National College of Ireland

Higher Diploma in Computing

Software Development

2020/2021

Vanessa de Oliveira Lyra

X19234554

vanelyra@hotmail.com

EasyPub – Stock Management System

Technical Report



**Table of Contents**

[Executive Summary 3](#_Toc351559315)

[1 Introduction 3](#_Toc351559316)

[1.1 Background 3](#_Toc351559317)

[1.2 Aims 3](#_Toc351559318)

[1.3 Technologies 3](#_Toc351559319)

[1.4 Structure 3](#_Toc351559320)

[2 System 3](#_Toc351559321)

[2.1 Requirements 3](#_Toc351559322)

[2.1.1 Functional requirements 3](#_Toc351559323)

[2.1.2 Data requirements 3](#_Toc351559324)

[2.1.3 User requirements 3](#_Toc351559325)

[2.1.4 Environmental requirements 3](#_Toc351559326)

[2.1.5 Usability requirements 3](#_Toc351559327)

[2.2 Design and Architecture 3](#_Toc351559328)

[2.3 Implementation 3](#_Toc351559329)

[2.4 Testing 3](#_Toc351559330)

[2.5 Graphical User Interface (GUI) Layout 3](#_Toc351559331)

[2.6 Customer testing 3](#_Toc351559332)

[2.7 Evaluation 3](#_Toc351559333)

[3 Conclusions 3](#_Toc351559334)

[4 Further development or research 3](#_Toc351559335)

[5 References 3](#_Toc351559336)

[6 Appendix 3](#_Toc351559337)

[6.1 Project Proposal 3](#_Toc351559338)

[6.2 Project Plan 3](#_Toc351559339)

[6.3 Requirement Specification 3](#_Toc351559340)

[6.4 Monthly Journal 3](#_Toc351559341)

[6.5 Other Material Used 3](#_Toc351559342)

# Executive Summary

The main object of this project is to develop an application aiming independent or small chain pubs. The program allows the users to record sales, print receipts for customers, manage users, manage suppliers and products by adding new, editing or deleting them, record goods receipt, waste products or returned items. The system also allows the customer to generate sales and stock reports, the reports can be displayed on screen or exported to excel.

Users are divided in two categories: user and admin. The group defined as “user” has access to the point of sales area only and the “admin” group has total access to the application, this feature is determined by the “admin” when creating the new user.

In case someone forgets its password or simply needs to change it, it can be done via the forgot password link in the login area.

# Introduction

## Background

## What Is an Inventory Management System

importance

Inventory Management is of great importance for companies, no matter how small they are, it is one of the keys to a successful business, although it is not an easy task it “can be a complex task for any business to master. Trying to optimise inventory stock levels can seem like an endless battle. Sometimes you end up with too much or too little. This trial and error saga continues for businesses who don’t have the right inventory management systems in place. Small businesses often lag when it comes to setting up their inventory management systems properly. Historically, inventory management systems were big investments and reserved for large companies.”( [www.unleashedsoftware.com](http://www.unleashedsoftware.com)).

Having a stock management software brings many benefits to small companies such as:

* avoiding costs and lost revenue coming from overstocking;
* automates tasks leading to reduced work load and time saving for managers once all stock records are available at the click of a button;
* manual form filling is discarded, reducing human error;
* reduction of shortages;
* better purchases.

Another great, but not usually affordable options is to have a stock and point of sales integrated which will update stock in real time, giving more accurate projections lowering risks of spending on items that will take long to sell. A software can display sales history and products that need to be reordered.

## Aims

Application objectives:

* Allow users to add, edit and delete suppliers, products records;
* Create new user, change password;
* Generate sales and stock reports;
* Record sales in real time;
* Print receipt to customers;
* Store data in an online database in real time;
* Record goods receipt, waste and returned products.

The objective of this project is to deliver a point of sales and stock management together in one application, accessible to small pubs, also a user friendly program considering pubs usually have a big rotation of staff, it is important to have an application that can be easily and fast learned.

