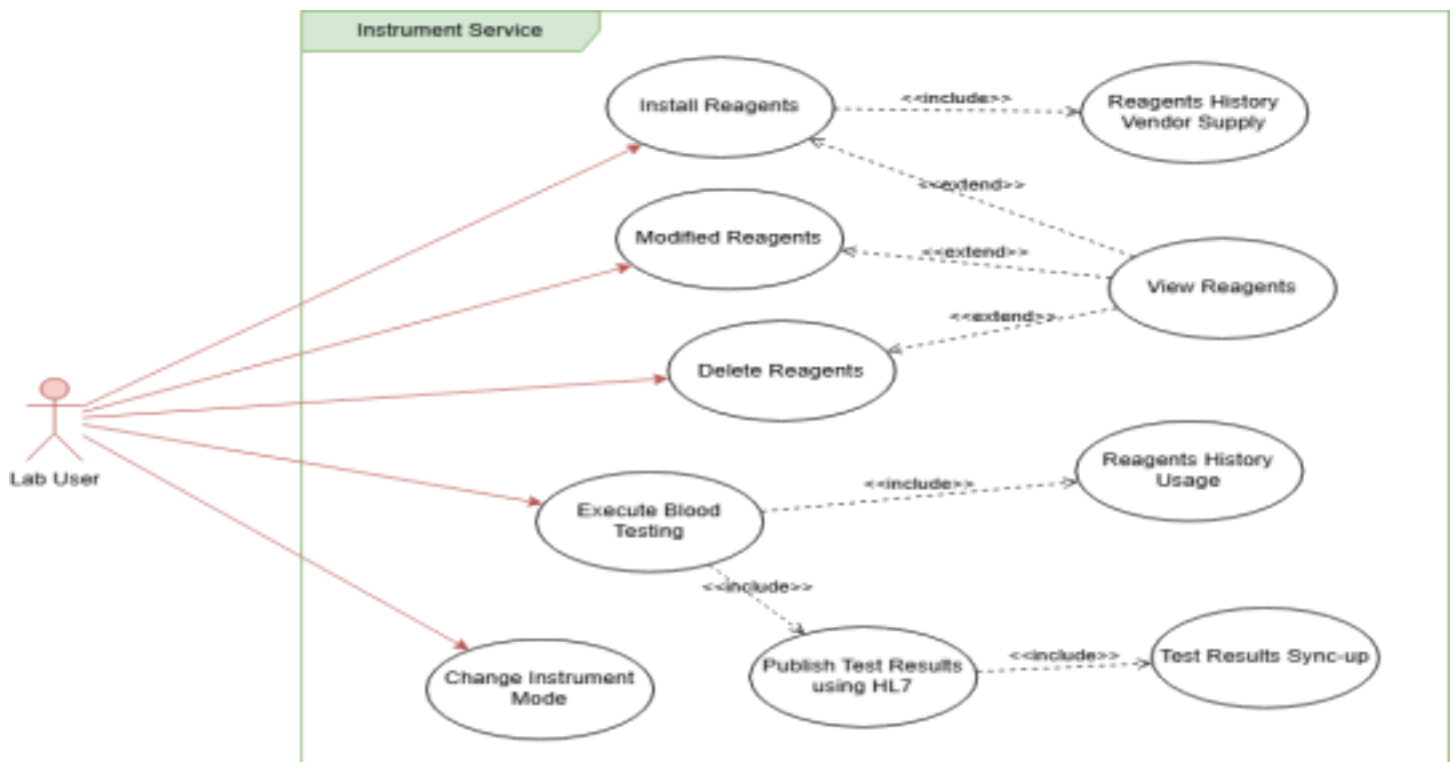


Figure 6: Instrument use case diagram

2.4 Configurations List

This table defines the configurations that can be used in the system. It can be updated or added with more configuration in the future.

Configuration name	Description	Default value
Lockout Policy	This security feature is activated when a user enters an incorrect password multiple time. It temporarily locks the account to prevent unauthorized access and safeguard sensitive information.	5 times
AI Auto review	An automated system that evaluates test order results without manual intervention. It enhances efficiency by identifying discrepancies, verifying accuracy, and ensuring compliance with predefined criteria.	off
Session Timeout	Determines the duration a user session remains active. After a set period of inactivity, the session automatically expires to maintain security and prevent unauthorized access.	15 minutes
Password Policy	Enforces password complexity requirements, expiration timelines, and reset procedures to strengthen security.	Minimum Length: 8– 12 characters (stronger security recommends 12+) Complexity Requirements: Must include uppercase & lowercase letters, numbers.
Expired Password	Ensures security by requiring users to update their passwords regularly	90 days
Automatic Deactivation	Once an account surpasses the inactivity threshold, it is automatically disabled to prevent unauthorized access.	30 days

2.5 Reagents List

This table defines the reagents needed for use during a blood testing process. Each time, the amount specified in this table must be used.

Reagents	Description	Usage per Run
----------	-------------	---------------

Diluent	Typically used in a 1:10 to 1:20 ratio with blood samples to maintain cell integrity. Used to dilute blood samples to ensure accurate counting of blood cells.	1 – 2 ml
---------	---	----------

Prepared By 0.7 11/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Lysing	Often added in precise microliter amounts (e.g., 50–200 µL) to break down red blood cells for white blood cell analysis.	50 – 200 µL
Staining	Used to stain specific blood components, such as reticulocytes for differential analysis.	50 – 100 µL
Clotting	Prevents the sample from clotting to allow smooth flow through the analyser.	50 – 100 µL
Cleaner	Clean the sample tubing, preventing contamination and clotting.	1 – 2 ml

2.6 Parameters List

A **Complete Blood Count (CBC)** test measures various components of the blood to help assess overall health and detect medical conditions. Here's a typical table of **normal range values** for key blood components:

Parameter	Description	Abbreviation	Normal Range
White Blood Cell Count	Measures the number of white blood cells (leukocytes) in the blood, which helps fight infection.	WBC	4,000–10,000 cells/µL
Red Blood Cell Count	Measures the number of red blood cells per unit of blood, which are responsible for carrying oxygen throughout the body.	RBC	Male: 4.7–6.1 million/µL Female: 4.2–5.4 million/µL
Hemoglobin	Measures the amount of hemoglobin in the blood, which is the protein in red blood cells that carries oxygen.	Hb/HGB	Male: 14–18 g/dL Female: 12–16 g/dL
Hematocrit	Represents the percentage of red blood cells in the blood volume, indicating oxygen-carrying capacity.	HCT	Male: 42–52% Female: 37–47%
Platelet Count	Measures the number of platelets in the blood, which are responsible for	PLT	150,000–350,000 cells/µL

	clotting.		
Mean Corpuscular Volume	Indicates the average size of red blood cells.	MCV	80–100 fL
Mean Corpuscular Haemoglobin	Represents the average amount of haemoglobin per red blood cell.	MCH	27–33 pg
Mean Corpuscular Haemoglobin Concentration	Calculates the average concentration of haemoglobin in red blood cells.	MCHC	32–36 g/dL

Prepared By 0.7 12/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Reference: [Complete Blood Count \(CBC\) - Lab Results - Hematology](#)

2.7 Privileges List

This table describes privileges for a default role for actors: **Admin, Service, Lab Manager, Lab users**, and a **Custom Role** with custom privileges.

Privilege	Description	Admin	Lab Manager	Service	Lab User	Custom Role (Default)
Read-only	Only have right to view patient test orders and patient test order results.	x	x			x
Create Test order	Have right to create a new patient test order	x			x	
Modify Test order	Have right to modify information a patient test order.	x			x	
Delete Test order	Have right to delete an exist test order.	x			x	
Review test order	Have right to review, modify test result of test order	x			x	
Add comment	Have right to add a new comment for test result	x			x	
Modify comment	Have right to modify a comment.	x			x	

Delete comment	Have right to delete a comment.	x			x	
View configuration	Have right to view, add, modify and delete configurations.	x		x		
Create configuration	Have right to add a new configuration.	x		x		
Modify configuration	Have right to modify a configuration.	x		x		
Delete configuration	Have right to delete a configuration.	x		x		
View user	Have right to view all user profiles	x	x			
Create user	Have right to create a new user.	x	x			
Modify user	Have right to modify a user.	x	x			
Delete user	Have right to delete a user.	x	x			
Lock and Unlock user	Have right to lock or unlock a user.	x	x			
View role	Have right to view all role privileges.	x	x			
Create role	Have right to create a new custom role.	x	x			

Prepared By 0.7 13/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Update role	Have right to modify privileges of custom role.	x	x			
Delete role	Have right to delete a custom role.	x	x			
View Event Logs	Have right to view event logs	x	x	x	x	
Add Reagents	Have right to add new reagents.	x		x	x	
Modify Reagents	Have right to modify reagent information.	x		x	x	
Delete Reagents	Have right to delete a reagents	x	x	x	x	
Add Instrument	Have right to add a new instrument into system management	x	x	x	x	
View Instrument	Have right to view all instrument and check instrument status.	x	x	x	x	
Activate or Deactivate	Have right to activate or deactivate instrument	x	x	x	x	

Instrument						
Execute Blood Testing	Have right to execute a blood testing	x		x	x	

2.8 Event Table

Event Id	Description
E_00001	Event message used when a new test order is created.
E_00002	Event message used when a test order is updated.
E_00003	Event message used when a test order is deleted.
E_00004	Event message used when a test result is modified.
E_00005	Event message used when new comment of test result is added.
E_00006	Event message used when comment of test result is modified.
E_00007	Event message used when comment of test result is deleted.
E_00008	Event message used when completed review.
E_00009	Event message used when activate or deactivate instrument.
E_00010	Event message used when lock or unlock a user.

3. Software Requirements Specification

The software requirements are group by service.

Prepared By 0.7 14/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

3.1 Identity and Access Management Service

3.1.1 Users Management

3.1.1.1 Create User

In the app, user can create new user accounts by clicking the "Create New User" button. This action opens a form where several required pieces of information must be inputted. The required fields include:

- ❖ Email (must be in a valid format)
- ❖ Phone number (must be in a valid format)
- ❖ Full name
- ❖ Identify number (must be in a valid format)
- ❖ Gender (must be either male or female)
- ❖ Age
- ❖ Address
- ❖ Date of birth (must be in MM/DD/YYYY format)

- ❖ Password (must adhere to a valid password format)

Once the required fields are filled out, the user can submit the form to create the new account. If the submission is successful, the system will create the new user account and log the event to track any changes made. However, if there are errors in the form input, such as invalid formats or missing information, an error message will appear, prompting the user to fill out the form again. The system will validate every required field before processing the request. If all fields pass validation, a new record is inserted into the database. If any validation fails, the user will receive a specific error message indicating which field(s) need correction.

3.1.1.2 Update User

In the application, user can update user information by clicking the "Update User Information" button. When this button is clicked, the user is required to input several fields of information, including:

- ❖ Full name
- ❖ Date of birth (must be in MM/DD/YYYY format)
- ❖ Age
- ❖ Gender (must be either male or female)
- ❖ Address
- ❖ Email (must be in a valid email format)
- ❖ Phone number (must be in a valid format)

After entering the required information, the user can submit the update. If the update is successful, the information for the user will be changed, and a confirmation of the update will be provided.

However, if there are errors in the provided information, such as invalid formats or missing fields, the system will display an error message, prompting the user to enter the information again. Before processing the update request, the system will validate all required fields. If all fields pass validation, the system will update the user's information in the database. If any field fails validation, the user will receive specific error messages indicating which field(s) need correction.

3.1.1.3 Get list of Users

In the application, user can view a list of users. This list can be organized based on specific criteria, allowing users to easily find and sort through information. Users can search for specific values related to user details, filter results based on selected criteria, and sort the list according to the columns provided.

Prepared By 0.7 15/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Once the filters and sorting options are applied, the system will display the list of users. If there are no users in the system, a message stating "No Data" will be shown instead of an empty list.

The default view should sort the data by full name in the designated column. Each user record displayed should include the following details:

- ❖ Full name
- ❖ Email
- ❖ Phone number
- ❖ Identify number
- ❖ Gender
- ❖ Age
- ❖ Address
- ❖ Date of birth

3.1.1.4 Delete User

In the application, user can delete a chosen user account. When the user selects an account to delete, the system processes the request to remove that account from the system.

If the deletion is successful, the chosen user account will be permanently removed, and an appropriate confirmation will be provided. Additionally, the system will log the event to track audit information, including details about the deletion and the identity of the administrator or manager who performed the action.

However, certain conditions must be met for the account to be deleted. The user account must not have been deleted previously, and it must exist in the system. If the account is already deleted or cannot be found, the system will inform the administrator or manager about this issue, preventing any unnecessary confusion.

3.1.1.5 View User Information

In the application, user can view detailed information for a specific user account. When the user requests to view an account, the system retrieves and displays the comprehensive details associated with that account. The output will show the complete account information, allowing the administrator or manager to review necessary data. However, certain conditions must be met for successful viewing. Specifically, the user account must not have been deleted, and it must exist in the system.

If the requested account is not found in the database, the system will inform the user that the account is not available, helping to maintain clarity and prevent confusion.

3.1.1.6 Change password

In the application, user can change a user account's password. When a user initiates the process to change the password, the system allows them to input the new password.

Once the new password is submitted, it will be updated in the database if the change is successful. After the password change is completed, the system will log the event to capture details about who changed the password and when the change occurred.

Prepared By 0.7 16/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

However, there are certain conditions to ensure the password change is valid. If the new password is the same as the old password, an error will appear, notifying the user that the new password must be different. Additionally, the new password must adhere to a valid password format; if it does not meet the required criteria, the system will return specific error messages to inform the user of the issues, such as if the old password is incorrect or if the new password does not meet validation standards.

3.1.2 Roles Management

3.1.2.1 Create Role

In the application, user can create a new role by providing certain required information. The necessary details to create a role include:

- ❖ Role name
- ❖ Role code
- ❖ Role description
- ❖ Add privileges for it or it will get a default privileges: Read-only

Once the required fields are filled out and submitted, the system will attempt to create the new role. If the creation process is successful, the system will confirm that the role has been created successfully.

However, specific conditions need to be satisfied for the role to be created. If a role with the same code already exists in the system, the new role cannot be created. To prevent duplicates and ensure data integrity, the system must validate all submitted information before processing the creation request.

Once the validation is complete, if all fields pass the necessary checks, a new record will be successfully inserted into the database. Conversely, if validation fails, the system generates specific error messages indicating which fields are invalid or problematic, guiding the user to correct the issues before re-attempting to create the role.

3.1.2.2 Update Role

In the application, user can update a role by providing the necessary information. To update a role, the user must input the following details:

- ❖ Role name
- ❖ Role description
- ❖ Privileges

Once the required fields are filled out and the update request is submitted, the system will attempt to update the role information. If the update is successful, a confirmation will be provided indicating that the role information has been successfully updated.

However, before processing the update request, the system must validate all required fields to ensure that the input is correct. If all fields pass validation checks, the system will proceed to update the record in the database. In the event that validation fails, the system will return specific error messages for each invalid field, allowing the user to correct any issues before re-submitting the update.

Additionally, if the update is executed successfully, the system will log the change, capturing details about which fields were modified and by whom. If there is a failure in updating the role for any reason, an error message will be displayed to inform the user of the unsuccessful attempt.

Prepared By 0.7 17/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

3.1.2.3 Get list of Roles

In the application, user can view a list of roles. This list can be organized through various functionalities, including searching for specific values that match the columns or filtering the data. Additionally, users can sort the roles by the columns available in the list.

When the users request to view the list, the system will display the roles accordingly. If there are no roles present in the system, a message stating “No Data” will be shown instead of an empty list, ensuring that users receive clear

feedback. By default, the data is sorted by the role name in the designated column, making it easier for users to navigate through the information. Each record in the list will provide the following details:

- ❖ Role name
- ❖ Role code
- ❖ Role description
- ❖ Privileges

3.1.2.4 Delete Role

In the application, user can delete a chosen role. When a user selects a role for deletion, the system processes the request to remove that role from the system.

If the deletion is successful, the selected role will be permanently removed, and the system will confirm that the role has been deleted.

However, certain conditions must be met for a role to be deleted. Specifically, the role must not have been previously deleted, and it must exist within the system. If the role has already been deleted or cannot be found, the system will notify the user of this issue.

Furthermore, when the deletion is executed successfully, the system will log the event, capturing details about who deleted the role and the role's information for future reference and auditing.

3.1.3 Authentication/Authorize

3.1.3.1 Login

In the application, users can log into the system by entering their credentials. The login process requires the user to provide their username and password.

Upon successful login, the user is granted access to the system and redirected to the homepage. This signifies that the user has been authenticated and can now utilize the application's features.

However, if there are any errors during the login attempt—such as entering an incorrect username or password—the system will display an error message, informing the user that the login has failed.

For the login process to be successful, certain criteria must be met: the user account must exist within the system, and the provided credentials must be valid. If these conditions are satisfied, the system will allow the user to log in and proceed to the homepage.

3.1.3.2 Logout

In the application, users can log out of their current session. When a user chooses to log out, the system will process the logout request.

Upon successful logout, the system will redirect the user to the login page and clear the session information associated with the logged-in user. This ensures that no remaining session data or cookies are left, effectively ending the user's access to the application.

However, if there is an issue with validating the login status, such as an incomplete session or other discrepancies, the system will display an error message, indicating that the logout process could not be completed. To ensure a smooth logout process, the system must successfully validate the current session status before clearing it. Additionally, it must clear the session or cookie information associated with the user properly.

3.1.3.3 Forget password

In the application, users can change their password if they forget it. When a user initiates the password reset process, they are prompted to enter their registered email address.

Upon verification, if the email exists in the system, the user will be allowed to proceed with creating a new password. Once the new password is submitted and successfully updated, the user will then be able to log in using their new password. If the user inputs an incorrect email address that does not exist in the system, the system will display an error message to inform the user of the issue.

For the process to be successful, the email provided must be associated with an existing account in the system. Additionally, the new password must comply with a valid password format as defined by the system's requirements. If these criteria are met, the user can successfully log in with the new password after it has been updated.

Moreover, when the password is changed successfully, the system will maintain an event log to capture details about when the password was updated, for auditing purposes.

3.2 Monitoring Service

3.2.1 Event Logs Management

3.2.1.1 Add New Event Log

In the application, whenever a new event message is published, the service is designed to capture this message automatically and insert it into the database.

Once the event message is successfully captured, a new event log will be created and stored within the system. This ensures that all relevant events are logged for future reference and analysis.

There are no specific actors involved in this process as it is automated by the service. Additionally, there are no alternative flows or logs associated with this procedure.

Prepared By 0.7 19/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

To meet the acceptance criteria, it is essential that the new event log is inserted into the database correctly and that all data contained in the event logs matches the published event message precisely. This requirement guarantees the integrity and accuracy of the logged information.

3.2.1.2 View List Event Logs

In the application, users can view a list of event logs. This functionality allows users to search for specific values by filtering through the columns or to sort the logs based on the available columns.

When a user accesses the event log feature, the system displays the list of patient test orders as captured in the logs. If the search yields no results, the system will indicate this by showing an empty list, along with a message that state “No Data.”

The data in the event logs should be sorted by the latest date in the designated column by default, ensuring that the most recent events are displayed prominently. Each record in the event log view will contain the following details:

- ❖ Action
- ❖ Event Log Message
- ❖ Operator (the user who performed the action)

3.2.1.3 View Event Log's Detail

In the application, users can view detailed information about selected event logs. When a user requests to view an event log, the system retrieves and displays the comprehensive details associated with that specific log. If the requested event logs are found within the system, the user will see all relevant data pertaining to those logs. However, if the system does not contain the requested event logs, the user will receive a notification indicating that the logs were not found.

To ensure a smooth operation, it is essential that the event logs have not been deleted and that they exist within the system prior to the user attempting to access them. If these criteria are met, the user can successfully view the detailed event log information.

3.2.2 Test Results Backup

3.2.2.1 Add New Raw Test Results

In the application, whenever a new event message with test results are published, the service is designed to capture the message automatically and insert it into the database for backup.

Once the test results message is successfully captured, a new backup data will be created and stored within the system. This ensures that all relevant raw test results are logged for future reference and can trace back when have any issues or can be sync-up test results when the other service was restarted.

There are no specific actors involved in this process as it is automated by the service. Additionally, there are no alternative flows or logs associated with this procedure.

To meet the acceptance criteria, it is essential that the new event log is inserted into the database correctly and that all data contained in the event logs matches the published event message precisely. This requirement guarantees the integrity and accuracy of the logged information.

Prepared By 0.7 20/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

3.2.2.2 Sync-up Raw Test Results

In the application, whenever a sync-up message is published by the Test Order Service, the system is designed to capture and handle the request automatically. Upon receiving the sync-up message, the service parses the message content and iterates through the test order details to validate, compare, and append relevant information to a list of data for

further processing.

Once the processing is complete, the system republishes the refined sync-up message for the Test Order Service to handle downstream operations.

If the system identifies a test order without any associated test results, it will automatically publish a request message to the Instrument Service. This message instructs the Instrument Service to retrieve and return the raw data for the corresponding test order. The raw data will then be processed by the system to ensure the test order has complete and accurate information. This mechanism ensures that every test order has the required results for operational consistency.

3.2.3 Health Check

3.2.3.1 Message Broker Health Check

In the application, the system shall automatically and continuously monitor the health and operational status of the configured Message Broker(s) to ensure their availability and proper functioning. This proactive monitoring mechanism is crucial for identifying potential issues before they impact critical services.

The system will perform **automated health checks** on the Message Broker at configurable intervals (e.g., every 60 seconds) to verify connectivity and operational status. If the broker is unresponsive, the system will retry the check (e.g., 3 times) and, upon confirmed failure, log a **detailed error event** (timestamp, error code, retry attempts) to a centralized logging system for troubleshooting. Once the broker recovers, a **restoration event** is logged. The system ensures minimal performance impact during checks and retains logs for **≥30 days**. If the broker is down, dependent operations (e.g., message publishing) pause until connectivity is restored, while non-dependent functions continue unaffected. This ensures reliability and rapid issue resolution.

3.3 Warehouse Service

3.3.1 Instrument Management

3.3.1.1 Add new instrument

In the application, users can add a new instrument along with its associated reagents and configurations. When creating a new instrument, users have the flexibility to leave the reagents or configurations empty, or they can opt to clone these settings from another existing instrument.

Once the new instrument is successfully inserted into the system, a confirmation is provided to indicate that the process has been completed successfully. Additionally, the system will log this action to maintain an audit trail, capturing details about who inserted the new instrument, as well as the associated reagents and configurations.

It is essential that the instrument being added must not already exist in the database. If a duplicate instrument is detected, the system will prevent the insertion to maintain data integrity.

Prepared By 0.7 21/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

3.3.1.2 View all instruments

In the application, users can view a complete list of instruments. When a user accesses this feature, the system retrieves and displays all instruments available in the database.

The output should present a comprehensive list, with data sorted by the latest date in the designated column by default. This arrangement ensures that the most recently added or updated instruments are displayed prominently for user convenience. If there are no instruments available in the system, a message stating “No Data” will be shown instead of an empty list, providing clear feedback to the user.

3.3.1.3 Check instrument status

In the application, users can check the status of an instrument to determine its current state. When a user requests to view the instrument status, the system responds by providing the current state of the instrument, whether it be "Ready," "Processing," "Maintenance," or "Error."

If an instrument is found to be in an "Error" state, the system is designed to automatically perform a recheck when the user attempts to use the instrument. Following this recheck, if the instrument can transition to a "Ready" state, it will do so. However, if the recheck reveals that the instrument remains in an "Error" state, the system will notify the user with a detailed error message, outlining the issue.

The acceptance criteria for this process indicate that the system must successfully return the state of the instrument. If the instrument is in an "Error" state and fails to switch to "Ready" after the recheck, it must remain in the "Error" state, ensuring that users are fully informed about any ongoing issues.

3.3.1.4 Activate & Deactivate Instrument

In the application, users can activate or deactivate instruments to indicate their availability for use. When an instrument is marked as "Inactive," it signifies that the instrument is currently unavailable due to maintenance or repair needs. Upon deactivation, the status of the instrument is updated accordingly, and confirmation is provided that the instrument is now marked as "Inactive."

If a user later determines that the instrument is ready for use again, they can reactivate it, allowing for its reassignment to test orders. Notably, the system is designed to prevent any test orders from being assigned to deactivated instruments, maintaining operational integrity and ensuring that users do not attempt to use non-functional equipment.

Additionally, as part of the deactivation process, the system will create a scheduler that automatically deletes instruments that have been deactivated for three months, thereby helping to keep the system organized and efficient. When an instrument's status is successfully updated, the system logs the action by saving an event record, which includes details about who performed the action, the timestamp, and the specific instrument involved.

3.3.1.5 Auto delete deactivated instrument

In the application, when an instrument is manually deactivated, it initiates the process to eventually delete that instrument from the system. Once the deactivation is confirmed, the system will remove the instrument from the list of

available instruments.

After a successful deletion, the system will log this action, capturing the timestamp for when the instrument was deleted, which helps maintain an accurate audit trail. If the instrument is re-activated before it is deleted, the system will automatically cancel the deletion process, ensuring that it will not be removed.

The acceptance criteria for this process stipulate that the instrument must be successfully deleted from the system upon manual deactivation. Conversely, if the instrument is activated again, the scheduled deletion will be cancelled, and the instrument will remain in the database, preserving its availability for use.

3.3.2 Reagents History Tracing

3.3.2.1 Reagents History for Vendor Supply

In the application, users can access and view a comprehensive history of reagents supplied by vendors. When a user requests this history, the system processes the request to provide detailed records of all incoming reagent shipments. This allows users to track procurement, verify deliveries, and manage inventory based on vendor supply information. This history record for each received reagent shipment shall include, but not be limited to:

- ❖ Reagent Information: Name, Catalog Number, Manufacturer, CAS Number (if applicable).
- ❖ Vendor Details: Vendor Name, Vendor ID.
- ❖ Supply/Order Details: Purchase Order (PO) Number, Order Date, Receipt Date, Quantity Received, Unit of Measure.
- ❖ Batch Information: Lot Number, Expiration Date.
- ❖ Receiving Information: User who received the shipment, Date and Time of receipt, Initial Storage Location.
- ❖ Status: (e.g., "Received," "Partial Shipment," "Returned").

Additionally, when a new reagent shipment is received and logged into the system, the system will automatically create a new entry in this vendor supply history. This automated logging ensures that a complete and accurate audit trail is maintained for every reagent entering the inventory, supporting accountability and compliance with regulatory standards.

It is critically important that the data in the vendor supply history remains accurate, complete, and immutable (append only) to ensure traceability. If a user requests historical data for a specific reagent, a specific vendor, or within a defined date range, the system shall filter and present only the relevant records. However, if no specific filters are applied, the system shall display the full history of all reagent shipments received from all vendors. Furthermore, the system ensures that the historical records are immediately available upon a new receipt, reflecting the most up-to-date state of incoming inventory.

3.3.2.2 Reagents History Usage

In the application, users can view the history of reagent usage. When a user requests to see the history, the system processes the request and provides detailed records that include information about each reagent's usage, such as the quantity, date and time of use, and the responsible user. This allows users to review and track all previous interactions with reagents for operational and auditing purposes.

Prepared By 0.7 23/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Additionally, when a new reagent usage occurs, the system automatically logs this event in the history, storing essential details like the reagent's name, the action performed, the user who performed it, and the timestamp. These logs

help maintain an audit trail for accountability and ensure traceability for compliance with regulatory requirements.

It is critically important that the data in the reagent usage history remains accurate, complete, and tamper-proof. If a user requests historical data for a specific reagent, the system filters and provides only the relevant records for that reagent. However, if no specific reagent is specified, the system displays the full history of all reagent usage. Furthermore, the system ensures that any changes to reagent usage (such as edits or deletions) are immediately reflected in the history records to maintain consistency across the platform.

Access to reagent history is restricted to authorized users only, ensuring sensitive data about reagent handling is protected.

3.3.3 Configurations Management

Configurations need to be saved in the Warehouse Service, and other services must synchronize these configurations whenever they are created, updated, or deleted. Can reference Section: 2.4 Configurations List and 2.5 Parameters List

3.3.3.1 Create Configurations

In the application, users can create a new configuration based on the available configurations list. When a user decides to create a new configuration, they must fill out the necessary fields required by the system. Upon successful submission of the configuration details, the system will process the request and create the new configuration, confirming that the configuration has been created successfully.

Importantly, when a new configuration is created, it will also sync up to other services automatically, ensuring that all relevant systems are updated with the latest configuration settings. It is crucial that the system validates all required fields before proceeding with the request to ensure that all necessary information is complete. Additionally, the configuration being created must be unique, and the system must prevent any duplication of existing configurations.

3.3.3.2 Modify Configurations

In the application, users can modify the value of an existing configuration. When a user decides to make changes, they can input the new value for the configuration. Once the modification is submitted, the system processes the request and updates the configuration value successfully.

Upon a successful update, the system will save an event log to track which fields have been changed and by which user, ensuring proper auditing of configuration changes. Additionally, it is important that the modified configuration must not be deleted and must exist within the system for the update to be valid. Furthermore, after a configuration has been modified, the updated value will also sync up to other services automatically, ensuring that all relevant systems are informed of the latest configuration settings.

3.3.3.3 Delete Configurations

In the application, users can delete a configuration. When a user chooses to proceed with the deletion, the system processes the request to remove the specified configuration from the system. Upon successful deletion, the system confirms that the configuration has been successfully deleted.

Prepared By 0.7 24/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Additionally, when a configuration is deleted, the system will log this action to keep track of which user performed the deletion. This event log helps maintain an audit trail for accountability purposes. It is critically important that the

configuration must exist in the system for the deletion to occur; if it does not exist, the deletion cannot be processed. Furthermore, once a configuration has been deleted, this change will sync up with other services automatically to ensure that all relevant systems reflect the updated state.

3.3.3.4 View Configurations

In the application, users can view the configurations stored in the system. When a user chooses not to provide any specific configuration input, the system processes the request by returning all configurations available in the system. Upon successfully retrieving all configurations, the system confirms the delivery of the configuration data in a structured format such as JSON, containing all key-value pairs.

Additionally, when the user specifies a particular configuration to view, the system processes the request to return only the requested configuration. This ensures that the user can focus on specific data without additional clutter from unrelated configurations. It is critically important that the specified configuration must exist in the system; if it does not exist, the retrieval cannot be processed, and an appropriate error message will be returned to notify the user. If the input provided is invalid or malformed, the system ensures the user is informed of the issue by returning an error message.

3.4 Patient Service

3.4.1 Patient Medical Record Management

3.4.1.1 Add Patient Medical Record

In the application, users can create or add a new patient medical record. The process involves entering relevant patient information into the system, resulting in the successful addition of a new record.

To ensure proper management and security of patient records, several acceptance criteria must be met. First, only authorized users—such as doctors, lab supervisors, and administrators—are permitted to access the full list of patient medical records, reinforcing the importance of data privacy and compliance with regulations like HIPAA, GDPR, PII.

Once the record is added, the system must display a searchable and paginated list of all patients, including critical information such as Patient ID, full name, date of birth, and last test date. Users should be able to search for patients using multiple parameters like name, ID, or date of visit, and filter the displayed list by test type, date range, or the instrument used.

Selecting a patient from the list should open their detailed medical record view, enabling users to access comprehensive information regarding that patient's medical history and current treatments. Additionally, the system must log every access to a patient's medical record for auditing and compliance purposes to maintain a transparent and accountable healthcare environment.

3.4.1.2 Update Patient Medical Record

In the application, users can update existing patient medical records. This process involves modifying relevant information within the record, which results in the successful update of the patient's medical information. To maintain security and integrity, certain acceptance criteria must be fulfilled. Only authorized users, such as doctors, lab technicians, and administrators, are permitted to make updates to patient medical records. The system allows for the updating of several key fields, including:

- ❖ Patient demographic information (such as name and contact details)
- ❖ Test results and their interpretations
- ❖ Details of instruments and reagents used during testing
- ❖ Clinical notes or comments related to the patient's care

Before any updates are processed, the system must validate that all required fields are filled correctly and that data is within acceptable ranges. Additionally, the system logs all updates for auditing purposes, capturing important details such as the identity of the user making the change, the timestamp of the update, and the specific fields that were modified.

Upon successful completion of the update, users will receive a confirmation message. Conversely, if the update fails due to missing data or other permission-related issues, the system will display an appropriate error message to guide the user. Furthermore, the system will retain previous versions of the record, ensuring that a comprehensive version history is available for audit and rollback purposes.

3.4.1.3 Delete Patient Medical Record

In the application, users have the authority to delete patient medical records when necessary. When a user initiates a deletion, the system responds by processing the request, which results in the successful removal of the specified record. To ensure this process is secure and well-regulated, several acceptance criteria must be adhered to. First, only users with administrative privileges are authorized to delete a patient's medical record, ensuring that such actions are restricted to qualified personnel. Before proceeding with the deletion, the system must prompt the user with a confirmation dialog, clearly warning them that the action is irreversible.

Upon successful deletion, the system logs the event to maintain an audit trail. This log must include details such as the identity of the user who performed the deletion, the timestamp of the action, and pertinent information about the patient ID and the record that was deleted.

Additionally, the system must prevent deletion if the medical record is linked to any ongoing or pending test orders, safeguarding against potential disruption in patient care. Once the deletion is successfully completed, the system will display a confirmation message informing the user of the successful removal.

If the deletion fails, due to permission issues, system errors, or other conflicts, the system must provide an appropriate error message to guide the user. Furthermore, in compliance with data retention policies or regulations, the system should support soft-delete options (such as archiving or backup) to maintain legal and regulatory compliance regarding patient data.

3.4.1.4 View All Patient Medical Records

In the application, users can view a comprehensive list of patient medical records. This feature allows users to efficiently search and filter the list based on specific criteria. Users can search for patients by various parameters, including name, Patient ID, or date of visit, ensuring quick access to relevant records. Additionally, users can sort the data by different columns and filter results by test type, date range, or the instrument used for testing.

The system is designed to display a paginated list of all patients, including essential information such as Patient ID, full name, date of birth, and last test date. This functionality ensures that users can easily navigate through the records, with the option to select any patient from the list to view their detailed medical record. Access to patient medical records is restricted to users with appropriate permissions, such as administrators, doctors, and lab supervisors, to maintain confidentiality and comply with data privacy regulations.

Every access to patient records is logged for audit and compliance purposes, ensuring that the system maintains a clear record of who accessed which information. Additionally, the system enforces role-based access control to uphold data privacy and ensure compliance with relevant regulations, such as HIPAA and GDPR.

3.4.1.5 View detail one patient medical record

In the application, users can view detailed medical records for selected patients. This process begins when a user selects a specific patient, prompting the system to retrieve and display the relevant detailed information associated with that patient.

The system ensures that only authorized users, such as doctors and lab technicians, are permitted access to patient medical records, safeguarding sensitive information. Upon selecting a patient, the system presents a comprehensive overview that includes vital details such as:

- ❖ Patient ID, name, date of birth, and contact information
- ❖ A list of all test orders associated with the patient, including timestamps
- ❖ Detailed test results, presenting values along with reference ranges and interpretations
- ❖ Information about the instrument used for each test
- ❖ The batch and lot number of reagents utilized in the testing process
- ❖ Any error or warning messages linked to the test results
- ❖ Notes or comments that may have been added by lab staff or physicians regarding the patient's care

Moreover, the system provides functionality to filter records based on date range, test type, or the instrument used, enhancing the user's ability to find specific information quickly.

Each access to a patient's medical record is logged for audit purposes, ensuring accountability and compliance with data governance standards. If a record is found to be incomplete or contains flagged results, the system will display an appropriate warning, alerting users to review the information carefully.

3.5 Test Order Service

3.5.1 Test Order Management

3.5.1.1 View Patient Test Order

In the application, users can view a comprehensive list of patient test orders. This feature allows users to search and filter the list based on specific criteria, ensuring efficient access to relevant information. Users can search or filter by similar values within the columns, as well as sort the list according to various columns.

The system will display the list of patient test orders, sorting the data by the most recent date in the designated column by default. If no test orders are available, the system will clearly indicate this by displaying the message “No Data” instead of leaving the list empty.

Each record in the list provides essential details, including:

- ❖ Patient name
- ❖ Age
- ❖ Gender
- ❖ Phone number
- ❖ Status of the test order (e.g., Pending, Cancelled, Completed)
- ❖ Created date
- ❖ User who created the order
- ❖ Run date
- ❖ User who ran the test

3.5.1.2 View Detail Patient's Test Order.

In the application, users can access and view the detailed information associated with a patient's test order. This process allows users to examine all relevant data regarding the specific test order, providing insight into the patient's status and testing history.

For the viewing process to be successful, certain acceptance criteria must be met. The patient's test order must exist in the system and should not have been deleted. If the status of the test order is marked as "Completed," then the system must display all corresponding test result details. In contrast, if the test order is not completed, it is acceptable for the test results to be empty.

Furthermore, if there are any comments associated with the test order, the system must show all comments to provide a comprehensive overview of the patient's testing experience and any notes made by healthcare professionals. If the specified test order cannot be found or does not contain any test results data to display, an appropriate message should inform the user of this situation.

3.5.1.3 Create Patient's Test Order.

In the application, users can create a new patient's test order by entering essential information. The required fields for creating a new test order include the patient's name, date of birth, age, gender, address, phone number, and email. Upon successful submission of this information, the system processes the request and confirms that the new patient's test order has been created successfully. However, before the order can be finalized, the system must validate all required fields to ensure the accuracy and completeness of the data.

The email address provided must adhere to a valid email format, and the date of birth must be entered in the correct format (MM/DD/YYYY). If all fields pass these validation checks, the system will insert a new record into the database. Conversely, if the validation fails, the system will return error messages specific to each invalid field, guiding the user in correcting any mistakes. If the creation of the new patient's test order is unsuccessful, an appropriate alternative flow will be activated, indicating the failure of the process.

3.5.1.4 Modify Patient's Test Order

In the application, users can update a patient's test order by providing essential information. The required fields for updating a test order include the patient's name, date of birth, age, gender, address, and phone number. Upon inputting this information, the system processes the request to update the patient's test order. If the update is successful, the system confirms that the new details have been successfully applied to the record. To ensure the integrity of the data, the system must validate all required fields before processing the update request. For instance, the email must be in a valid email format, and the date of birth must be entered in the correct format (MM/DD/YYYY). If all fields pass validation checks, the system proceeds to update the record accordingly. In case the validation fails—due to incorrect data or missing information—the system will return specific error messages for each invalid field, guiding the user to make the necessary corrections.

Furthermore, when an update is made successfully, the system logs the event to maintain an audit trail, capturing information about the lab user who performed the update and the specific fields that were changed in the test order.

3.5.1.5 Delete Patient Test Order

In the application, users have the authority to delete a patient's test order when necessary. When a deletion request is initiated, the system processes the action to remove the specified test order from the records. Once the deletion is successful, the system confirms that the patient's test order has been deleted. For this process to proceed smoothly, certain acceptance criteria must be met. Firstly, the patient's test order must exist in the system; if an attempt is made to delete an order that has already been deleted or cannot be found, the system will identify this alternative flow and notify the user.

In addition, when a deletion is executed successfully, the system will maintain an event log that captures important details for auditing purposes. This log must include information about the lab user who performed the deletion, along with specifics concerning the deleted test order.

3.5.2 Patient Test Order Results Management

3.5.2.1 Add New Test Results

In the application, whenever a new event message containing test results is published, the service is designed to automatically capture the message and process the test results for database insertion.

Upon successfully capturing a test results message, the system reads the data formatted in the **HL7 (Health Level Seven)** structure, compares it, and transforms the raw data into processed data. This transformation ensures that raw test

results are converted into a structured, processed format that can be efficiently inserted into the database. This mechanism guarantees seamless integration of test results into the system's data repository.

Once the test results are successfully parsed and transformed, the system takes the **processed data** through flagging set configuration to flag and highlight parameters then inserted them into the designated database tables (Section 3.5.2.5). If parsing or validation fails (e.g., due to malformed HL7 data), the system logs the error and routes the message to a quarantine queue for manual review or reprocessing.

This process operates entirely as an automated service, with no specific actors involved. The design prioritizes reliability and ensures proper handling of test results, including real-time processing and recovery mechanisms during disruptions.

3.5.2.2 New Test Results Sync-up

To prevent data loss during service downtime or crashes, the system implements a **fault-tolerant backup flow**. All incoming raw HL7 messages are stored in a **persistent message queue** before processing or can get direct from **Monitoring Service**. If the service is interrupted, unprocessed messages remain in the queue. When the service restarts, it automatically retrieves and processes these queued messages, ensuring no test results are lost.

Refer to section 3.2.3.2 Sync-up Raw Test Results.

3.5.2.3 Review Test Order Results

In the application, users can update the status of a completed patient's test order. When a test order is selected for review, the system processes the request to update the status to "Reviewed." This process may also involve making minor adjustments to the test result values based on the review findings.

Once the update is successfully completed, the system confirms that the status of the patient's test order has been updated to "Reviewed," and any modified values are logged accordingly. Additionally, if required, an alternative flow may be utilized, such as an AI-driven automatic review of the test results, which can streamline the process.

For adherence to operational standards, several acceptance criteria must be met prior to proceeding with the update. The test order must not have been previously deleted, must exist within the system, and must be in a "Completed" status before it can be marked as "Reviewed." Furthermore, any changes made to the test result values must remain within acceptable ranges to ensure data accuracy.

Upon successful modification, the system logs the review action, capturing vital information such as the identity of the lab user who performed the review and the specific test order details that were adjusted.

3.5.2.4 AI Auto Review Test Order Results

In the application, users can utilize an AI Review Mode to automatically assess a completed patient's test order. When activated, the AI Review Mode initiates a review process, provided that there is enough data recorded to facilitate an automated review.

Upon engaging this mode, the system evaluates the completed test order and makes any necessary adjustments to the test result values, based on its analysis. Once the review process is successful, the status of the patient's test order is updated to "AI Reviewed," indicating that an AI-assisted evaluation has taken place.

Several acceptance criteria must be met for this process to proceed effectively. The patient's test order must exist within the system and must not have been previously deleted. Additionally, the order's status must be marked as "AI Reviewed" for the AI review to occur. Any changes made during the AI review process must remain within defined acceptable ranges, ensuring data accuracy.

Following the modification, the system logs the review event, capturing key details such as the timestamp of the AI review and the specific test order information that underwent assessment.

3.5.2.5 Flagging Set Configuration Sync-up

The system shall **synchronize flagging set configurations** (e.g., rules for abnormal results, thresholds, or alerts) and apply them to parsed test results to dynamically highlight critical or out-of-range values. This ensures that test results are automatically flagged based on the most up-to-date criteria.

The system shall continuously monitor and synchronize with the source of the flagging set configuration. This mechanism ensures that any updates to the configuration are promptly received by the system. The specific details of the flagging configuration (e.g., criteria, thresholds, associated flags) are defined in Section 3.3.3 of the documentation.

When the flagging set configuration is updated, the system shall immediately adopt the new configuration for all subsequent test result processing. No restart or manual intervention should be required to apply updated rules. After parsing test results from HL7 messages, the system shall apply the latest flagging configurations to:

- Identify results that meet criteria for highlighting (e.g., abnormal values, critical alerts).
- Flag results with visual indicators (e.g., color-coding, icons) in the user interface.

All changes to the flagging set configuration (e.g., new version deployed, specific rules modified) shall be logged, including timestamp and source of the update. Log instances where test results are flagged, including the applied rule and result value.

3.5.3 Comment Management

3.5.3.1 Add Comment

In the application, users can add a new comment to a test order or its corresponding results. This process involves inputting a comment that provides additional information or context regarding the test order.

Once the comment is successfully submitted, it is confirmed that the new comment is inserted into the system. To ensure the integrity of the comment entry, several acceptance criteria must be met. Each new comment must include identification of the user who made the comment, ensuring accountability. Additionally, the comment must contain a message relevant to either the test order or the test order results.

The system allows for multiple comments to be associated with a single test order or test order result, facilitating thorough documentation and communication among users. Importantly, comments cannot be left empty; there must be content within the comment field to ensure that meaningful information is conveyed.

3.5.3.2 Modify Comment

In the application, users can modify an existing comment associated with a test order. To initiate this process, the user selects a comment that they wish to change and inputs the new information they want to include. Once the modification is successfully applied, the system confirms that the comment has been updated with the new information. Several acceptance criteria must be met for this update to be valid. First, the comment being modified must exist in the system; if it has been deleted or does not exist, the update cannot proceed. Additionally, the new comment must not be empty; it should provide meaningful information related to the test order.

When the comment is successfully updated, the system records this action in an event log to maintain an audit trail. This log should capture details about the lab user who made the change and the specifics of the modified comment.

3.5.3.3 Delete Comment

In the application, users can delete a comment associated with a test order. When a user decides to delete a comment, they will select the specific comment they wish to remove from the system.

Upon successful deletion, the system confirms that the comment has been removed. To ensure the integrity of this action, several acceptance criteria must be met: the comment targeted for deletion must exist within the system, meaning it hasn't been previously deleted or removed. Additionally, the system must log the deletion action to maintain an audit trail; this log should capture details about the lab user who performed the deletion, along with any relevant information regarding the deleted comment.

3.5.4 Report

3.5.4.1 Export Excel

In the application, users can initiate a request to export patient test orders. This process involves generating an Excel file containing detailed information about the test orders.

When a user makes the export request, the output will be an Excel file that includes several important columns:

- ❖ Id Test Orders
- ❖ Patient Name
- ❖ Gender
- ❖ Date of Birth
- ❖ Phone Number
- ❖ Status
- ❖ Created By
- ❖ Created On
- ❖ Run By
- ❖ Run On

For the export to be successful, several acceptance criteria must be met. The test orders to be exported must not have been deleted, and they must exist in the system. If the list of test orders is empty, the system should default to exporting all test orders for the current month. Users can print all test orders regardless of their status, but if the status is not

"Completed," the columns for "Run By" and "Run On" will remain empty.

The system should follow a default naming convention for the Excel file, formatted as "Test Orders-Patient Name-Date Export." If a user specifies a name for the file, it must be a valid string without any special characters.

Prepared By 0.7 32/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Additionally, the export process should not block access to other features of the system and must include a notification regarding the progress of the export operation, ensuring users are informed while maintaining a smooth workflow.

3.5.4.2 Print Test Results

In the application, users can request to print a patient's test order results. When this request is made, the system generates a PDF file that includes two tables detailing the relevant information.

The first table in the PDF contains essential details about the test order, such as:

- ❖ Id Test Orders
- ❖ Patient Name
- ❖ Gender
- ❖ Date of Birth
- ❖ Phone Number
- ❖ Status
- ❖ Created By
- ❖ Created On
- ❖ Run By
- ❖ Run On

The second table includes the specific test results, and any comments associated with the patient's test order, providing a comprehensive overview of the patient's testing history.

For the process to be successful, several acceptance criteria must be met. The test order being printed must not have been deleted and must exist in the system. Additionally, it can only be printed if the status of the test order is marked as "Completed."

The default naming convention for the PDF file should follow the format "Detail-Patient Name-Date Print." If a user specifies a name for the file, it must be a valid string free from any special characters.

Moreover, the printing process must not block access to other features of the application, and users should receive notifications regarding the progress of the print job to keep them informed.

3.6 Instrument Service

3.6.1 Instrument Test Flow

3.6.1.1 Change Instrument Mode

In the application, users can change the **instrument mode** to manage its operational status: **Ready**, **Maintenance**, or **Inactive**.

If set to **Ready**, the instrument is available for **patient testing**. If set to **Maintenance**, the instrument is available for **quality control (QC)**, **quality assurance (QA)**, or repairs. If set to **Inactive**, the instrument is awaiting maintenance and is not operational.

Users must provide a **reason** for switching to **Maintenance** or **Inactive** modes, while switching to **Ready** requires

confirmation that the instrument has passed QC checks. The system will validate and log every mode change, including the user who performed the action, the timestamp, previous and new mode, and any reason provided.

This ensures an **audit trail** for accountability and helps users monitor instrument statuses on a **real-time dashboard**, enabling them to identify which instruments are available for use, undergoing maintenance, or inactive.

Prepared By 0.7 33/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

3.6.1.2 Blood Sample Analysis

In the application, lab users can initiate the analysis of blood samples by inputting them into the system. Each sample requires a valid test order and barcode to proceed. If a test order is missing but the barcode is valid, the system will automatically create a new test order and notify the user to update and match it to the correct patient later. Samples without a valid barcode will be skipped and logged to avoid errors.

The instrument must confirm that the reagent levels are sufficient before initiating the run. If reagents are insufficient, the system halts the process and notifies the user.

During processing, the system ensures test orders are traceable to the instrument performing the analysis. If the Test Order Service is unavailable, the run will still execute, and the system will sync and update test results once the service is restored. When the analysis is complete, **test results are converted to HL7 format** and published (Refer to Section 3.6.1.7).

Users receive **real-time notifications** for sample status updates, and the instrument's status automatically changes to **"Available"** after the workflow. For multiple cassettes and samples, the instrument seamlessly processes the next cassette in sequence. This ensures smooth operation, traceability, and efficient handling of blood samples.

3.6.1.3 Publish Test Results Using HL7

When the system finishes processing blood samples, **raw test results are converted into HL7 messages** and published to two services: **Test Order Service** and **Monitoring Service**.

The Test Order Service processes the HL7 message to match the results with the correct test order, converts it into a database-compatible format, and updates the record.

Meanwhile, the Monitoring Service stores the raw HL7 messages as a **backup to ensure data safety** in case of system issues. This process ensures accurate result recording and redundant storage for reliability.

Then, the raw test results are inserted into database at service as **another backup** and mark it **sent** if the message is published successfully or **fail** if the message is published failed.

3.6.1.4 Test Results Sync-up

When the service receives a sync-up request from the Monitoring Service, it checks the test results for the requested Test Orders. If test results are available, they are added to the response message; if no results are found, the response will be empty, and the Test Orders are marked as not having test results. This ensures the sync-up process reflects the status of the test results accurately.

Refer to section 3.2.3.2: Sync-up Raw Test Results.

3.6.1.5 Manual Delete Raw Test Results

In the application, users can **manually delete old raw test results directly from the instrument**. This capability is provided primarily to **release valuable memory space on the instrument itself**.

Prepared By 0.7 34/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Instruments, especially those in high throughput testing environments, have finite memory resources. Raw test results, while crucial initially, can accumulate rapidly. If left unchecked, this accumulation can lead to several problems:

- Performance Degradation: Slowing down the instrument's processing speed for new tests.
- Operational Stoppage: The instrument might eventually run out of memory, halting operations entirely and preventing new tests from being run.
- Data Overload: Making it cumbersome to navigate or manage results directly on the instrument interface. By allowing manual deletion, the system empowers operators to proactively manage instrument resources, ensuring sustained operational efficiency and preventing system slowdowns or failures due to memory constraints.

A crucial prerequisite for any deletion is ensuring that "the old raw test results are ready for deletion because they have been successfully sent and stored in the Monitoring Service." To prevent irreversible data loss. Raw test results are primary data so the user cannot delete them locally until a confirmed backup exists off-instrument. This directly leverages the established "Test Results Sync-up" flow (refer Section 3.6.1.4).

After successful deletion, the system **MUST** publish an event detailing for the audit trail:

- Who performed the deletion.
- Which Test Order/Barcode was deleted.
- When (Timestamp) the deletion occurred.

3.6.1.6 Auto Delete Raw Test Results

In the application, the system has a background job to verify and auto delete old raw test results to release the memory in the instrument. Ensures continuous, automated cleanup of instrument memory, preventing slowdowns or stoppages without manual intervention. This increases operational efficiency and instrument uptime.

The logic for identifying "old" results and the deletion process should align with, or reference, the existing "Manual Delete Raw Test Results" flow (Sec 3.6.1.5) for consistency and reuse of validated rules.

As same Manual Delete, which raw test results are deleted must have an event log and have same information for the audit trail.

3.6.2 Reagents Management

3.6.2.1 Install Reagents

In the application, users can install a new reagent by providing required details, including the reagent name, quantity, expiration date, **vendor information** (e.g., vendor name, vendor ID, contact), and other relevant metadata. Upon submission, the system validates the input to ensure all mandatory fields are complete, the expiration date is set to a **future date**, and the quantity is greater than zero. If validation passes, the system processes the request, confirms the successful installation, and **links the reagent to its vendor in the Reagent History**.

The system then logs the action in an **audit trail**, capturing the user's performed the installation, the timestamp, reagent details (e.g., lot number, expiration date), and **vendor information** (name, ID) to enable traceability. This data is also inserted into the **Reagent History** module, which maintains a permanent record of all installed reagents and their associated vendors. This ensures that users can trace which reagent was supplied by which vendor, including historical context for audits or compliance reviews.

Prepared By 0.7 35/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Before processing the installation request, the system must validate that all required fields are filled out correctly. The expiration date must be set to a future date to ensure the reagent is usable, and the amount provided cannot be zero, as that would indicate a lack of reagent for use. After installation, the reagent must be available for operational use within the system.

Refer to section 3.3.2.1 Reagents History for Vendor Supply.

3.6.2.2 Modify Reagents

In the application, users can update the status of a reagent. This process involves providing essential information, including the reagent name and whether the reagent is currently in use or not.

Once the user submits this information, the system processes the request and updates the reagent's status, accordingly, indicating whether it is now in use or not. Upon a successful update, the system also logs this activity, tracking who made the changes to the reagent status for auditing purposes.

It is important to ensure that the reagent has not been deleted previously and that it exists within the system. Additionally, if the reagent is already marked as "not in use," trying to update its status to "not in use" again will trigger an error. Similarly, attempting to mark a reagent as "in use" when it is already in that state will also result in an error message, preventing any conflicting statuses.

3.6.2.3 Delete Reagents

In the application, user can delete a reagent from the system. When a user decides to delete a reagent, the system processes this request and confirms that the reagent has been successfully removed.

Upon successful deletion, the system will log the action, recording which user performed the deletion and capturing the relevant information about the reagent. This logging feature is vital for maintaining an audit trail and ensuring accountability for changes made within the system.

For the deletion process to proceed, certain acceptance criteria must be met: the reagent must currently exist in the system, and it must not have been marked for deletion prior to the request. If these conditions are not satisfied, the system will prevent the deletion from occurring, maintaining the integrity of the reagent records.

3.6.3 Configuration Management

3.6.3.1 Sync Up Configuration

In the application, users can sync-up instrument configurations to ensure that each instrument is updated with the

correct settings. Configurations are categorized into two types: General Configurations and Specific Configurations. General Configurations apply universally to all instruments, providing basic operational parameters shared across all models.

Specific Configurations are tailored to the instrument's type and ensure they correctly match the instrument model (e.g., firmware version, calibration settings).

Prepared By 0.7 36/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

When a sync-up is initiated, the system identifies the instrument type and fetches the appropriate general and specific configurations. This guarantees the correct model-specific settings are applied, preventing operational mismatches. Users can perform this sync to update instruments with accurate configurations, ensuring they function optimally within their intended parameters.

If the specified instrument ID is not found, or its model/type cannot be determined, the system shall return an appropriate error message and halt the sync-up process.

If a configuration set (general or specific) is missing for a valid instrument, the system shall log the issue and notify the user/administrator.

If sync-up successfully, need to publish an event log to log who perform the sync-up features. This process ensures **compatibility**, **accuracy**, and minimizes errors by aligning configurations with the instrument's requirements.

4. Other Requirements

- Each service has its own database.
- All features need an API detail design.
- Have High Level Design.

5. Integration

- Other services need to contact IAM Service to identify users and privileges for running features. - Other services need to contact Monitoring to save the event logs to tracking information.

6. Non-functional Requirement

6.1 Reliability

The system must operate stably, maintaining accuracy and preventing critical failures to ensure secure storage and access to patient data.

It must remain stable when integrating new services, with automated monitoring and error-handling mechanisms to prevent disruptions.

Should maintain 99.9% uptime, with failover mechanisms ensuring continuous logging during system failures. Logs must be redundantly stored across multiple locations for backup.

The IAM system must be highly reliable, with redundant components and failover mechanisms to minimize downtime and support efficient recovery.

6.2 Scalability

Must efficiently scale to support increasing patients, medical records, and healthcare facilities without compromising performance.

Should enable flexible scalability to integrate new services seamlessly.

System architecture must support horizontal and vertical scaling to handle growing data and user demands.

Prepared By 0.7 37/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

The IAM system must scale to accommodate more users and devices, ensuring high performance for authentication requests.

6.3 Supportability

The system should allow for easy maintenance, upgrades, and expansions without disrupting hospital or healthcare centre operations.

6.4 Availability

The system should allow for easy maintenance, upgrades, and expansions without disrupting hospital or healthcare centre operations.

6.5 Performance

Must ensure fast processing and instant responsiveness for efficient access to medical data.

Should handle high log ingestion rates (thousands of events per second).

Must support scalable storage to accommodate growing log data.

Latency should be minimal to enable real-time tracking.

The IAM system should provide fast authentication and authorization responses, minimizing latency to ensure a seamless user experience. It should be able to process authentication requests in milliseconds and support thousands of concurrent users without performance degradation. Load balancing and caching mechanisms should be implemented to optimize performance during peak usage times.

6.6 Security & Privacy

Logs should be encrypted both at rest and in transit (e.g., AES-256 encryption).

Should comply with security standards like ISO 27001, HIPAA, PII, or GDPR.

Must adhere to high security standards to protect sensitive user data and prevent unauthorized access. Should implement strong encryption for data at rest and in transit, along with secure authentication protocols (e.g., Multi Factor Authentication).

Must support continuous monitoring, auditing, and alerting for real-time threat detection and response. Must comply with data protection regulations (e.g., GDPR, CCPA), ensuring privacy-enhancing features like data minimization, anonymization, and consent management.

6.7 Compatibility

New services must ensure compatibility with the existing platform without requiring major architectural changes.

6.8 Maintainability

Adding or updating services should be seamless without causing system downtime.

Prepared By 0.7 38/40 Last modified on 10/09/2025 23:38:00

Laboratory Management

Comprehensive documentation should be provided for integration processes, ensuring efficient deployment and maintenance by the technical team.

