Final Report
COSC 499
LIMS 0
Team Members: Keira Shepherd (Tech Lead), Sayo Everden (Project Manager), Shaya Selincourt (Client Liaison), Will McFarland (QA Lead)

Introduction

We are group 0 working on the A user-friendly Information Management System (LIMS) The client's target is to prepare a user-friendly LIMS which will have all the options a lab could imagine with different module systems.

In this report, we will summarize the requirements the client has presented to us, as well as the planning, designing, building, and testing we have done in creating our capstone project.

Description

A Laboratory Information Management System (LIMS) is an internal software program with multiple modules that manage laboratory workflows and information including but not limited to samples, test orders, and results. It aims to improve lab productivity and efficiency by keeping track of data associated with samples, experiments, laboratory workflows and instruments. It should be a plug and play software-based integrated LIMS to minimize the requirement of a software engineer in a lab facility. It will be used by lab workers as well as the clients who are requesting tests. The lab workers will enter information about samples, workflows, and lab information such as equipment and tests. The clients will request tests and receive results and status updates. Both groups will have to log in to the LIMS web portal to do anything within the system.

In the broader market, most LIMS offerings contain several faults: they are not modular, are expensive to maintain, and are often ugly and hard to use. Our LIMS implementation aims to make it easy for customer users to request a selection of tests and is able to handle the entire employee workflow for a test request being fulfilled.

User Groups

Clients

The clients are people who use the application to request tests. They will access the LIMS through a web browser, request tests, and send samples to the lab. They will also view the results of the tests and make payments through the application. The report will be generated for the clients. The sample is identified by account number and order number without personal identification. Clients can access their accounts only and their information is kept confidential.

Client functionality:

- Register
- Order test
- Track test status
- View result and report of their test

Lab Workers

Lab workers are the people working in a lab. They are the main users of the system and they will spend the most time using our application. Lab workers will access our application from the desktop. Lab employees have their own username and password to sign into the LIMS system and it will give you permission accordingly. They will add to and update the data in our application.

There are two different levels of authority among lab workers., technical staff and managers or heads of staff.

Technical staff

Technical staff handle lab processes following direction from LIMS and specification from each laboratory. Laboratory technical staff have access to a web-portal from the Employee page. They can view all open test orders (no personal details of clients), edit inventory status, view lab test processes and step, view and edit test order status,view and edit equipment schedule and view training docs.

Laboratory technical staff functionality:

- Receive order submission to generate an order number
- Receive samples, perfortm inspection and distribute inspected samples.
- Perform test and submit results - Have access to view all the orders, equipment, distribution, notification, schedule, testing, results, inventory, invoice and reports.

Lab Managers/Head technical staff and QC/ QA

Lab managers, head technical staff and QC/QA also have access to the admin dashboard. When you sign in with your username and password, if you have permission, it will direct you to the employee homepage. You can access the admin dashboard through the navigation bar of the employee homepage. With higher permission, they can approve/deny test orders, view all open test orders, view and edit inventory status, view and edit lab test processes and steps view and edit test order status, view and edit equipment schedule, view and edit training docs (edit only those specific to the laboratory), input procedures specific to lab, input pricing structure for lab, set up instruments and handle inventory orders.

Lab Managers/Head technical staff and QC/ QA functionality:

- Has access to all the functions as technical staff.
- Supervise, accept and deny tests and results.
- Set specifications for each laboratory and tests.

User Stories

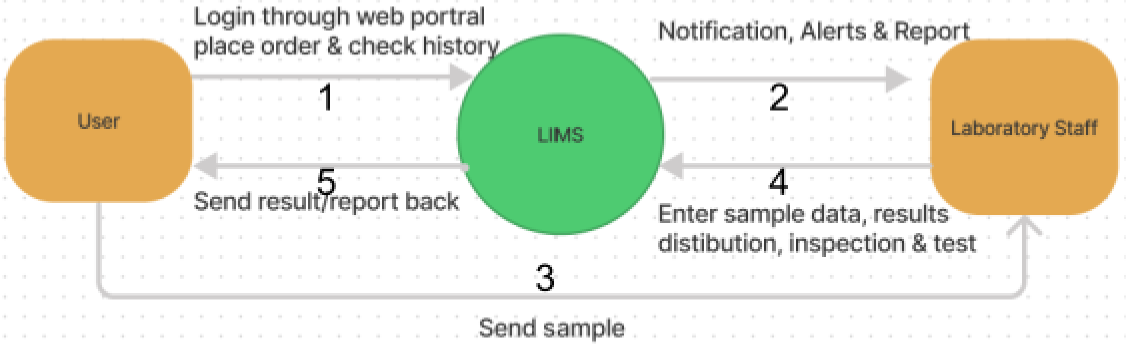
Client Successful Test Order

1. Client has samples they want to be analyzed and particular tests they want done
2. Client needs to complete a sample submission form specifying the quantity of the test and type of test (recommended package or customized test)
3. Order submission is accepted and your status on the website change to “ready to ship”
4. Mail or deliver samples to the lab
5. Lab receives sample
6. Conducts samples are processed
7. Tests are conducted by lab employees
    a. This may result in inventory updates
    b. Samples will be tracked through the barcode system
8. Employees enter test results into LIMS
9. Head technical staff approve the final result and proceed for reporting
10. Client uses the client portal to see test results

DFD Diagrams

Level 0

The level 0 data flow diagram models how the different user groups interact with the system. Our main user groups are Clients and technical staff. Each user has to register at the client's website in order to generate a unique account number. Then the user will send samples following instructions given by the laboratory. After quarantine and inspection, laboratory staff will enter data into the system. After testing and analysis, appropriate notifications, alerts, reports or invoice are sent to both clients and laboratory staff. Users and laboratory staff will interact with the system through a web-based portal.

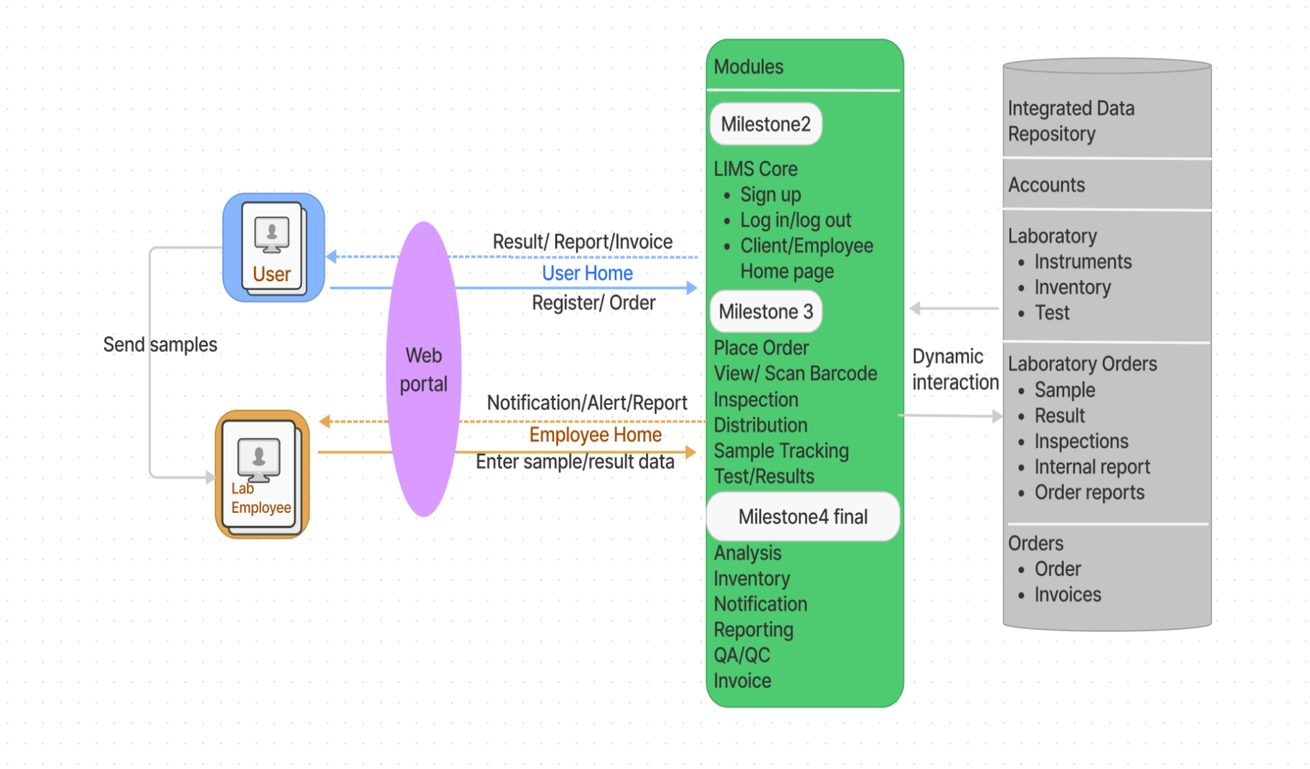

Login through web portral
place order & check history
1
User
LIMS
Notification, Alerts & Report
2
Laboratory Staff
5
Send result/report back
4
Enter sample data, results
distibution, inspection & test
3
Send sample

The level 0 data flow diagram models how the differen user groups interacr with the system

Level 1

The level 1 data flow diagram models how the different users interact with the system, and how the system internally communicates data.

Based on whether the account is a client or lab worker account, the system will redirect you to either the Client homepage or Employee homepage. Users and lab workers access LIMS through the web portal. Initial user data come from each user when they create an account and the order form provides sample data through a web portal. Lab employees enter data upon receiving and inspecting samples. The LIMS has features to distribute, schedule, test, analyze and track samples and it will be integrated to enable automation and collaboration between features. The software will interact with the database dynamically to enable real-time reports and invoices.



Functional Requirements

Our functional requirements for this project have changed significantly over the course of the year.

Original Function Requirements

Below are the functional requirements original proposed in September.