## Technologies

The technologies chosen to develop this projects were determined regarding my previous experience in college. The back end, to store all data on this application is MySQL and PHP localhost, to the front end I chose Java and Netbeans 12.3 IDE, libraries:

* JDK11;
* JCalendar 1.3.2;
* rs2xml;
* mySQL connector 8.0.25;
* POI 5.0.0;
* GitHub
* Cloud
* frames

## Definitions, Acronyms and Abbreviations

POS (point of sale)

JDK 11

JCalendar

R2xml

mySQL connector 8.0.25

POI 5.0.0

Jtextfield

joptionpane

## Use case diagrams

### Diagram Description automatically generated

### Requirement 1 <MAKE SALE>

#### Description & Priority

Using the sales section, the user inputs all the products and quantities sold to the customer and payment type. The screen will display the total amount of the purchase so the customer can be charged correctly. In case of any error, the transaction can be voided. This use case has high priority to the systems proper functioning.

#### Use Case

**Scope**

The scope of this use case is to allow the user to input every sale on the system.

**Description**

This use case describes the procedure to input sales and payment type and also void incorrect transactions.

**Use Case Diagram**

Diagram, letter

Description automatically generated

**Flow Description**

**Precondition**

The user already logged in and entered the sales module.

**Activation**

This use case starts when an user makes a sale.

**Main flow**

1. The user enters the sales module on the system
2. The user inputs the customer order
3. The user inputs the payment type
4. The system displays total amount of the order
5. The user finishes the sale <See E1>

**Exceptional flow**

E1 : <Incorrect sale input>

1. The user chooses the void feature
2. The user inputs the wrong product from the sale
3. The system voids the sale
4. The system returns to the sales module

**Termination**

The use case terminates when the user finishes the sale.

**Post condition**

Stocks and sales reports are updated.

### Requirement 2 <Register user>

#### Description & Priority

The admin user accesses the register module and creates a new user, inputting name and password. The system will generate automatically a numerical ID to the new user. The new user can choose to change the password or keep the one created by the admin. The priority for the system is low, once the admin has access to every feature on the system.

#### Use Case

**Scope**

The scope of this use case is to allow the user to create new users and change password.

**Description**

This use case describes the procedure to register new users on the system and how to change password when necessary.

**Use Case Diagram**

Diagram

Description automatically generated

**Flow Description**

**Precondition**

The system is in register user module.

**Activation**

This use case starts when an admin enters the register user module.

**Main flow**

1. The system identifies the admin account type
2. The user inputs name and password to the new user
3. The system creates a numerical login and displays it.
4. The admin informs the new user login details
5. The user enters the system using login and password<See A1>

**Alternate flow**

A1 : <Change password>

1. The user inputs login and password
2. The user clicks “Change password” button
3. System checks details
4. The system opens a screen to change password
5. User inputs old password and new password
6. User clicks “Change password” button
7. System check details and display change password message
8. System goes back to login area.

**Termination**

The system terminates when it displays the login info to the new user.

**Post condition**

The system goes into a wait state

### Requirement 3 <Maintain product>

#### Description & Priority

The admin will sign in and create, edit or delete products. High priority, requirement is vital to systems correct operation.

#### Use Case

**Scope**

The scope of this use case is to allow the user to create, edit or delete products from the system and also create products with a BOM.

**Description**

This use case describes the correct procedure to maintain products data up to date.

**Use Case Diagram**

Diagram

Description automatically generated**Flow Description**

**Precondition**

The user logged in using an admin account.

**Activation**

This use case starts when an admin goes to the products module.

**Main flow**

1. The user chooses the edition or creation of product
2. The system displays the correct form
3. The user created or edit product <See A2>
4. The system records the edition or creation
5. The system updated the database

**Alternate flow**

A2 : <Deletion of product>

1. The user will choose edition mode
2. The user finds product to be edited
3. The system displays product information
4. The user clicks the “Delete” button
5. The system updates the database

**Termination**

The system terminates when the user clicks Create, delete or edit buttons.

**Post condition**

The system goes into a wait state

### Requirement 4 <Maintain supplier>

#### Description & Priority

The admin will sign in and create, edit or delete suppliers. Low priority, at this stage of the project, suppliers are only for information.

#### Use Case

**Scope**

The scope of this use case is to allow the user to create, edit or delete suppliers from the system.

**Description**

This use case describes the correct procedure to maintain suppliers data up to date.

**Use Case Diagram**

Diagram

Description automatically generated

**Flow Description**

**Precondition**

The user logged in using an admin account.

**Activation**

This use case starts when an admin goes to the suppliers module.

**Main flow**

1. The user chooses the edition or creation of suppliers data
2. The system displays the correct form
3. The user created or edit supplier <See A3>
4. The system records the edition or creation
5. The system updates the database

**Alternate flow**

A3 : <Deletion of product>

1. The user will choose edition mode
2. The user finds supplier to be edited
3. The system displays supplier information
4. The user clicks the “Delete” button
5. The system updates the database

**Termination**

The system terminates when the user clicks Create, delete or edit buttons.

**Post condition**

The system goes into a wait state

### Requirement 5 <Maintain stock>

#### Description & Priority

The admin will sign in and create, edit stock data. High priority, requirement is vital to systems correct operation.

#### Use Case

**Scope**

The scope of this use case is to allow the user to create, edit stocks from the system. Also allows waste input.

**Description**

This use case describes the correct procedure to maintain stocks up to date.

**Use Case Diagram**

Diagram

Description automatically generated**Flow Description**

**Precondition**

The user logged in using an admin account.

**Activation**

This use case starts when an admin goes to the stock management module.

**Main flow**

1. The user chooses to edit stock data or add waste
2. The system displays the stock form
3. The user updates stock
4. The system records the change
5. The system updates the database

**Termination**

The system terminates when the user clicks save button.

**Post condition**

Stocks are updated.

# System

## Requirements

### Functional requirements

The functional requirements continued the same in general, the only aspect that changed in a certain way was the create report topic. The user should be able to generate reports from several aspects of the system, due to the short period to complete the project, only two reports are available, stock levels and sales records.

### Data requirements

Data is stored in a localhost in PHP, the user is able to analyse data through tables available, generate reports and export them to excel for other analysis and documentation. Admin users can manipulate date and interact with the whole system.

### User requirements

The main user requirements to a stock management system are:

* Reliability and security: the system has to be secure and deliver real-time information accurately.
* User-friendly: has to be easy to use and does not require extensive training
* Agility: provide fast and easy inventory control.
* Scalability: able to allow additional features requiring effortless configuration.
* Traceability: the user needs to be able to track its inventory.
* Analytics: the system must provide a analytics features and reports, this is a very important part of Stock Management.

### Usability requirements

The application design was developed based on usability. Pubs usually have a big rotation of staff all year around, so it was important to develop an easy to remember application, that would not require long training time, that could be intuitive and mainly not demanding.

### Recover requirement

Database

The Database can be easily exported from PHP and it is saved on the local server. It allows the user to access data without internet, in case any internet outages, the system will work accordingly it is more secure since it is in the companies device.

GitHub

Every change to the project is uploaded to GitHub, it stores and allows project management in a cloud, also it is possible to keep a version control of the project. Every change in the project was fetched to GitHub using GitHub Desktop, it detects the changes in your repository and the user upload it to the cloud.

Adress: https://github.com/vanelyra/EasyPub

Graphical user interface, text, application, email

Description automatically generated

*Github backup*

### Security requirement

Using a local server, makes it easier for the company to control access to it since the data is not transferred on the internet. The customer will need a antivirus and activate a firewall to ensure security.

## Design and Architecture

### Database

#### Design

Graphical user interface

Description automatically generatedThe user, executes the application and its commands on the client side via terminal, the server reads and interprets and executes the commands using the SQL language.

#### Structure and tables

Database and its tables

Graphical user interface

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

### Diagram, schematic Description automatically generatedSystem architecture

### Communication diagram

Diagram

Description automatically generated

This diagram presents the user interaction to the application. According to its classification – admin or user, he/she can interact with the whole appliction or only part of it.

Users classified as simply “user”, are allowed to: login, recover password. In case of sucesfull login, he/she will be redirected to a Main Page which will allow acess to the point of sales area.

Users classified as “admin”, after the login, will be redirected to a Main Page which grants acess to all application areas: Point of sales, Category, Others, Product, Stock, Suppliers. For safety reasons, only admins will be responsible for maintain data, except for sales, in the application.

### Design

The biggest objectives regarding design in this application were: keep it user-friendly and clean. The frames have the same design in green and white, there was not enough time to make the application look modern, I only simply changed the background and button colours.

The whole application follows the same design patterns as mentioned before. For user input, Jtextfields were adopted, the visual part of the data is displayed by tables, buttons were used to execute the applications queries, most warnings are displayed using JOptiopane, but some of them, used for validation purposes, are displayed in textfields. Some comboxbox`s were employed to retrieve data from one table of the database to another or to retrieve data from the database to textfields.

Graphical user interface

Description automatically generated

## Implementation

The application consists in 11 JFrames representing the different sections of the program and two classes that execute parallel tasks to the application correct functioning:

### Frames:

* Login: the user inputs login, password and user type to have your access granted or denied to the system. It also has access to the “Forgot Password” screen.
* Forgot password: The user needs to input username and answer a security question to recover his password.
* Main: after “admin” user login, this screen is displayed, it grants access to other pages on the application such as: Category, Others, Products, Sale, Stock, Suppliers.
* MainUser: after user type “user” login, this screen is displayed, it grants access the sales page only.
* Category: allow the user to add, edit and delete categories and also consult any information regarding this section in the table available.
* Others: this page is divided in two sections, sales and stock reports can be generate in this page, and also new users can be created.
* Products: allow the user to add, edit and delete products and also consult any information regarding this section in the table available.
* Receipt: displays the data representation of the sales recorded on sale screen.
* Sale: that is the point of sales of the application, the user inputs products and quantities and the system calculates the sale amount.
* Stock: where the user is allowed to update the stock, input any goods receipt, waste or transaction voids.
* Suppliers: allow the user to add, edit and delete suppliers and also consult any information regarding this section in the table available.

### Classes

* ConnectDB: this class connects holds the connection to the database.
* TextfieldLength: used for validation purposes, maximum character input in textfields

### Code snippets

#### ConnectDB class

Connection to database

Graphical user interface, text, application

Description automatically generated

Class ConnectDB, connects java to database, the JDBC driver is loaded than the driver manager connects using the database url, root and password.

#### TextFieldLength

Graphical user interface, text, application, email

Description automatically generated

This class validates the maximum number of characters allowed in a textField by extending PlainDocument. The number of characters will be defined in each frame by calling this function.

A screenshot of a computer

Description automatically generated with low confidence

That is the example of this function being called in a Jframe, the number of characters match the ones in the database.

#### Login

Checking users

Table

Description automatically generated with medium confidence

Checks if username or password are empty, if yes, a message appears to the user. Using a prepared statement, system verifies with database if user, password and user type are correct, in affirmative case, the user is redirected to the main area according to your type, if not, “login failed” message is displayed.

#### Forgot password

Recovering password

Graphical user interface, text

Description automatically generated

When user presses enter after typing username, the system checks against database if its exists, is yes, the security question is displayed to the user, if not, user not found message is displayed.

#### Category, products and suppliers

Constructor

Graphical user interface, text, application, email

Description automatically generated

Properties defined under constructor:

* Connection with database calling ConnectDB class;
* Frame cannot be resized by the user;
* When closed, frame is only hidden and do not close the system, system is terminated when user closes login or main page.
* Calling method to update Jtable with database data in real time.
* Method called to limit character in Jtextfields

Custom ID numbers

Text

Description automatically generated with low confidence

Method creates a custom ID numbers with letter and numbers retrieving the last – or maximum – ID registered in the database.

Populating JTable with database

Graphical user interface, text, application, email

Description automatically generated

This method populates the Jtable with database by using data from resultSet with DBUtils library.

Button clear

Graphical user interface, text

Description automatically generated

When this button is clicked, it clears all inputs in the JTextFields.

Button Add

Text

Description automatically generated with medium confidence

When the button Add is clicked, this method checks if any JtextField is empty, in affirmative case, a message will be displayed asking the user to input the information required, it checks if the ID field is filled, if it is, the new item cannot be added since the ID already exists. The query necessary for the task is insert and the JtextFields are matched to the right position to be added in the database. The method to update the database is called, a message of success in registering is display and the event matching the button clear is called to clear all data filled in the JtextFields.

Button Edit

Timeline

Description automatically generated with medium confidence

When button edit is clicked, method checks if the ID field is filled, if not, a message is displayed to the user, if it is, to query is executed updating the database with the text in the fields matched with its columns accordingly. The table is updated and the button cleat event is called.

Button delete

Timeline

Description automatically generated

When button edit is clicked, method checks if the ID field is filled, if not, a message is displayed to the user, if it is filled, the system displays a yes or no warning message, asking if the user really wants to delete data, if the user chooses YES, the system will delete the row matching the ID selected.

Selecting data from table

Text

Description automatically generated

The user can select the data to be deleted or edited by clicking the row corresponding to the item. The info from the table will populate the JTextFields according to the mapping inside the method.

Search field

Text

Description automatically generated

The user can find data present on Jtables easily by using the search field, every character inputted by the user triggers the search automatically.

JComboBox

Graphical user interface, text

Description automatically generated

Method populates JcomboBox with data from database according to the selected table and column.

#### Others

Button Generate

Text

Description automatically generated

This button populates the report table, if no dates are chosen, it displays all sales records, if the user selects the dates it will update the table accordingly.

Button clear

Text

Description automatically generated

This button is located in the sales report panel and clears info from the JDateChooser fields.

Button print

A picture containing table

Description automatically generated

This button exports data present in the Stock Report panel to excel.

Method Excel

Text, letter

Description automatically generated

Method allows user to choose the path to save the excel file generated.

#### Receipt

Graphical user interface

Description automatically generated with medium confidence Table

Description automatically generated

This code snippet shows the template of the sales receipt generated. It prints all data from the Jtable located in the sales Jframe.

#### Sale

Button Add

A picture containing text

Description automatically generated

When the button add is clicked, this method is called, it check if the quantity (mandatory field) is filled, in affirmative case, it populates the Jtable with product, price, quantity it also calculates the total (quantity x price) and fill JtextField and Jtable with the result. This method also checks if there is enough inventory to the item being sold.

Button Pay

Timeline

Description automatically generated with low confidence

Graphical user interface, text, application

Description automatically generated

This function is called when button pay is clicked, it updates multiple tables in the database: sale, sale\_item and the column quantity on the product table according to the serving size registered in the database.

Button print

Text

Description automatically generated

Method called when button print is clicked, calls the method on the Receipt frame and make it visible to the user.

Button remove

Text

Description automatically generated

Removes a selected row in the Jtable, recalculates total and updated total JTextField.

ComboBox feature

Graphical user interface, text, application

Description automatically generated

This combobox is editable, using “AutoCompleteDecorator.decorate(cbProduct);”, the user can find the required product by typing in the comboBox head.

#### