Implementing Laboratory Information Management System (LIMS) at National Control Laboratory(NCL), DGDA

by

Sheikh Zamiur Rahman
18101581

An internship report submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
April 2024

Declaration

It is hereby declared that

1. The report submitted is my own original work while completing degree at Brac University.
2. The report does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The report does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I have acknowledged all main sources of help.

Student's Full Name & Signature:

Sheikh Zamiur Rahman
18101581

Approval

The report titled "Implementing Laboratory Information Management System(LIMS) at National Control Laboratory(NCL), DGDA" presented by Sheikh Zamiur Rahman (ID: 18101581) has been deemed satisfactory in meeting the requirements for the degree of B.Sc. in Computer Science and Engineering.

Examining Committee:

Industrial Supervisor:

Tanvir Khan Topu
Software Engineer
Orange Solutions Ltd.

Academic Supervisor:
(Member)

Arif Shakil
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

This internship report details the development of a Laboratory Information Management System (LIMS) for the National Control Laboratory (NCL) of Bangladesh. LIMS software facilitates the organization and management of laboratory data, samples, and procedures. By implementing a LIMS, NCL can streamline daily workflows, enhance data security and accuracy, and significantly reduce manual errors.
NCL, under the Directorate General of Drug Administration (DGDA), plays a crucial role in ensuring the quality, safety, and efficacy of drugs available in Bangladesh. They oversee the import, manufacture, and distribution of all medications throughout the country. Currently, NCL relies on manual processes for data entry, report generation, and maintaining an audit trail. This paper describes the development of a LIMS in collaboration with supervisor Tanvir Khan Topu, aiming to automate these tasks and create a centralized data repository.
The new LIMS offers significant advantages. It automates data entry and report generation, eliminating the risk of human error and saving valuable time. Additionally, it provides a centralized platform for storing and retrieving laboratory data, ensuring improved accessibility and data integrity.

Keywords: LIMS; NCL; DGDA

Acknowledgement

In the name of Allah, the Entirely Merciful the Especially Merciful. All praise is due to Allah and His help for allowing me to complete my internship and this report.
I would like to express my sincere gratitude to my supervisor, Tanvir Khan Topu, for his invaluable guidance, mentorship, and encouragement throughout this internship. His expertise in LIMS development was instrumental in shaping my understanding and skills during this project. His constructive feedback consistently helped me improve my work.
I am also grateful to Arif Shakil sir for his ongoing support throughout the year. His assistance with submissions and answering my reporting questions proved invaluable.
My sincere thanks to Orange Solutions Ltd. for providing me with the opportunity to intern at the National Control Laboratory (NCL) and work on this impactful project.
Finally, I extend my heartfelt gratitude to my family and friends for their unwavering support, encouragement, and understanding during this entire process. Their unwavering belief in me helped me persevere through challenges and achieve my goals.journey.

Table of Contents

Declaration i

Approval ii

Abstract iii

Acknowledgment iv

Table of Contents v

List of Figures vii

Nomenclature viii

1 Introduction 1
- 1.1 Problem Statement . . . 1
- 1.2 Objectives . . . 2
- 1.3 Methodology . . . 2
  - 1.3.1 Primary Data . . . 2
  - 1.3.2 Secondary Data . . . 2

2 Company, Regulatory Bodies, and Laboratory Processes 3
- 2.1 Company Profile . . . 3
  - 2.1.1 Provided Products . . . 4
  - 2.1.2 Technology and Framework . . . 4
  - 2.1.3 Software . . . 4
- 2.2 About DGDA and NCL . . . 5
  - 2.2.1 DGDA . . . 5
  - 2.2.2 NCL . . . 5
  - 2.2.3 Brief Process Description of NCL . . . 6
  - 2.2.4 Drug Wing Flow Diagram . . . 7
  - 2.2.5 Vaccine Wing Flow Diagram . . . 8
  - 2.2.6 OOS Form Generation Flow Diagram . . . 9

3 My Learning 11
- 3.1 ZAB Framework . . . 11
  - 3.1.1 Extension: [.column] . . . 12
  - 3.1.2 Extension: [.def] . . . 12
  - 3.1.3 Extension: [.menu] . . . 13
  - 3.1.4 Extension: [.screen] . . . 13

3.2 ZABadmin Panel 15
3.3 Crystal Report 16
3.4 MS SQL 17

4 My Contribution **20**
4.1 First Phase: Laying the Foundation 20
4.1.1 Beyond the Core: Expanding Functionality 21
4.1.2 Collaboration and Quality Assurance 21
4.2 Second Phase: Streamlining Reporting with Automation 21
4.3 Third Phase: Deployment and Beyond 23
4.4 Challenges and Overcoming Obstacles 23

5 Conclusion **24**

Bibliography **25**

List of Figures

2.1 Industrial Footprints of OSL 3
2.2 DGDA logo 5
2.3 NCL, Sample Receive 6
2.4 NCL, Drug Wing Flow Diagram 7
2.5 NCL, Vaccine Wing Flow Diagram 8
2.6 NCL, OOS Flow Diagram 9
2.7 NCL, All Modules 10

3.1 Columns of Tables 12
3.2 Creating a Table 12
3.3 Creating a Menu 13
3.4 Initialization 13
3.5 Modification of Fields/Columns 14
3.6 Closing a Screen 15
3.7 Source Master Screen 15
3.8 ZABadmin Panel 15
3.9 NCL, Sample Receive Report Design and Result 16
3.10 Process Flow Procedure 17
3.11 Data-table Building Procedure 18
3.12 NCL, Sample Receive Report Procedure 19

4.1 NCL, Master Screens 20
4.2 Sample Barcode 21
4.3 Final Report 22

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI Artificial Intelligence

$CAPA$ Corrective and Prevention Activities

DC Deputy Chief

$DGDA$ Directorate General of Drug Administration

ERP Enterprise Resource Planning

GA Government Analyst

$HTML$ Hyper-Text Markup Language

IoT Internet of Things

$LIMS$ Laboratory Information Management System

NBR National Board of Revenue

NCL National Control Laboratory

OAT Operational Acceptance Test

OOS Out of Specification

OSL Orange Solutions Ltd.

PHP Hypertext Preprocessor

QMS Quality Management System

SRS Software Requirement Specification

UAT User Acceptance Test

UG United Group

$UNOPS$ United Nations Office for Project Services

Chapter 1

Introduction

The Laboratory Information Management System (LIMS) [3] is an indispensable software for automating business processes, namely in drug and vaccine testing laboratories worldwide.
This project, sponsored by the United Nations Office for Project Services (UNOPS), holds significant value for the Bangladeshi government by streamlining daily processes for drug, chemical, and vaccine testing and administration. Since the initial report, we have implemented substantial improvements at NCL. The project is on track for completion and operational use by June 31st, with ongoing efforts focused on ensuring a successful implementation at the NCL site.
Our primary objective, aligned with the sponsor's expectations, is to fully automate workflows at NCL and eliminate the need for handwritten documentation. We are making significant progress towards this goal, having fulfilled nearly all user requirements at NCL.

1.1 Problem Statement

Established in 2016, the National Control Laboratory (NCL), under the Directorate General of Drug Administration (DGDA), plays a critical role in ensuring public health by testing and regulating the quality, safety, and efficacy of drugs, vaccines, and other medicines entering Bangladesh. NCL processes a significant volume, averaging 600-700 samples monthly.
However, NCL currently relies on manual data management practices using logbooks and registers. This traditional approach presents several challenges:

- **Inefficiency:** Data entry is a time-consuming process, hindering laboratory workflow.
- **Data Accessibility:** Retrieving, verifying, and archiving data requires manual sorting through extensive paper records, hindering efficiency and accuracy.
- **Data Security:** Handwritten records are vulnerable to loss or damage due to unforeseen events like fire or water damage.
- **Lack of Centralization:** Manual data management hinders the creation of a centralized data repository, limiting data accessibility and analysis capabilities.

These limitations prevent NCL from achieving optimal efficiency and data integrity in its critical role of safeguarding public health.

1.2 Objectives

The limitations of NCL's current manual data management system necessitate the implementation of a Laboratory Information Management System (LIMS). This project aims to deliver a comprehensive software solution that fully automates NCL's laboratory processes and business operations. By automating these workflows, the LIMS will significantly improve efficiency, accuracy, and productivity:

- **Automated Data Entry:** Eliminate time-consuming manual data entry, streamlining workflows and minimizing errors.
- **Streamlined Report Generation:** Automate report generation, enabling faster and more efficient data analysis and communication.
- **Comprehensive Audit Trail Maintenance:** Maintain a secure and detailed audit trail for all actions performed within the system, ensuring data integrity and regulatory compliance.
- **Centralized Archiving:** Implement a centralized archiving system for all laboratory data, ensuring secure and accessible data storage for future reference and analysis.

These objectives directly address the challenges associated with NCL's current manual system and pave the way for a more efficient, accurate, and data-driven approach to laboratory operations.

1.3 Methodology

This report is primarily based on my internship experience at Orange Solutions Ltd. (OSL), supplemented by information from relevant sources.

1.3.1 Primary Data

- **Personal Experience**: Firsthand knowledge gained during project development.
- **Team Meetings**: Collaborative discussions and planning sessions with the development team, business analysts, quality testers and project manager.
- **Visits to NCL**: Direct observation of NCL's workflow and user needs.
- **Communicating with Users**: Gathering user requirements and feedback through meetings and interactions.

1.3.2 Secondary Data

- **OSL Website**: Information about OSL's expertise and relevant services.
- **Internet**: Research conducted online to supplement project knowledge.

Chapter 2

Company, Regulatory Bodies, and Laboratory Processes

2.1 Company Profile

The company **Orange Solutions Ltd.** [5] is ISO 9001:2015 certified. Using cutting-edge technology, **OSL** provides an integrated business processor that manages the complete enterprise in real-time from a centralized system. By providing enterprises of all sizes and domains with end-to-end solutions that include functionality, analysis, data security, and integration. Enterprise Resource Planning (ERP) Solutions, Business Process Automation, Web and Smart Apps, Business Intelligence, IoT, AI, IT Infrastructure Development, and Process Automation are covered comprehensively. Figure 2.1 shows the Industrial footprints of OSL so far.

Figure 2.1: Industrial Footprints

2.1.1 Provided Products

They have over 15+ years of experience, provide 50+ modules and over 200 satisfied and active customers. They provide the following products:

- ZABPro- Leading ERP flagship software of OSL
- ZABLite- Cloud-based ERP solutions
- ActFin- Packaged accounting software
- HCMPro- Human resource management solution
- ShopSense Pro- Advanced super shop management software
- ShopSense Lite- Basic retail management software
- MediSense- Hospital management solution
- InnLytics- Fully web-based Hotel management software
- FoodTastic- Restaurant management solution
- VATPrime- NBR Approved VAT Solution.

note: Information collected from https://makeitorange.com.bd/

2.1.2 Technology and Framework

OSL uses the following:

- ZAB framework. *[note: The whole ERP system is build on ZAB framework, a flagship product of OSL]*
- JAVA
- JavaScript
- React
- Flutter
- CSS
- html, php
- SQL
- ORACLE

2.1.3 Software

- Microsoft VS Code
- Notepad++
- Microsoft SQL Server Management Studio
- Crystal Report

2.2 About DGDA and NCL

2.2.1 DGDA

Figure 2.2: DGDA logo

The Directorate General of Drug Administration (**DGDA**), [4] located in Mohakhali, Dhaka, is the regulatory body responsible for upholding the standards of pharmaceutical products in terms of quality, effectiveness, and safety, in accordance with applicable laws and regulations. The organization is making concerted efforts to fulfill the requirements of the pharmaceutical industry in the country, thereby addressing the demands of the populace through the assurance of medicine safety, effectiveness and integrity.

2.2.2 NCL

NCL is an acronym that stands for National Control Laboratory [2]. The testing laboratory operates under the authority of **DGDA**. The individuals in question bear the responsibility of receiving samples, conducting tests, and delivering the outcomes pertaining to both existing and new pharmaceuticals and giving authorization for them to enter the pharmaceutical market. They consists of two wings:

- Drug
- Vaccine

Vaccine wing has three divisions:

- Chemical
- Microbiology
- Animal Lab

2.2.3 Brief Process Description of NCL

NCL receives samples from across the nation. Each sample is assigned a unique sample ID upon reception by **NCL**. There are further information that are initially populated with data provided by the manufacturer.

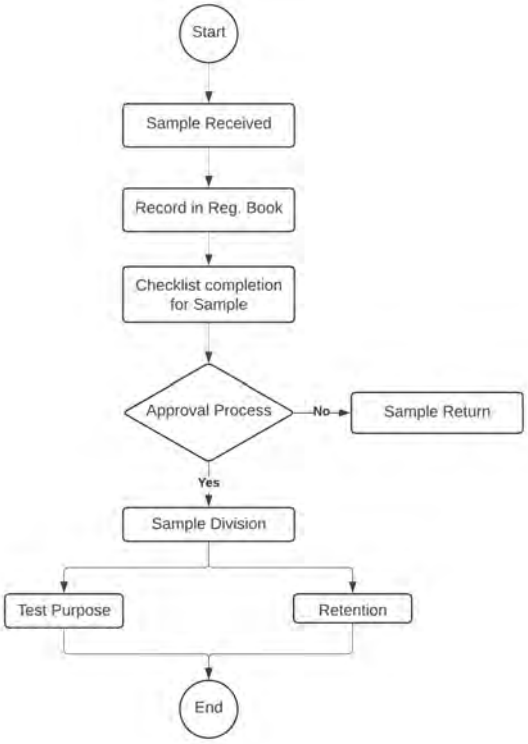


Figure 2.3: NCL, Sample Receive Flow Diagram

The completion of the checklist is contingent upon several factors, including the comprehensive assessment of the sample body, packaging integrity, and the availability of requisite documentation and supplementary apparatus. The formulation of the test request form is undertaken by the office assistant, who tailors it to the specifications provided by the sender or manufacturer. Compliance with these stipulations is imperative for the analyst during the testing phase. Subsequently, both the sample receiving checklist and the test request form necessitate approval from the designated authority before the sample can proceed to testing.
Upon approval, the sample is then allocated by the General Administration (**GA**) to respective Unit Heads, who subsequently delegate the tests to analysts within their purview. Analysts proceed with their assigned tasks and tests, documenting their findings within the Analytical Worksheet. The culmination of this process entails the input of final results derived from testing. In instances where the sample meets the required standards and garners approval from both the Unit Head and the GA, a comprehensive final report is generated.
Conversely, should the sample fail to meet the specified criteria or yield substandard results, the analyst is mandated to initiate an Out of Specification (**OOS**) form and promptly inform the GA. At this juncture, the GA reserves the discretion to initiate a retest, the execution of which involves assigning a different analyst to conduct a blind test using the same sample. Upon completion of the retest, a final determination

regarding the sample's status is made based on the resultant data, culminating in the issuance of a comprehensive final report to the sample's originator. This essentially concludes the main process flow of NCL and our LIMS software currently handles all of these and more. The flow diagram of these processes are given below:

2.2.4 Drug Wing Flow Diagram

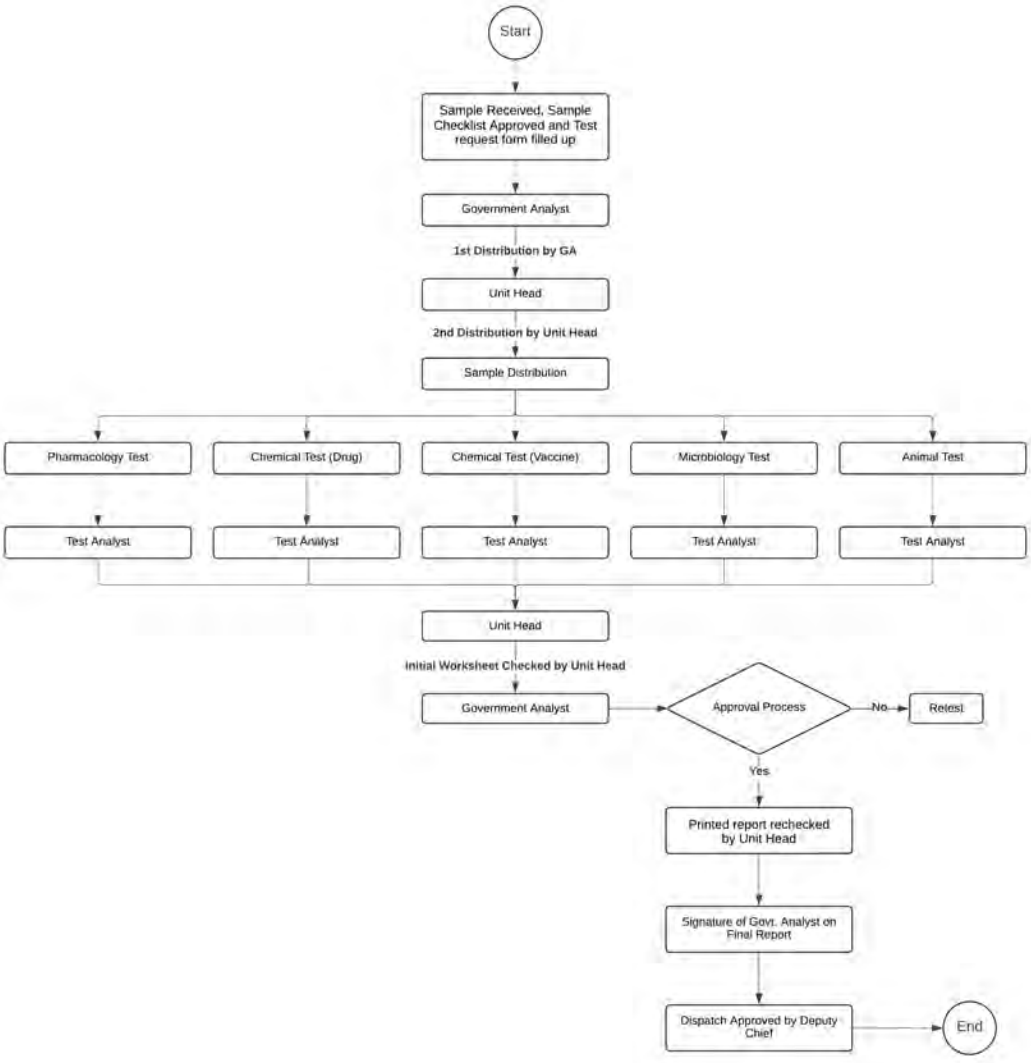


Figure 2.4: NCL, Drug Wing Flow Diagram

The NCL flowchart portrays a streamlined sample analysis process. Upon arrival, analysts verify the accompanying paperwork. A two-step distribution follows, involving a government analyst and a unit head. Next, designated analysts conduct various tests (pharmacology, chemistry, microbiology, animal testing) on the sample. Unit heads then review the analysts' worksheets. Approved results are compiled into a final report, checked by the unit head, and signed off by a government analyst. Finally, a deputy chief authorizes dispatch, concluding the process. This multi-step approach with defined handoffs likely ensures data integrity and analysis quality at the NCL.

2.2.5 Vaccine Wing Flow Diagram

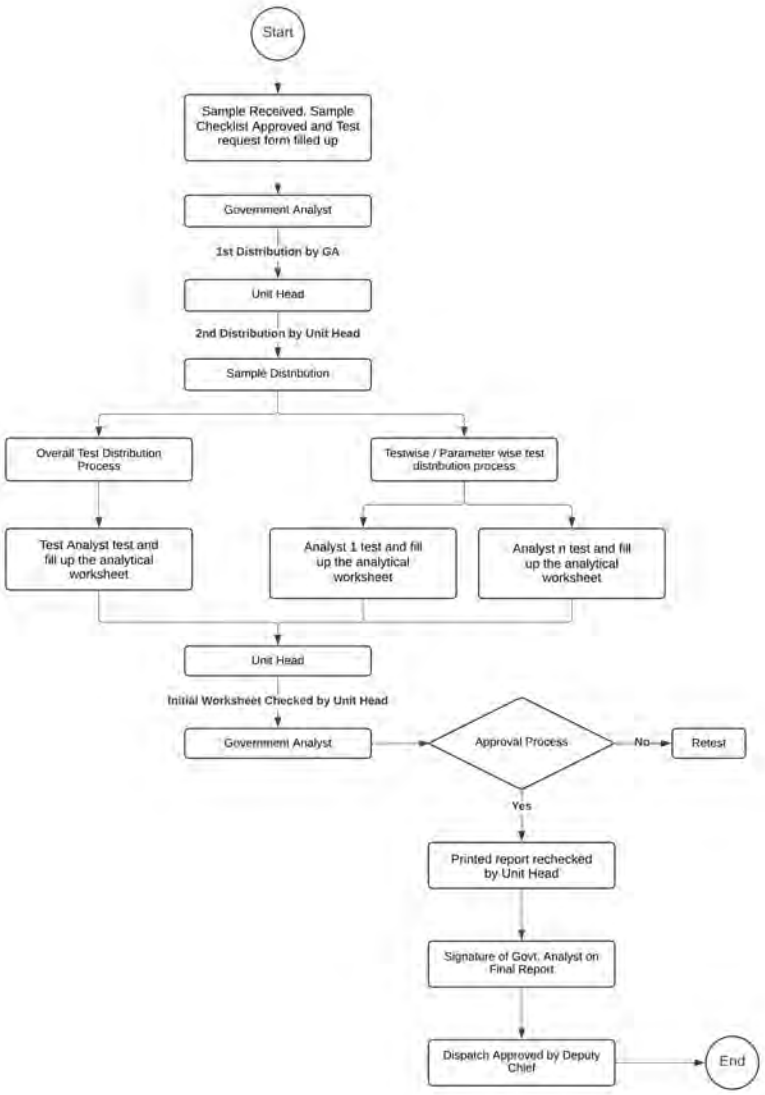


Figure 2.5: NCL, Vaccine Wing Flow Diagram

The provided flowchart outlines the workflow within the National Control Laboratory's vaccine wing. Upon receiving a vaccine sample, analysts verify a checklist before a government analyst assigns it for testing. The flowchart then splits, indicating two potential testing paths based on either the type of test or specific parameters being analyzed. Analysts conduct the assigned tests and document their findings on worksheets. A unit head reviews these worksheets, and if approved, the process progresses to report generation. Rejected worksheets trigger a retesting loop. Finally, a government analyst signs off on the completed report, and a deputy chief dispatches it, finalizing the process.

2.2.6 OOS Form Generation Flow Diagram

The following flowchart outlines the Out-of-Specification (OOS) process used within NCL for quality control function. Upon encountering an OOS result, the analyst conducts a self-audit and informs the unit head. The unit head evaluates the OOS form and directs it for retesting or review by a government analyst. Following potential retesting and review, the unit head approves conforming results or initiates further investigation. The process concludes with final approval, report generation, and OOS result publication.

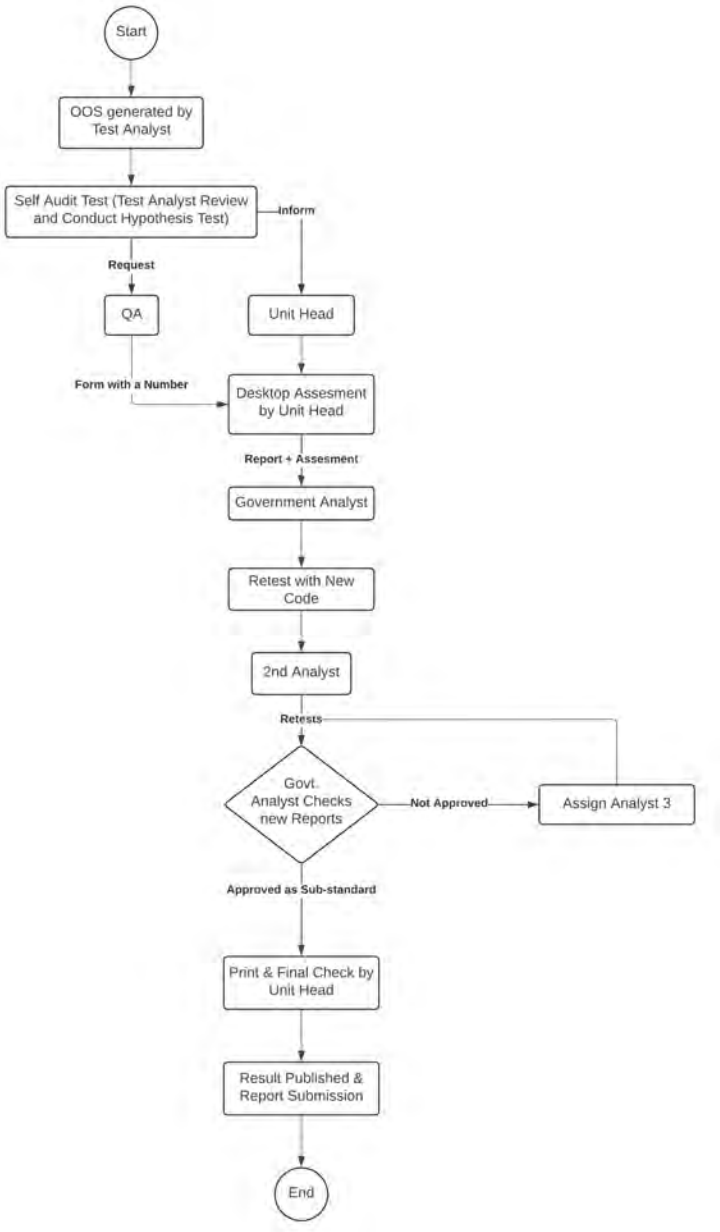


Figure 2.6: NCL, OOS Flow Diagram

Besides these our software will also handle the following modules of **NCL** which are already completely made and fully functional currently:

Figure 2.7: NCL, All Modules

- **Administration:** Helps manage users, define roles, assign privileges etc.
- **Inventory Management:** Handles store Requisition, issue from store, return entry and approval etc.
- **Approval Setup:** Sets up the approval Process, superior setup, report signatories etc.
- **Quality Management System (QMS):** Helps input and generate Incident report, Document Change report, Discontinuation report, Corrective and Prevention Activities (CAPA) report.
- **Equipment Services:** Records Equipment Register, Service Request, Log etc.
- **Animal Lab Process:** Handles all test data input, checking and approval of Animal lab which is a very important part of Vaccine Wing. This module also includes Animal requisition and receiving forms.

Chapter 3

My Learning

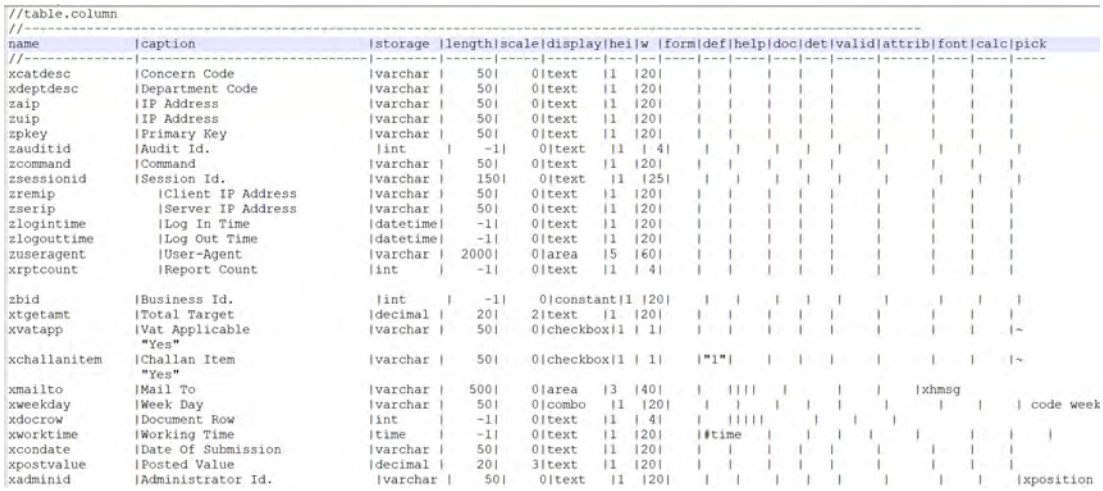
My internship at Orange Solutions Ltd. (OSL) from November 2022 to September 2023 proved to be a valuable learning experience. Initially, I faced challenges due to the unfamiliar syntax used in OSL's ZAB framework. However, with guidance from my supervisor, Tanvir Khan Topu, and my own perseverance, I steadily improved my understanding.
By February 2023, I had built the project's core structure, including master data screens and the main process. Regular visits to the client site, NCL, allowed me to refine each screen and module, addressing identified issues and ensuring functionality.
The next phase involved generating reports based on user input, completed by mid-May 2023. User Acceptance Testing (UAT) and Operational Acceptance Testing (OAT) revealed further areas for improvement, which we addressed.
As of July 1st, 2023, the LIMS software is successfully operational at NCL. While there's room for further user experience enhancements, we've received positive feedback thus far.

3.1 ZAB Framework

Similar to other frameworks, ZAB [1] facilitates efficient development by providing pre-defined syntax and libraries. This reduces the initial time investment required for starting a project from scratch. As Orange Solutions Ltd.'s (OSL) flagship framework, ZAB utilizes a unique set of commands and attributes that initially presented a learning curve. However, with consistent use, I gained a deeper understanding and began to appreciate its benefits. The framework has the following extensions:

3.1.1 Extension: [.column]

Any file with .column extension is where all the columns needed to build a table in the Database (**DB**) is kept. It has predefined syntax such as name, type, length, default value, attribute etc.



```
//table.column
//-------------------------------------------------------------------------------------------------
name            |caption                    |storage |length|scale|display|hei|w  |form|def|help|doc|det|valid|attrib|font|calc|pick
//--------------|---------------------------|--------|------|-----|-------|---|---|----|---|----|---|---|-----|------|----|----|----
xcatdesc        |Concern Code               |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
xdeptdesc       |Department Code            |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
zaip            |IP Address                 |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
zuip            |IP Address                 |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
zpkey           |Primary Key                |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
zauditid        |Audit Id.                  |int     |    -1|    0|text   |1  | 4|    |   |    |   |   |     |      |    |    |
zcommand        |Command                    |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
zsessionid      |Session Id.                |varchar |   150|    0|text   |1  |25|    |   |    |   |   |     |      |    |    |
zremip          |Client IP Address          |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
zserip          |Server IP Address          |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
zlogintime      |Log In Time                |datetime|    -1|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
zlogouttime     |Log Out Time               |datetime|    -1|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
zuseragent      |User-Agent                 |varchar |  2000|    0|area   |5  |60|    |   |    |   |   |     |      |    |    |
xrptcount       |Report Count               |int     |    -1|    0|text   |1  | 4|    |   |    |   |   |     |      |    |    |

zbid            |Business Id.               |int     |    -1|    0|constant|1 |20|    |   |    |   |   |     |      |    |    |
xtgetamt        |Total Target               |decimal |    20|    2|text   |1  |20|    |   |    |   |   |     |      |    |    |
xvatapp         |Vat Applicable             |varchar |    50|    0|checkbox|1 | 1|    |   |    |   |   |     |      |    |    |~
                 "Yes"
xchallanitem    |Challan Item               |varchar |    50|    0|checkbox|1 | 1|    |"1"|    |   |   |     |      |    |    |~
                 "Yes"
xmailto         |Mail To                    |varchar |   500|    0|area   |3  |40|    |   ||||  |   |   |     |      |xhmsg
xweekday        |Week Day                   |varchar |    50|    0|combo  |1  |20|    |   |    |   |   |     |      |    |    | code week
xdocrow         |Document Row               |int     |    -1|    0|text   |1  | 4|    |   |||||     |   |   |     |      |    |    |
xworktime       |Working Time               |time    |    -1|    0|text   |1  |20|    |#time   |   |   |   |     |      |    |    |
xcondate        |Date Of Submission         |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |
xpostvalue      |Posted Value               |decimal |    20|    3|text   |1  |20|    |   |    |   |   |     |      |    |    |
xadminid        |Administrator Id.          |varchar |    50|    0|text   |1  |20|    |   |    |   |   |     |      |    |    |xposition
```

Figure 3.1: Columns of Tables

3.1.2 Extension: [.def]

The extension .def is used to create tables. A file with this extension has to be declared in the following way [Figure 3.2] and with then going into the **zabadmin** panel and with the click of a button the table is created.

```
xcodeslab.def
1  product ZAB ERP
2
3  table xcodeslab
4    caption "Codes"
5    columns ztime, zutime,zauserid,zuuserid,zid,xtype,xlcode,xlong,zactive,~
6              xadd,xothers,xmodel,xobvcond,xsampleid,xtypeobj,xsigndate9,xanaldt
7
8
9
10   primary key zid,xlcode,xtype
11
12   foreign key
13         zid references zbusiness.zid
14   end foreign key
15
16
17
18   end table
19
20
```

Figure 3.2: Creating a Table

3.1.3 Extension: [.menu]

The extension .menu helps create a menu that can hold other sub-menu and screens. For example nclcnp.menu holds the screens for NCL Lab Codes, Master data and Parameters.

```
//NDCL LAB SCREENS.menu
//-------------------------------------------------------------------
option                              |action
//-------------------------------------------------------------------
//
//Employee Information                          |#classPath+"/login?screen=pdmsthrdn"
//Employee Information Check                            |#classPath+"/login?screen=pdmstchk"
//Employee Information Approve                          |#classPath+"/login?screen=pdmstaprv"
Test Master     |#classPath+"/login?screen=testmaster"
Parameter Test Master     |#classPath+"/login?screen=parameterheader"
Source Master     |#classPath+"/login?screen=srcmastertest"
Division Master         |#classPath+"/login?screen=divisionmaster"
Lab/ Wing Master         |#classPath+"/login?screen=lbmaster"
Product Master     |#classPath+"/login?screen=caitemall"
Item Group     |#classPath+"/login?screen=xlcodesgitem"
Item Category     |#classPath+"/login?screen=itemcategory"
Generic Master                    |#classPath+"/login?screen=genericmsater"
Store Master     |#classPath+"/login?screen=strmaster"
Sub-Store Master     |#classPath+"/login?screen=substrmaster"
Pack Size Master     |#classPath+"/login?screen=pksizemater"
```

Figure 3.3: Creating a Menu

3.1.4 Extension: [.screen]

The extension .screen converts a file into a screen that the users can see, edit and operate on. This is essentially the place where the VIEW is built and manipulated according to the need of the particular module.

- **Initialization**

```
screen srcmastertest

  sidebar list one
    sections form detail, jscript myscript

    list one
       caption "Source List"
       table xcodes
       order xcode, xtype
      // fixed xtype
       rows 20
       select "xtype='Source Master'"
       objects  xcode attrib(link "login?screen=srcmastertest&command=Show&xcode=?"), -
                xprefix, xbrand,desc equals( xmadd+' '+xdistrictop+'- '+ xzip)

       header "Source Name","Prefix", "Brand Name" , "Address and Post Code"
  end list


  form detail
       caption "Source Master"
       table xcodes
       primarykey  xcode,xtype
       order xcode,xtype
       //return "login"
       //fixed xcodes,xtype,xrow
       layout 2
       pstyle 3
       objects Show, Clear,Add,Update,Delete,~//Back
             xcode, xprefix, xbrand, xdistrictop, xmadd, xzip, xlong, zactive, xtype display(hide)
```

Figure 3.4: Initialization

The ZAB framework utilizes a specific syntax for building screens. Each screen definition starts with the "screen" call followed by a chosen name. Sidebar lists are declared using the "sidebar" tag. Form elements for insertion, update, or deletion are defined within "sections" along with JavaScript code (jscript).

To create a form, a header is defined with the "caption" tag. Next, the table name and its primary key are specified, followed by the data selection order

for the query. ZAB offers pre-defined styles using "pstyle" and "layout" tags. Finally, the "objects" tag lists the fields to be displayed on the screen. As per OSL convention, button names appear on the top row, followed by the remaining fields.

- **Modification of Fields/Columns**

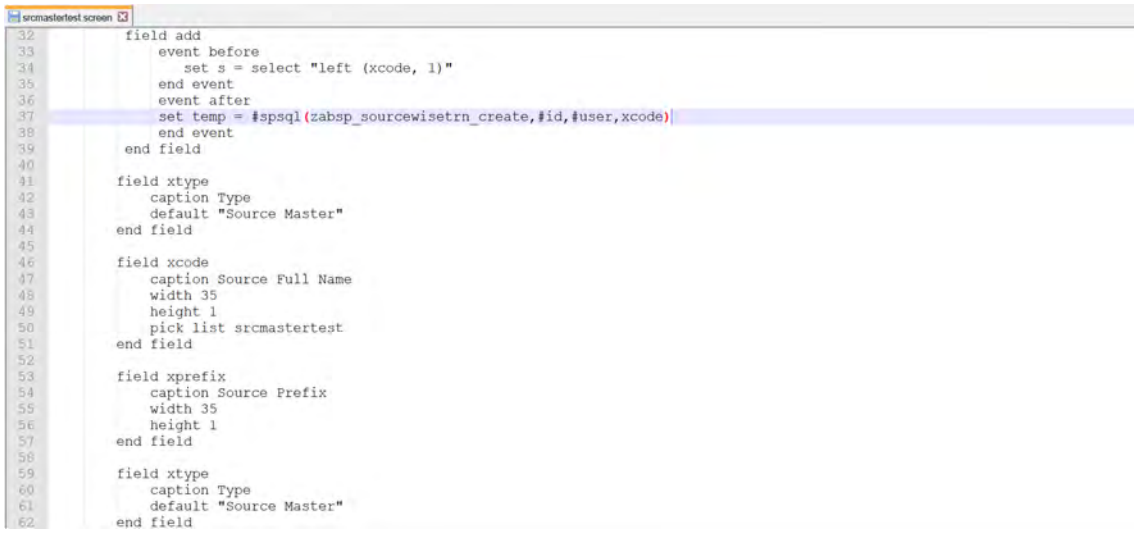


Figure 3.5: Modification

Basic field information like caption, size, length, and default value are initially declared in the table.column definition. Further modifications are done using the field fieldName tag. Fields can have attributes like default to set defaults or pick list with listName to choose from a pre-defined list.

The event before tag defines actions executed before a specific button's operation. Conversely, the event after tag specifies actions to be performed afterward.

Line 37 in Figure: 3.5 demonstrates calling the zabsp_sourcewisetrn_create procedure using #spsql(zabsp_sourcewisetrn_create,#id,#user,xcode). This procedure takes three arguments: #id (business ID), #user (user ID), and xcode (source name).

- **Closing**
  A form is concluded with the "end form" tag. Optional CSS and JavaScript code are to be placed afterwards [As seen in Figure 3.6]. Finally, the entire screen definition is closed with the "end screen" tag.

```
        embed onsubmit="submitit(this)"

       end form

jscript myscript

       <script language="javascript" type="text/javascript">
       var detail
       function clicked(b){
         detail=b.value
       }
       function budgeted(b){
         detail="budgeted"
       }
       function submitit(form){

           if (detail=="Budget"){
           form.screen.value = "acbudget"
           form.searchbutton.value = "Top"
           //return false
         }
       }

       </script>
    end jscript


end screen
```

Figure 3.6: Closing

- **Result**

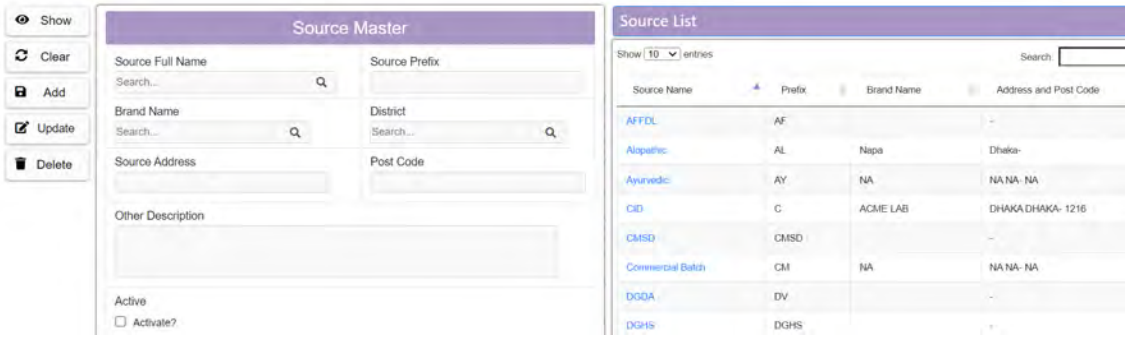


Figure 3.7: Source Master

Figure 3.7 showcases the srcmastertest.screen screen generated by the combined code. It displays a list on the left side defined by the "sidebar" tag and arranges the form elements in the same order as specified in the "objects" tag.

3.2 ZABadmin Panel

This panel is incredibly helpful as it allows to create or drop a table or alter tables by inserting new column or dropping one with the click of button. It has other functionalities too as seen in [Figure 3.8]

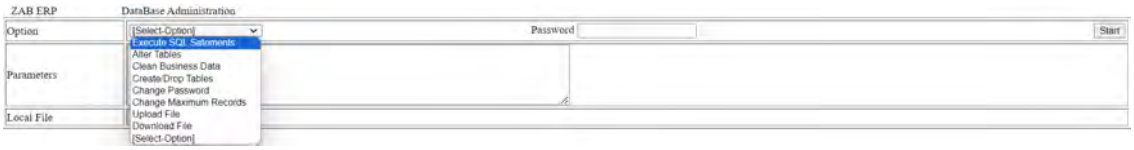


Figure 3.8: ZABadmin Panel

3.3 Crystal Report

In February 2023, I began utilizing Crystal Reports, a powerful report-building software. I leveraged its design capabilities to develop a comprehensive suite of reports crucial for NCL's daily operations. These reports not only streamline data visualization but also incorporate e-signatures from responsible personnel. This implementation significantly contributes to NCL's transition towards a paperless and automated workflow, enhancing efficiency and reducing manual workloads.

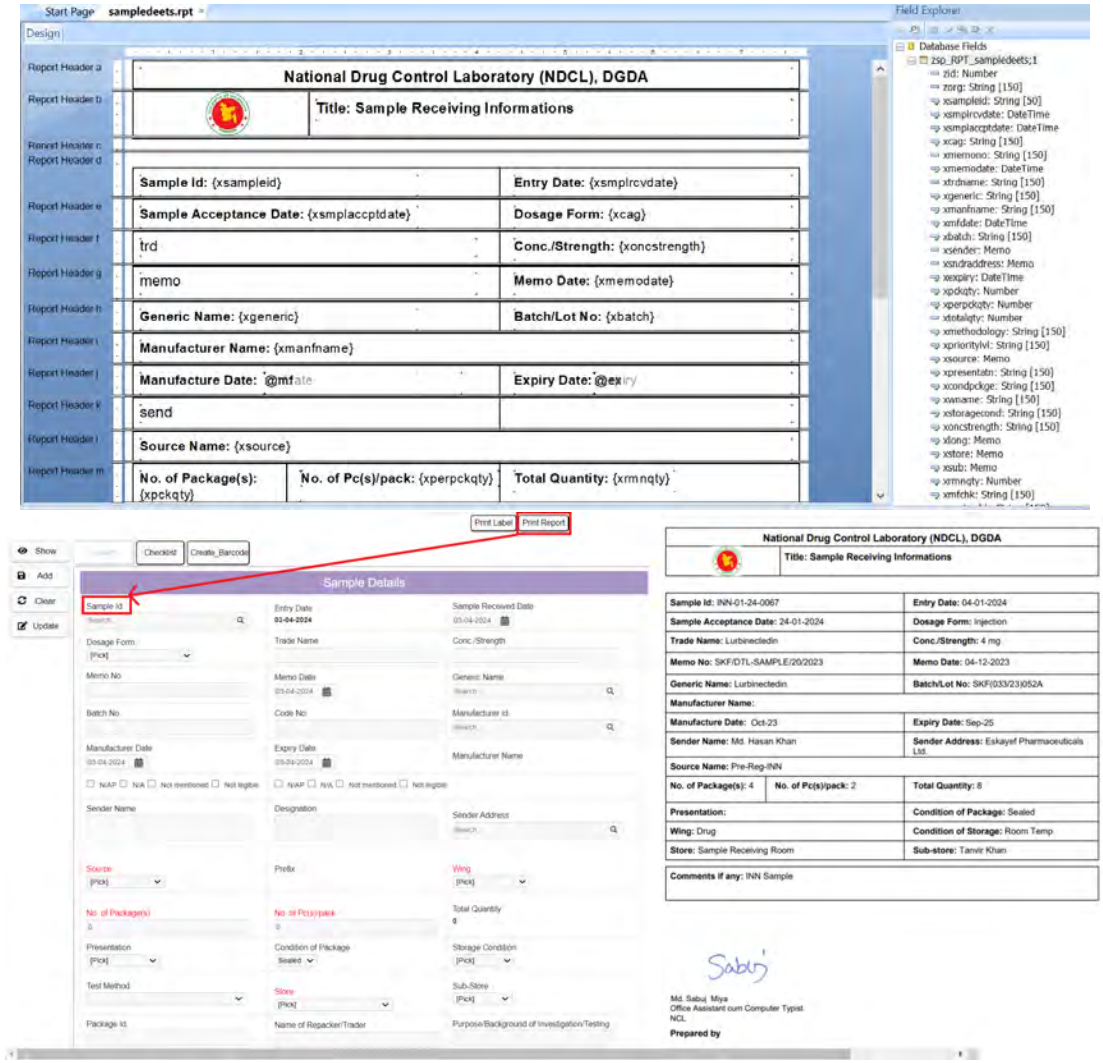


Figure 3.9: Sample Receive Report Design and Result

3.4 MS SQL

My internship at OSL significantly expanded my knowledge of Microsoft SQL Server [6] While familiar with basic queries (CREATE, DROP, UPDATE, ALTER, INSERT) from my studies, real-world experience provided a deeper understanding. This included:

- Effective Query Writing: I honed my skills in crafting optimized queries that ensure system performance.
- Conditional Logic: Mastering conditional statements allowed me to write more robust and flexible queries.
- SQL Procedures: Initially a challenge, I now confidently utilize procedures for process flow management, data table manipulation (building and editing), and report generation.

1. **Process Flow Procedure**

Figure 3.10: Process Flow Procedure

This stored procedure (`zabsp_sourcewisetrn_create`) to add source transaction records. It checks for duplicates in `xtrn` using `@zid` and `@sourcenam`. If none exist, it inserts a new record with creation details, username `@user`, source name `@sourcenam`. It is imperative for generating Sample Id. as it keeps track of each source last count and increments it accordingly.

2. **Data-table Building Procedure**

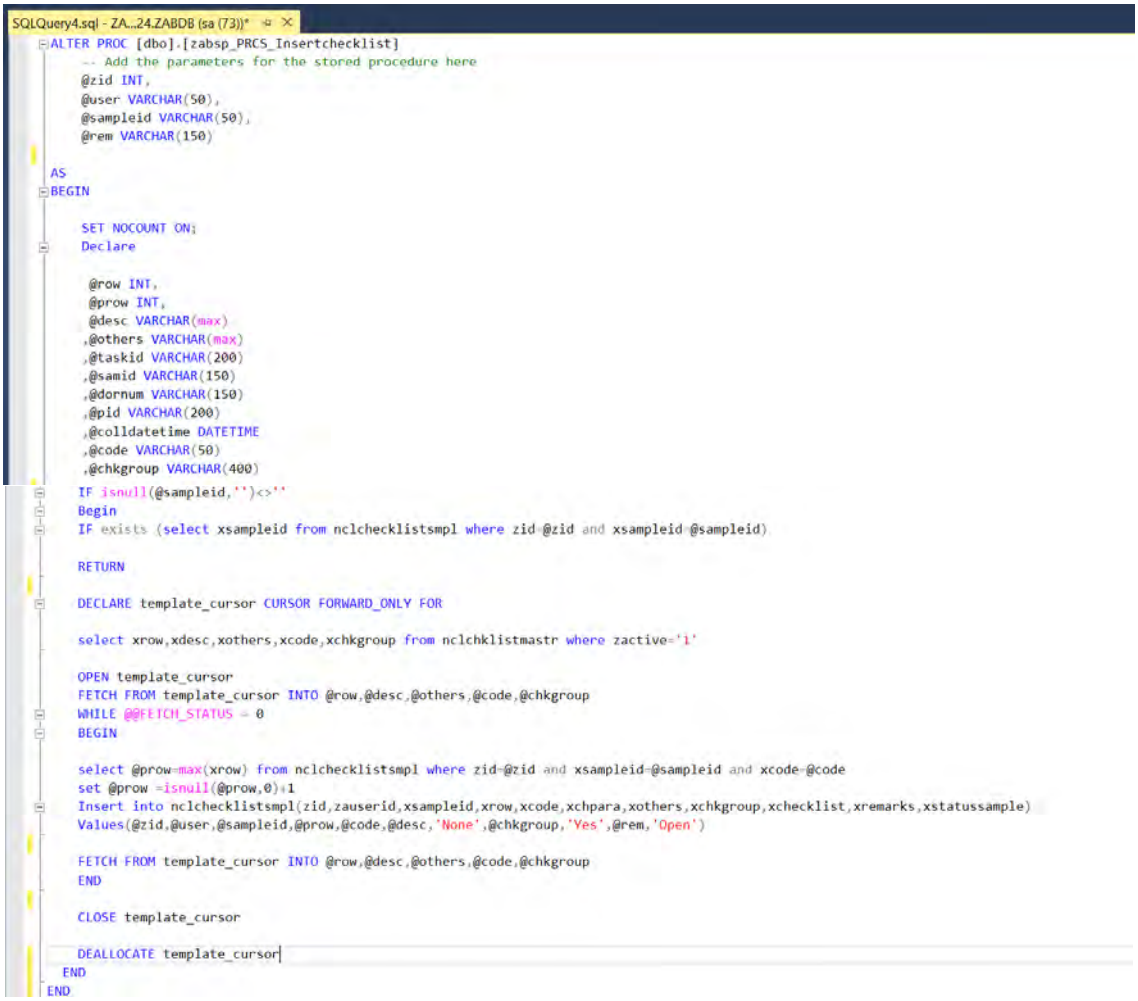


Figure 3.11: Data-table Building Procedure

The (`zabsp_PRCS_Insertchecklist`) procedure selects checklist parameters from the `nclchklistmastr` table where `zactive` is '1'. It retrieves details such as row number `xrow`, parameters `xdesc`, additional information `xothers`, and checklist group `xchkgroup`. These details are crucial for populating the checklist for a new sample entry, ensuring comprehensive data capture and adherence to predefined checklist templates.

3. **Report Creating Procedure**

```
SQLQuery5.sql - ZA...24.ZABDB (sa (58))*
--EXEC zsp_RPT_sampledeets 400111,'N/AP-05-23-0005'
ALTER PROC [dbo].[zsp_RPT_sampledeets]
      @zid INT,
      @sampleid VARCHAR (50)

AS

DECLARE
      @zorg VARCHAR(150),
      @smplrcvdate DATETIME,
      @smplaccptdate DATETIME,
      @cag VARCHAR(150),
      @memono VARCHAR(150),
      @memodate DATETIME,
      @trdname VARCHAR(150),
      @generic VARCHAR(150),
      @manfname VARCHAR(150),
      @mfdate DATETIME,
      @batch VARCHAR(150),
     @sender VARCHAR(max),
     @sndraddress VARCHAR(max),
     @expiry DATETIME,
     @pckqty INT,
     @perpckqty INT,
     @totalqty INT,
     @methodology VARCHAR(150),
     @priorityLvl VARCHAR(150),
     @source VARCHAR(max),
     @presentatn VARCHAR(150),
     @condpckge VARCHAR(150),
      @wname VARCHAR(150),
      @storagecond VARCHAR(150),
      @oncstrength VARCHAR(150),
      @long VARCHAR(max),
      @store VARCHAR(max),
      @sub VARCHAR(max),
      @rmnqty INT,
      @mfchk VARCHAR(150),
      @expirechk VARCHAR(150),
      @rcvid VARCHAR(50)

      DECLARE @table TABLE( zid INT,
      zorg VARCHAR(150),
      xsampleid VARCHAR (50),
      xsmplrcvdate DATETIME,
      xsmplaccptdate DATETIME,
      xcag VARCHAR(150),
      xmemono VARCHAR(150),xmemodate DATETIME,xtrdname VARCHAR(150),xgeneric VARCHAR(150),xmanfname VARCHAR(150),xmfdate DATETIME,xbatch VARCHAR(150),
      xsender VARCHAR(max),xsndraddress VARCHAR(max),xexpiry DATETIME,xpckqty INT,xperpckqty INT,xtotalqty INT,xmethodology VARCHAR(150),
      xpriorityLvl VARCHAR(150),xsource VARCHAR(max),xpresentatn VARCHAR(150),xcondpckge VARCHAR(150),xwname VARCHAR(150),xstoragecond VARCHAR(150),
      xoncstrength VARCHAR(150),xlong VARCHAR(max),xstore VARCHAR(max),xsub VARCHAR(max),xrmnqty INT,xmfchk VARCHAR(150),xexpirechk VARCHAR(150),
      xrcvid VARCHAR(50)
      )

SELECT @zorg=zorg FROM zbusiness WHERE zid=@zid

      DECLARE spdo_cursor CURSOR FORWARD_ONLY FOR

      select xsampleid,xsmplrcvdate,xsmplaccptdate,xcag,CAST(xmemono AS nvarchar),xmemodate,CAST(xtrdname AS nvarchar),xgeneric,m.xmanfname,xmfdate,xbatch,
      CAST(xsender AS nvarchar),CAST(xsndraddress AS nvarchar),xexpiry,xpckqty,xperpckqty,xtotalqty,xmethodology,xpriorityLvl,xsource,xpresentatn,
      xcondpckge,n.xwname,xstoragecond,xoncstrength,n.xlong,c.xlong,b.xlong,xrmnqty,xmfchk,xexpirechk,xrcvid
      from nclsample n left join manufacturer m on n.zid=m.zid and n.xmanfname=m.xman
      left join xcodes c on n.zid=c.zid and n.xstore=c.xcode
      left join xcodes b on n.zid=b.zid and n.xsubstore=b.xcode
      where n.zid=@zid and n.xsampleid=@sampleid

      OPEN spdo_cursor
      FETCH FROM spdo_cursor INTO @sampleid,@smplrcvdate,@smplaccptdate,@cag,@memono,@memodate,@trdname,@generic,@manfname,@mfdate,@batch,
      @sender,@sndraddress,@expiry,@pckqty,@perpckqty,@totalqty,@methodology,@priorityLvl,@source,@presentatn,@condpckge,@wname,
      @storagecond,@oncstrength,@long,@store,@sub,@rmnqty,@mfchk,@expirechk,@rcvid

      WHILE @@FETCH_STATUS=0
      BEGIN

      INSERT INTO @table(zid,zorg,xsampleid,xsmplrcvdate,xsmplaccptdate,xcag,xmemono,xmemodate,xtrdname,xgeneric,xmanfname,xmfdate,xbatch,
      xsender,xsndraddress,xexpiry,xpckqty,xperpckqty,xtotalqty,xmethodology,xpriorityLvl,xsource,xpresentatn,xcondpckge,xwname,
      xstoragecond,xoncstrength,xlong,xstore,xsub,xrmnqty,xmfchk,xexpirechk,xrcvid
      )
      VALUES (@zid,@zorg, @sampleid,@smplrcvdate,@smplaccptdate,@cag,@memono,@memodate,@trdname,@generic,@manfname,@mfdate,@batch,
      @sender,@sndraddress,@expiry,@pckqty,@perpckqty,@totalqty,@methodology,@priorityLvl,@source,@presentatn,@condpckge,@wname,
      @storagecond,@oncstrength,@long,@store,@sub,@rmnqty,@mfchk,@expirechk,@rcvid)

      FETCH NEXT FROM spdo_cursor INTO  @sampleid,@smplrcvdate,@smplaccptdate,@cag,@memono,@memodate,@trdname,@generic,@manfname,@mfdate,@batch,
      @sender,@sndraddress,@expiry,@pckqty,@perpckqty,@totalqty,@methodology,@priorityLvl,@source,@presentatn,@condpckge,@wname,
      @storagecond,@oncstrength,@long,@store,@sub,@rmnqty,@mfchk,@expirechk,@rcvid
      END
```

Figure 3.12: Sample Receive Report Procedure

The (`zsp_RPT_sampledeets`) procedure retrieves detailed information about a specific Sample Id. `@sampleid` within the context of a given organization `@zid`. It gathers attributes such as organization name sample dates, category, memo details, trade and generic names, manufacturer details, batch, sender info, expiry date, packaging quantities, methodology, priority level, source, presentation, packaging condition, warehouse name, storage conditions etc. This data is fetched from tables like `nclsample`, `manufacturer`, and `xcodes`, and stored in a temporary table `@table`, which is then returned to generate data for the report generation through crystal report.

Chapter 4

My Contribution

From day one at OSL, I was eager to tackle the LIMS project. Resource videos provided a springboard for rapid knowledge acquisition, and I quickly transitioned from eager learner to contributing developer.
The project unfolded in three distinct phases, each offering opportunities to showcase my growing skillset across various software modules.

4.1 First Phase: Laying the Foundation

My initial task was designing and building master data screens encompassing all essential fields for the software. This critical foundation was completed within two weeks. These screens are displayed in Figure 4.1.

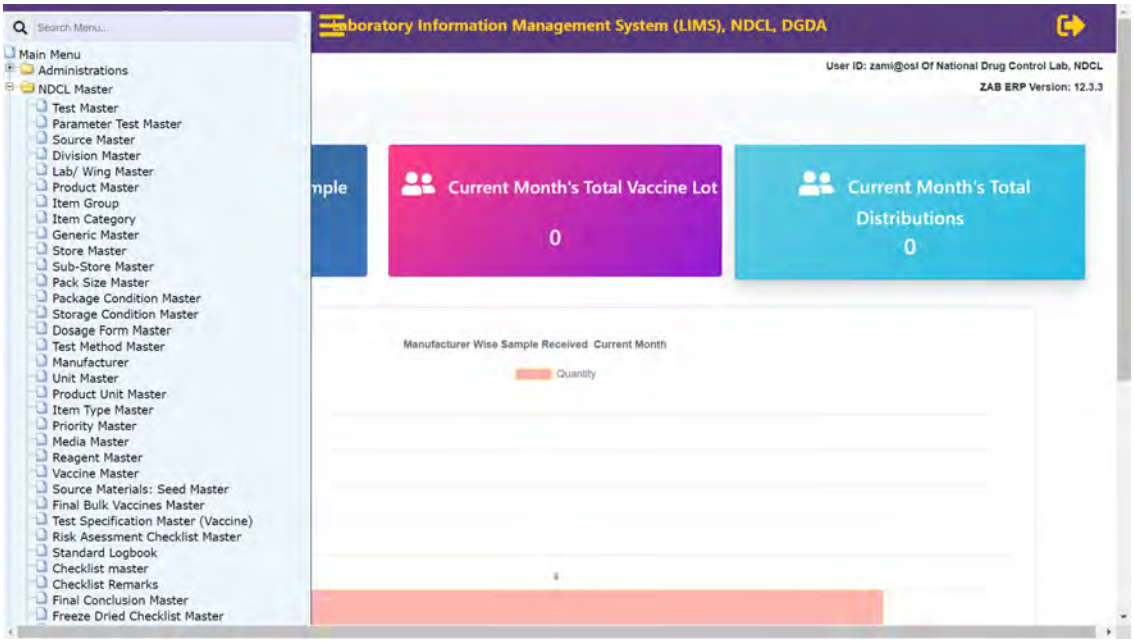


Figure 4.1: NCL, Master Screens

With the groundwork laid, I tackled the core functionalities. I developed screens like "Sample Details," which facilitates sample receiving and entry. Additionally, I created a procedure to generate unique barcodes for each sample, as shown in Figure 4.2.

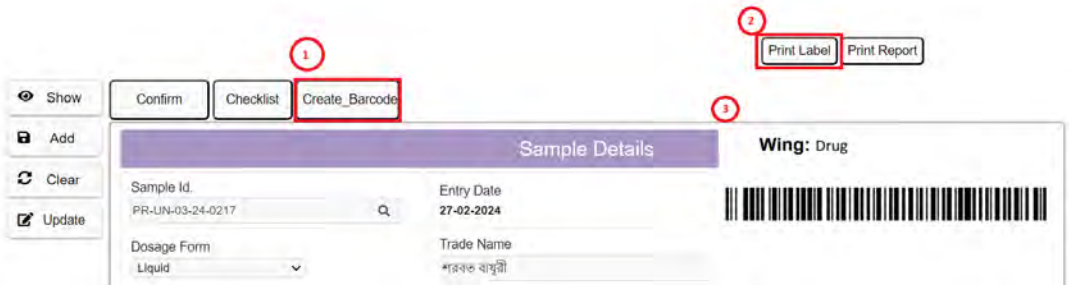


Figure 4.2: Sample Barcode

Following this, I built the "Test Request and Checklist" module alongside its approval screen. Once samples were distributed and tested, I completed the "Analytical Worksheet," where analysts record results and data for each sample. Each of these tasks involved creating various UI elements like lists, dropdowns, and procedures.

4.1.1 Beyond the Core: Expanding Functionality

Furthermore, I independently planned, designed, and developed the entire "Animal Lab" process. This included constructing an "Animal Requisition and Receive" form equipped with a checklist and integrated functionalities for inputting and approving data within the module.

4.1.2 Collaboration and Quality Assurance

In collaboration with my supervisor, I successfully completed the "QMS" module. The process begins with requesting and approving a QMS form within the software. Subsequently, depending on the request type, analysts can generate reports like "Incident", "Discontinuation", "Document Change" or "CAPA" and submit them electronically to the Unit Head and QA for streamlined approval.

4.2 Second Phase: Streamlining Reporting with Automation

The second phase presented a crucial challenge: designing and automating NCL's entire report generation process within a tight one-month deadline. This phase marked my introduction to Crystal Reports, a new tool that initially felt cumbersome. However, by meticulously tackling the intricacies, we navigated this phase efficiently and delivered on time.

The transformation in report generation is best exemplified by the sample details report generation process already shown previously. Before LIMS anyone requiring any information regarding a particular sample needed to visit the sample store and their unit heads but now all it needs is selecting a Sample Id. and clicking the "Print Report" button [As shown in Figure: 3.9]

Near the end of this phase I also was required to learn Java class reporting which relied heavily on HTML and CSS as well. I was required to first design the report according to already established NCL template using HTML and CSS then connect

it to HttpServlet using JAVA to fetch the required data from DB and complete the report. Previously, the manual system relied heavily on communication:

- Sample details and checklist retrieval from the sample receiver
- Test request form and assigned analyst information from unit heads
- Analytical worksheet data from the analyst
- Physical GA approval
- Finally, the typist would compile and type the entire report before printing.

Our solution revolutionized this process. Now, generating a report requires just a single button press!
The user simply selects a Sample Id. from the approved list and clicks "Search" This action triggers a .class file which contains a query that fetches all relevant information for that specific ID from the database. The data is then compiled and presented as a final report, complete with all necessary signatures, ready for printing and delivery to the sample sender (as illustrated in Figure 4.3).

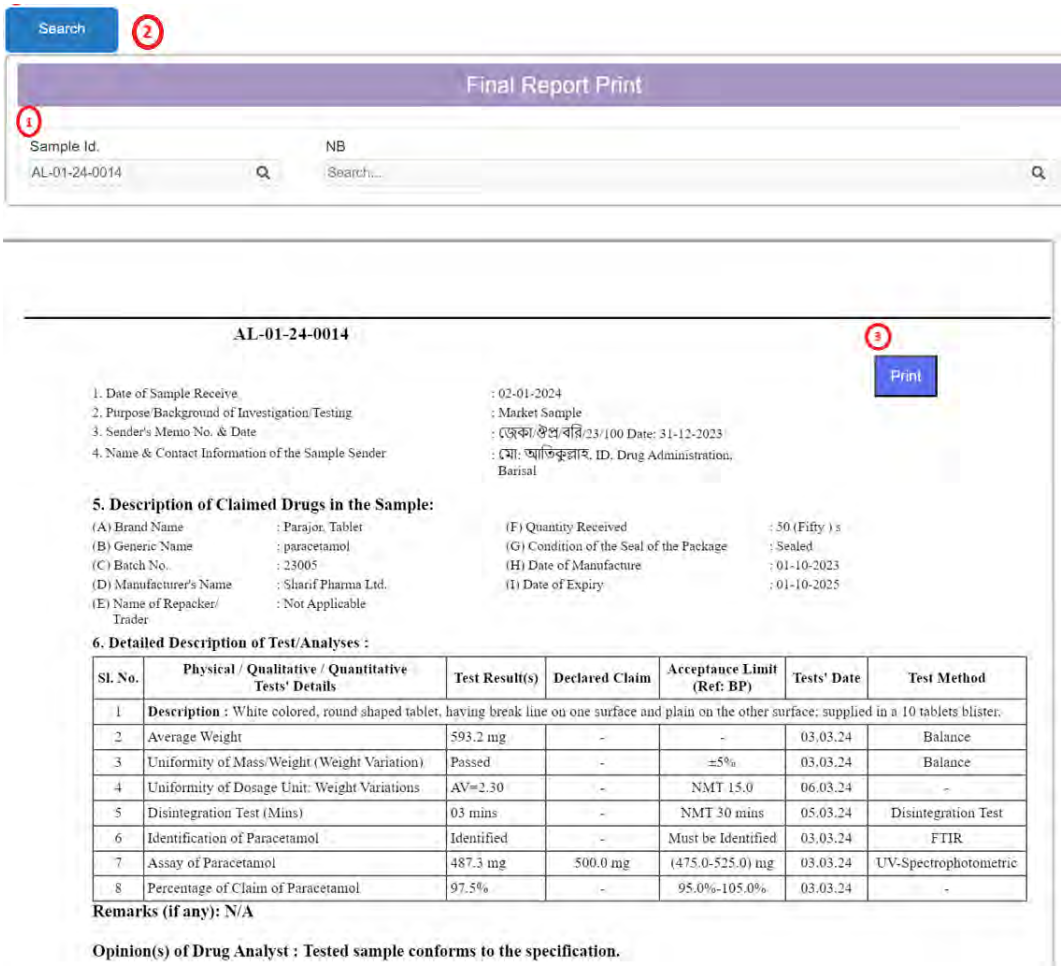


Figure 4.3: Final Report

4.3 Third Phase: Deployment and Beyond

The third phase marked the culmination of months of effort: transitioning the system from testing servers to the live production environment at NCL. This critical phase began with two weeks of comprehensive user training to ensure smooth software adoption. Since July 1st, 2023, the LIMS has been successfully operational at NCL. Building upon the core functionalities, I further enhanced the system by creating several data-driven reports tailored to NCL's specific needs:

- Sample Report: This comprehensive report allows users to generate detailed information for samples received within a defined date range. Additionally, sorting capabilities by manufacturer, source, or generic type provide valuable insights.
- Sample TAT Report: This report focuses on turnaround time (TAT) analysis for each sample. Users can track the time taken for crucial steps like receiving, checklist creation, distribution, result entry, and dispatch, enabling improved efficiency monitoring.
- Equipment List: This report serves as a centralized repository for managing equipment details across both the Drug and Vaccine wings. It encompasses service logs, origin information, model numbers, and equipment status.
- Reagent List: This report streamlines reagent management by providing a consolidated view of current and new reagents, including manufacturer details, manufacturing and expiry dates, potency, and other relevant information.

These reports, along with others available at the user's fingertips, empower NCL to centralize and streamline operations across all departments.

4.4 Challenges and Overcoming Obstacles

As with any significant undertaking, this project presented its own set of challenges. Initially, the ZAB framework, a new language with distinct syntax, presented a learning curve. However, through perseverance and dedication, I was able to master it effectively.

Another initial hurdle was writing and executing SQL procedures. My first attempt took two frustrating days, but with invaluable guidance from my supervisor, Tanvir Khan Topu, I overcame this obstacle and now consider procedures my preferred method for report generation.

Perhaps the most significant challenge was user adoption. Transitioning from long-standing paper-based record-keeping ingrained in many users' workflows proved difficult. While some users remain hesitant, embracing new technologies often takes time. Fortunately, the governing body and many others recognize the long-term benefits of LIMS, including improved organization, time savings, and cost reduction.

I want to reiterate that successfully navigating these challenges and flourishing in my first professional experience is largely attributed to Tanvir Khan Topu's unwavering support, guidance, and understanding. His mentorship fostered an environment that enabled me to perform confidently and complete my tasks with a sense of accomplishment.

Chapter 5

Conclusion

The internship experience at Orange Solutions Ltd has proven to be a highly valuable opportunity for my professional advancement and personal development. The process of developing the Laboratory Information Management System (LIMS) software has presented a series of difficulties, but has also yielded significant benefits. I have been fortunate to have the opportunity to collaborate with a highly committed and very skilled supervisor in Tanvir Khan Topu, benefiting from his extensive knowledge and expertise, as we worked together towards a shared objective. During the course of the development process, I have been able to not only apply my previous knowledge in software engineering and database administration but also improve in every aspect of them and hone my skills.

One of the most notable observations derived from this internship experience has been the direct observation of the influence exerted by our Laboratory Information Management System (LIMS) software on the optimisation and efficacy of laboratory procedures at the National Control Laboratory in Bangladesh. The software developed has effectively optimized data management processes, enhanced data integrity, and fostered efficient cooperation among personnel inside the laboratory.

This internship has also reinforced the importance of effective communication and teamwork in the software development process. The experience of working closely with colleagues, exchanging ideas, and resolving challenges together has been instrumental in my personal and professional growth. I am grateful for the mentorship and guidance provided by the team, which has played a pivotal role in my learning journey.

I think due to the culmination of all these I have been able to have just the right amount of impact in the project for my superiors to take notice and offer me to become a permanent member of the OSL team. I have gladly accepted that request and have been a United Group employee for 6 months now. I would like to reiterate and conclude this report by thanking my supervisor Tanvir Khan Topu for always being supportive and like an actual big brother to me and I will forever be grateful to him.

Bibliography

[1] ZAB, *ZABERP Web Based ERP Solution — Leading ERP and Cloud*, Accessed on September 15, 2023, 2020. [Online]. Available: https://www.zaberp.com/new/.

[2] World Health Organization, *Quality Control Laboratories: Bangladesh National Control Laboratory*, Accessed on September 15, 2023, 2022. [Online]. Available: https://extranet.who.int/pqweb/quality-control-labs/bangladesh-ncl.

[3] CloudLIMS, *What is a LIMS?* Accessed on September 15, 2023, 2023. [Online]. Available: https://cloudlims.com/what-is-a-lims/.

[4] Directorate General of Drug Administration (DGDA), *Official Website of Directorate General of Drug Administration (DGDA), Bangladesh*, Accessed on September 15, 2023, 2023. [Online]. Available: https://dgda.portal.gov.bd/.

[5] M. I. Orange, Accessed on September 15, 2023, 2023. [Online]. Available: https://makeitorange.com.bd/.

[6] Microsoft Corporation, *Microsoft SQL Server 2019*, Accessed on September 15, 2023. [Online]. Available: https://www.microsoft.com/en-us/sql-server/sql-server-2019.