Core Components

This section of our requirements creates a base for our project. The objective is to be able to deliver correct and complete test results to the client. Including receiving, initial processing, analytical laboratory testing, and test result reporting and consolidation.
This means we need to set up our database. We need a login system for the customers and employees. The customers have access to their test results. The employees are able to add new tests and tasks.

Must-Have Components

The must-have components include a testing schedule, sample tracking/chain of custody, inventory control, general lab reporting, and quality assurance and control checkpoints.
For the testing schedule, we need to add to our database to account for the different parts of the tests. Test requests are processed in the order they are received (queue of tests). However, if

something is prioritized it changes to earlier in the queue depending on the level of prioritization of itself and the other tests in the queue.

There needs to be a way for the employees to view the testing queue.

For sample tracking, we have to update our database to account for the samples and possible locations of the sample. The employees must have a page for entering unique codes that brings them to the sample's page.

For inventory control, we need to update our database with inventory items. Employees should be able to see amounts of locations of inventory items, as well as be able to update the inventory.

For general lab reporting, reporting needs to be accessible by the customers. We need to allow for all types of reports. We will need to have somewhere to store the reports or the information to generate the report.

The final aspect of our second peer testing phase is the QA/QC management. We need to be able to add tasks for QC and QA. We also need to allow for control documents to be managed and completed within the LIMS.

Good-To-Have Components

Good-To-Have contents include a proactive testing schedule, statistical analysis, and billing for lab services.

The proactive test schedule allows the customer to generate a QR code, as well as print shipping labels, and instructions for the shipping of samples. The database needs to be able to locate the samples as "in transit" while they are being shipped. The tests schedule needs to take into account how long tests take and their start time, end time.

For the statistical analysis, we create an appropriate statistical analysis of the outputs, and reports needed internally.

The billing for lab services will collect and record billing revenue. There should be a page for customers to see their billing invoices. The employees should have a way to access customer billing invoices and information to change any invoices as needed.

Nice-To-Have Components

The training and education resources for the lab staff.

The stretch goal requires that we are able to track the laboratory personnel's education, certifications, and date of employment. There should be an internal training schedule and a way to track which employees have attended which training.

Updated Functional Requirements January 2022

In January, the client and the teams re-assessed our functional requirements and considered what would be the must-have functional requirements. The further details for these must-have requirements were given week by week.

Backend Work

Using the database structure shared by the 3 different LIMS teams, implement the database tables. Create a set of testing data, so that we have data to test functions with. Connect the database to the rest of our application so that it is usable, for reading and writing. The database structure will be determined from the details of the other steps.

Registration

A new user should be able to register as a client. The lab admin should be able to create new lab works and lab admin. This registration process should allow the newly created user all the correct permissions.
Client information required is

- Company Name
- Contact Person
- Contact Info (address and phone)
- Registration Date
- Username
- Password
- Account Number
- Email

Lab workers and admin require

- Username
- Password
- Job Title
- Email
- Name

Login

Any user should be able to login and the login state should be maintained as the user navigates the various pages.

Order Submission

Clients should have access to Shopping Cart, Results, and Order History.
A logged-in client can submit an order of tests. Clients should be able to select samples with quantity and the specific test or package they desire. The system will calculate the price. The system will generate a shipping label and barcodes after the order has been placed.
An order number will be generated for each purchase client. A client will have one account number and multiple orders. Each order will have a display order code that is (account_number)-(order_number) for example, 4-10.
Clients can also track their samples. They will see the status of the sample and its related order code. Status of the sample can be

- Ready to Ship
- Received

- In Progress
- Result
- Adverse Event

Sample Received and ID

Items related to the database will be accessible through a barcode system. The users of the system will scan a barcode and it will allow them easy access to the items in the database, such as samples.
Sample codes are (order-number)-(sample-id) e.g. 4-101
Distribution labeling codes are as follows (order-number)-(sample-id)-(lab-id) e.g. 4-101-M
Test sample labeling codes are as follows (order-number)-(sample-id)-(lab-id)-(test-id) e.g. 4-101-M-1
These are the identification codes that should be seen by users.

Inspection

For each order received by the lab, you should have information displayed about the order including company name, contact person, email, address, phone number.
For each order you will need to inspect the order and its samples for package integrity, quantity received, and material integrity, then the user determines if the inspection result is pass or fail.
Each inspection is related to a lab worker or admin.
This should be saved for later re-viewing.

Distribution

After samples pass inspection, they can be distributed to a lab or placed in storage. We should store the location of the samples in the database, and have it be changed based on where a sample is placed after distribution. The location will be storage, or a lab (e.g. Analytical Lab).

Inventory

Items in the inventory will be sorted into tabs and subtabs.

- Chemicals
  - Solvents
  - Organic Compounds
  - Inorganic Compounds
  - Standards
- Buffers and Media
- Consumables
- Raw Materials
- Products Tracking

This displays the information about the inventory items. You will be able to search using a search bar for different items in the inventory.
The information needed for each time in inventory is

- Recorder

- Catalog No.
- Date of Last Order
- Item Name
- Vendor
- Purity
- Stock Location
- Unit
- Cost per Unit
- Stock quantity (ml)
- Total Value
- Recorder Level (ml)
- Item Recorder Quantity (L)
- Item Discontinued?
- Notes

There will also be a manufactured product inventory, which will have

- Product Inventory number
- Date purchased/collected
- Vendor
- Vendor License Information
- Original Stock amount
- Remaining stock
- Storage location

We will also need a place to view activity on the manufactured product

- Project Number
- Inventory Number
- Contacts
- Vendor
- Room Taken From
- Room Placed In
- Personnel Moving
- Actitivty Description
- Date
- Number of Manufactured
- Number of Products Destroyed
- Date of Destruction
- Remaining Product Number
- Total Remaining

Based on the dates, inventory will be displayed in a table with the following

- Month
- Opening Inventory
- Total Addition
- Reduction
- Adjustment
- Closing Inventory

There are a series of workflows related to the inventory that will also need to be implemented.

Workflow 1: Tracking Inventory
Ability to track the use of consumables and laboratory supplies by scanning applied codes prior to use.
Ability to count inventory and monitor use.
Ability to assign codes to consumables and laboratory supplies.

Workflow 2: Inventory Status & Requisition
Ability to monitor consumable and laboratory supplies and set ordering schedules and record delivery dates and inventory quantities required.
Ability to alert appropriate users on quantity status.
Ability to generate order times with pre-entered delivery times.
Ability to log when consumables are being used during testing, monitor inventory levels and anticipate usage.
Ability for users to request consumables/lab supplies for non-standardized testing/one-off experiments that do not run routinely.
Ability for users to request media/buffers/solutions that are made internally.

Workflow 3: Inventory Fulfilment & Replenishment
Ability to create codes and labels for requested items with information such as a user that requested the item and the user that approved it as well as the technician that made it.
Ability to block orders in certain cases. Administrative permission.

Analysis

Analysis will be split by lab. Each lab will have one page. For each sample in that lab you will need to display the following

- Sample ID
- Reference SOP #
- Lab Personnel
- Test Code
- Results
- Out of Specification
- Pass/Fail
- Report Generation

Reports

There are 3 different types of report
Each lab will have a report, the whole company will have a grouped report, and finally an invoice report for the clients.
The lab reports should have

- Sample ID
- Sample Type

- Test Code
- Date Received
- Date Published
- Lab Personnel
- Results
    - Test Number
    - Limit
    - Results
    - Pass/Fail
- Approved By

The lab-specific reports will be approved by the head of that lab. The company lab report will be signed by the head of lab operations.
The are no specific invoice requirements.

Notifications

There should be a notification sent by email for

- notification on tests requests by submitter and timeframe
- notification of overdue samples that have been requisitioned but have not
- arrived at the facility.
- notification on test requests.
- notification on samples received in any given time frame in.
- notification on sample types received.
- notification on test requests that have been mapped
- notification on which samples/tests need to be prioritized.
- notification on incomplete/skeleton records.
- notification on tracking information on samples.
- notification on quarantine status of samples.
- notification on sample locations.
- notification on user activities pertaining to tracking of samples.
- notification on notifications sent to client.
- notification on end of day tally of the number of samples in quarantine
- separated by sample type.
- notification on inspection reports, failures and releases.
- notification on additional testing requirements and subsequent testing by
- source test and reason.
- notification on failed tests due to run error.
- notification of list of samples for retesting and reason.
- notification on user activates.
- notification on instrumentation activities.
- notification on sample status
- notification on technician activates.
- notification on individual sample results.
- notification on raw batch data.
- notification on data analysis.

- notification on batch control ranges.
- notification on batch QC failures and failure codes.
- Control Documents for QC failures and failure codes.
- Control Documents for audit trail.
- notification on validation edits.
- notification on over-ride events.
- notification on test results and corresponding specification and regulatory
- requirements.
- notification of samples that do not meet specifications.
- notification of test results that are in queue to be reviewed.
- notification of test results that have been reviewed.
- notification of passed samples.
- notification of failed samples.
- notification of samples that are being retested.
- notification of rerun batches.
- notification of additional tests associated with the sample that were not
- initially requested.
- notification on sample retest and original test results.
- notification on batch rerun and original batch errors.
- notification of test processing time by priority.
- Test status reports.
- notification on a given sample's location from receipt to disposition by
- location and date.
- notification on rejected sample sending to the submitter.
- notification on the contents of a given refrigerator/freezer on any given date.
- notification on inventory at any given time.
- notification on replenishment requirements and orders placed.
- notification on expiration dates and upcoming expiration dates.

Permissions

Barcodes should only work for users who have permission to access the item related to the barcode. There are different types of Lab Admin and Workers who should have different permissions

- Lab Director: Authorised for all designations and activates and can sign off on reports and projects pertaining to all activities as listed below.
- Quality Control Officer: Authorised for all designations and activities and must be a secondary signatory on all control documents, sample failures, re-runs, quarantine/unquarantined procedures, deviation reports, out of specification reports, vendor qualification, instrument validation and all activities pertaining to quality management systems.
- Quality Person in Charge/Alternate: Authorised for all designations and activities pertaining to controlled substances. These include accepting samples, quarantining and releasing and destruction destinations.

- Head of Chemistry/Molecular: Authorized for all designations pertaining to lab activities. These include matters related to equipment, selection and verification of methods, validation of methods, sampling, test calibration, technical records, evaluation of measurements of uncertainty, validity of results, reporting, statements of conformity, failing tests, out of specification reporting, deviation reporting, etc.
- Technicians: Authorized for designations pertaining to lab activities depending on which division and what level of responsibility. These include, but are not limited to, running tests, checking specifications, checking storage and environmental controls. Technicians do not generally have authority on QC matters, validating reports, calibrating, conformity tests, out of specification reporting, deviations and authorizing reports.

Equipement Integration

All laboratory equipment needs to be integrated into the system to write the data directly to the LIMS.

Final Functional Requirements

As more details were given about each requirement over the second term, we had to re-access what was reasonable to expect for each requirement. Below are the final requirements.

Backend Work

Using the shared database structure, implement the database tables. Create a set of testing data, so that we have data to test functions with. Connect the database to the rest of our application so that it is usable, for reading and writing. The database structure will be determined from the details of the other steps.

Registration

A new user should be able to register as a client. The lab admin should be able to create new lab works and lab admin. This registration process should allow the newly created user all the correct permissions.
Client information required is

- Company Name
- Contact Person
- Contact Info (address and phone)
- Registration Date
- Username
- Password
- Account Number
- Email

Lab workers and admin require

- Username

- Password
- Job Title
- Email
- Name

Login

Any user should be able to login and the login state should be maintained as the user navigates the various pages.

Order Submission

Clients should have access to Shopping Cart, Results, and Order History.
A logged-in client can submit an order of tests. Clients should be able to select samples with quantity and the specific test or package they desire. The system will calculate the price. The system will generate a shipping label and barcodes after the order has been placed.
An order number will be generated for each purchase client. A client will have one account number and multiple orders. Each order will have a display order code that is (account_number)-(order_number) for example, 4-10.
Clients can also track their samples. They will see the status of the sample and its related order code. Status of the sample can be

- Ready to Ship
- Received
- In Progress
- Result
- Adverse Event

Sample Received and ID

Items related to the database will be accessible through a barcode system. The users of the system will scan a barcode and it will allow them easy access to the items in the database, such as samples.
Sample codes are (order-number)-(sample-id) e.g. 4-101
Distribution labelling codes are as follows (order-number)-(sample-id)-(lab-id) e.g. 4-101-M
Test sample labelling codes are as follows (order-number)-(sample-id)-(lab-id)-(test-id) e.g. 4-101-M-1
These are the identification codes that should be seen by users.

Inspection

For each order received by the lab, you should have information displayed about the order including company name, contact person, email, address, phone number.
For each order you will need to inspect the order and its samples for package integrity, quantity received, and material integrity, then the user determines if the inspection result is pass or fail.
Each inspection is related to a lab worker or admin.
This should be saved for later re-viewing.

Distribution

After samples pass inspection, they can be distributed to a lab or placed in storage. We should store the location of the samples in the database, and have it be changed based on where a sample is placed after distribution. The location will be storage, or a lab (e.g. Analytical Lab).

Inventory

Items in the inventory will be sorted into tabs and subtabs. Only the chemicals tab and subtabs need to be implemented.

- Chemicals
  - Solvents
  - Organic Compounds
  - Inorganic Compounds
  - Standards
- Buffers and Media
- Consumables
- Raw Materials
- Products Tracking

This displays the information about the inventory items. You will be able to search using a search bar for different items in the inventory.
The information needed for each time in inventory is

- Recorder
- Catalog No.
- Date of Last Order
- Item Name
- Vendor
- Purity
- Stock Location
- Unit
- Cost per Unit
- Stock quantity (ml)
- Total Value
- Recorder Level (ml)
- Item Recorder Quantity (L)
- Item Discontinued?
- Notes

Analysis

Analysis will be split by lab. Each lab will have one page. For each sample in that lab you will need to display the following

- Sample ID
- Reference SOP #
- Lab Personnel

- Test Code
- Results
- Out of Specification
- Pass/Fail
- Report Generation

Reports

There are 3 different types of reports
Each lab will have a report, the whole company will have a grouped report, and finally an invoice report for the clients.
The lab reports should have

- Sample ID
- Sample Type
- Test Code
- Date Received
- Date Published
- Lab Personnel
- Results
    - Test Number
    - Limit
    - Results
    - Pass/Fail
- Approved By

The lab-specific reports will be approved by the head of that lab. The company lab report will be signed by the head of lab operations.
The are no specific invoice requirements.

Notifications

From the list below, implement all the notifications for which we have already completed the related task.
There should be a notification sent by email for

- notification on tests requests by the submitter and timeframe
- notification of overdue samples that have been requisitioned but have not
- arrived at the facility.
- notification on test requests.
- notification on samples received in any given time frame in.
- notification on sample types received.
- notification on test requests that have been mapped
- notification on which samples/tests need to be prioritized.
- notification on incomplete/skeleton records.
- notification on tracking information on samples.
- notification on quarantine status of samples.
- notification on sample locations.

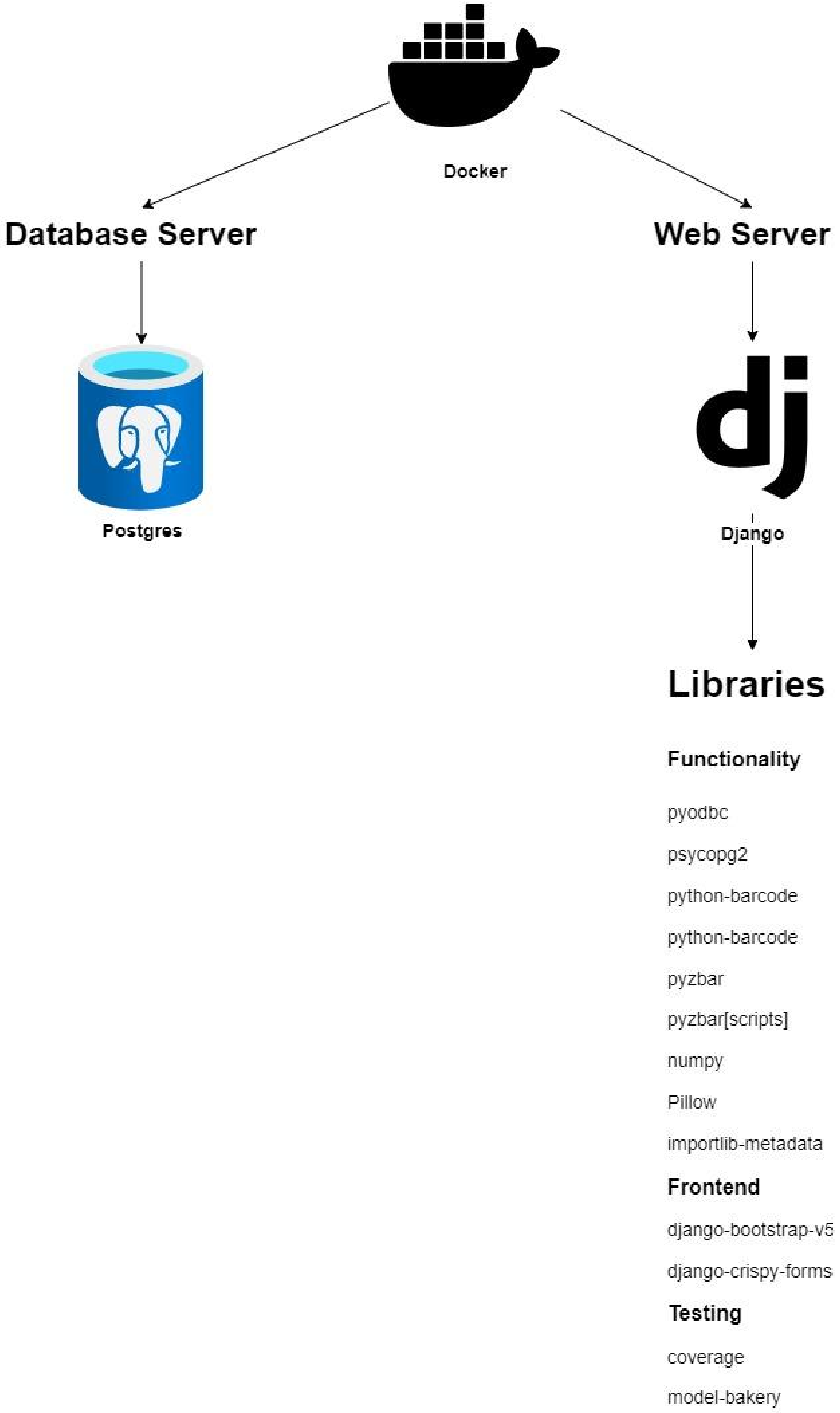
- notification on user activities pertaining to tracking of samples.
- notification on notifications sent to client.
- notification on end of day tally of the number of samples in quarantine
- separated by sample type.
- notification on inspection reports, failures and releases.
- notification on additional testing requirements and subsequent testing by
- source test and reason.
- notification on failed tests due to run error.
- notification of list of samples for retesting and reason.
- notification on user activates.
- notification on instrumentation activities.
- notification on sample status
- notification on technician activates.
- notification on individual sample results.
- notification on raw batch data.
- notification on data analysis.
- notification on batch control ranges.
- notification on batch QC failures and failure codes.
- Control Documents for QC failures and failure codes.
- Control Documents for audit trail.
- notification on validation edits.
- notification on over-ride events.
- notification on test results and corresponding specification and regulatory
- requirements.
- notification of samples that do not meet specifications.
- notification of test results that are in queue to be reviewed.
- notification of test results that have been reviewed.
- notification of passed samples.
- notification of failed samples.
- notification of samples that are being retested.
- notification of rerun batches.
- notification of additional tests associated with the sample that were not
- initially requested.
- notification on sample retest and original test results.
- notification on batch rerun and original batch errors.
- notification of test processing time by priority.
- Test status reports.
- notification on a given sample’s location from receipt to disposition by
- location and date.
- notification on rejected sample sending to the submitter.
- notification on the contents of a given refrigerator/freezer on any given date.
- notification on inventory at any given time.
- notification on replenishment requirements and orders placed.
- notification on expiration dates and upcoming expiration dates.

Permissions

Barcodes should only work for users who have permission to access the item related to the barcode. There are different types of Lab Admin and Workers who should have different permissions

- Lab Director: Authorized for all designations and activates and can sign off on reports and projects pertaining to all activities as listed below.
- Quality Control Officer: Authorized for all designations and activities and must be a secondary signatory on all control documents, sample failures, re-runs, quarantine/unquarantined procedures, deviation reports, out of specification reports, vendor qualification, instrument validation and all activities pertaining to quality management systems.
- Quality Person in Charge/Alternate: Authorized for all designations and activities pertaining to controlled substances. These include accepting samples, quarantining and releasing and destruction destinations.
- Head of Chemistry/Molecular: Authorized for all designations pertaining to lab activities. These include matters related to equipment, selection and verification of methods, validation of methods, sampling, test calibration, technical records, evaluation of measurements of uncertainty, validity of results, reporting, statements of conformity, failing tests, out of specification reporting, deviation reporting, etc.
- Technicians: Authorized for designations pertaining to lab activities depending on which division and what level of responsibility. These include, but are not limited to, running tests, checking specifications, checking storage and environmental controls. Technicians do not generally have authority on QC matters, validating reports, calibrating, conformity tests, out of specification reporting, deviations and authorizing reports.

Technical Specifications



Docker

Database Server

Postgres

Web Server

Django

Libraries

Functionality

pyodbc

psycopg2

python-barcode

python-barcode

pyzbar

pyzbar[scripts]

numpy

Pillow

importlib-metadata

Frontend

django-bootstrap-v5

django-crispy-forms

Testing

coverage

model-bakery

Test Report

During the second round of peer testing, some issues were discovered with our project. The two that we deemed to be the highest priority based on the frequency of feedback from users and the significance of the impact on the user's experience were:

1. Users often got stuck due to not understanding the technical terminology being used in our project
2. Users had a hard time finding an order they had just placed as a client

The first issue did not result in any changes on our part as the users for this project will likely be trained lab technicians who are familiar with the terminology, so this would be a lower priority issue in reality that would be resolved through the training of users.

The second issue was addressed by restructuring the client order page by sorting orders from newest to oldest and providing more information about each order.

In general, our peer testing participants found the styling in our project to be barebones and inconsistent in some areas. This was addressed by adding consistent and thoughtful styling throughout the project.

Unit tests were written for each module in our project. In total, 70 unit tests were written and passed in the final build of our project. This resulted in 88% overall test coverage with the areas lacking coverage primarily being code related to input forms, which are more difficult to test automatically. Many tests included integration between components and model fixtures that simulate real data. Our unit tests ensure that views make use of the correct HTML templates, permission structures are valid, and data model generation and modification processes are correct. In addition to running the unit tests, manual testing of typical and relevant processes was also conducted after each pull request to ensure no other components had been affected by recent changes. A testing guide was written for documentation purposes, and to ensure consistency of tests written by other members of our team and future tests that may be written by our client.

Unimplemented Requirements

Below are the requirements that were not met or were only partially implemented. All systems are in place so that these changes are possible.

Backend Work

Some database changes may be needed as new features are implemented.

Registration

Some changes may be needed as further permissions are implemented.

Order Submission

As it is currently implemented, our order system is quite rudimentary and may need to be redesigned in the future. It currently only allows for the order of one type of test per sample type, does not generate a price, and does not have a ‘shopping cart’ feature that would allow the user to add and edit items in their cart before ordering.

Sample Received and ID

Barcodes are only implemented for samples and will need to be implemented for the other items in the database.

Analysis

The analysis page at this time is only one page with subpages for each lab. The client originally requested that each of these subpages be a tab on the main analysis page.

Reports

The database is set up to allow for reports, however, the reporting functionality has not been implemented.

Notifications

Only event-based notifications for implemented modules have been created. Recurring notifications as well as notifications for modules we did not implement have not been created.

Permissions

The system permissions at this time only consider 3 user groups, and not the specific job titles within the lab workers or lab admin.

Miscellaneous

Originally, we had intended to create a custom color scheme and unique components for the UI to give it a unique feel. However, we didn’t have time to do this, and are using mostly Bootstrap 5 defaults. Furthermore, some elements can be modified to improve the user experience, such as the inspection validity indicators (valid, invalid) when looking at sample information.

Bugs

We are not aware of any bugs in the current build of our project.
Peer testing revealed some bugs and other issues that we addressed in the weeks following the sessions. Peer testing also led us to re-evaluate certain aspects of the design of our website and in many cases, we made changes in response to users’ feedback. In recent demos and testing, the primary source of error has been the result of incorrect or inconsistent setup and

environment, so effort has been put into creating an effective readme and first-time setup procedure for users setting up and running our project for the first time.

Project Guide

Everything you need to know about the project can be found in the project’s readme: https://github.com/shepkeira/LIMS/blob/main/README.md. This information is summarized below and includes additional information such as the environment variables that were not published in our repository.

Initial Repo Set Up

Our code can be found on our GitHub: https://github.com/shepkeira/LIMS
Our project uses docker and thus you will not need to install much else. You will need to have docker and docker-compose installed on your device. You will also need to install python version 3.7 on your device.
Each user will need to generate their own secret key using the following command

```
python -c "import secrets; print(secrets.token_urlsafe())"
```

Next you will need to update your .env file. The .env file should be in the LIMS_IMAGE folder. This file holds various system variables.
The .env file should look as follows:

```
SECRET_KEY=fake_key
DJANGO_SETTINGS_MODULE=src.settings
EMAIL_PASSWORD=email_app_password

MSSQL_HOST=fake_mssql_host_name
DB_USER=fake_user
DB_PASSWORD=fake_password

POSTGRES_DB=fake_postgres_db_name
POSTGRES_HOST=fake_postgres_host_name
POSTGRES_USER=fake_user
POSTGRES_PASSWORD=fake_password
```

Replace the `fake_key` with the key you just generated. The rest of the variables are as follows

```
DJANGO_SETTINGS_MODULE=src.settings
EMAIL_PASSWORD=tkotnqcnlgknzfkj

MSSQL_HOST=lims_db_server_mssql
DB_USER=root
DB_PASSWORD=Lab_rats2021

POSTGRES_DB=lims_db
```

```
POSTGRES_HOST=lims_db_server_postgres
POSTGRES_USER=root
POSTGRES_PASSWORD=Lab_rats2021
```

The database and email accounts can be replaced as necessary.
Email settings are defined in `settings.py`. To change the email address, change the `EMAIL_HOST_USER` and `EMAIL_HOST_PASSWORD` values (and other values if changing to a provider other than Gmail). The email host password is defined in the .env file. To change to a new Gmail account, ensure to use an app password and *not* the google account password.

Running the Project the First Time

To run the project locally, you will need to run the following command in the LIMS_IMAGE folder:

```
docker-compose up --build db-postgres
```

This will build up the Postgres database when you get the “database is ready to accept connections” you can CTRL+C to bring down the database.

Next run,

```
docker-compose up --build
```

Running the Project The Nth Time

Since the database and project docker instances are built we only need to start them up again with

```
docker-compose up
```

Installation and Start-Up Bugs

There is a known bug with the database startup. Because your database changes are stored locally, sometimes the database docker instance will not start up properly when switching branches. When this happens you can rebuild the database by first deleting all the contents of the `LIMS_IMAGE/db-postgres/data` folder. Then run this command

```
docker-compose up --build db-postgres
```

This will rebuild the database, using the data stored in `datadump.json`

Important Processes

There are a few other processes that are important to know

Creating New Super Users

Superusers are able to do all actions and are not restricted by user permissions. To create superusers start up the docker instances locally and then follow the below commands:

```
sudo docker exec -it lims_web_server python manage.py migrate
sudo docker exec -it lims_web_server python manage.py createsuperuser
```

Then follow the on-screen prompts to finish creating the user with the correct attributes.

Accessing the Admin Site

The admin site allows superusers to modify the database directly. This can be accessed by logging in as a superuser and clicking “Administrator Dashboard” which will redirect you to `localhost:8000/admin` which is the admin site.

Making Migrations

When you want to change something about the database through the models you will need to make migrations.

1. Start up the docker instance with `docker-compose up`
2. Make the change to the models
3. `docker exec -it lims_web_server bash`
4. `python manage.py makemigrations`
5. `python manage.py migrate`
6. Back up the changes to the datadump.json file within the LIMS_IMAGE folder `docker exec -it lims_web_server python manage.py dumpdata -o datadump.json`
7. You can now close the docker instances

Backing Up the Database

After making changes to the content of the database. For example, making a new superuser. You will need to back up the database. To create a database backup, run the following command while the container is running in the LIMS_IMAGE folder

```
docker exec -it lims_web_server python manage.py dumpdata -o datadump.json
```

Running the Tests

Tests can be executed with `docker exec -it lims_web_server python manage.py test tests/` while the docker image is running.

To use the coverage library, access a bash terminal in the container and run coverage run manage.py test tests/. Export the results by running `coverage html`. You can also use `coverage report` to see the results in the terminal.

Conclusion

Working with the gap between our limited capacity and the scope of this capstone project, we have built a system that can be passed on for further development by another team. Over the course of the year, we managed to successfully implement most of the required features, developed a solid database system, and implemented a great deal of testing that aided us in iterating and improving our project. We are satisfied by our choice of tech stack as Django proved to be a comfortable environment to work in, and Docker helped to limit environmental problems faced in development and transfer of ownership. The issues and unimplemented features in our project are well documented, and our code is designed such that these changes can be easily implemented.

Links

A link to your github repo you used to build the project

A link to your promo video

A link to your presentation video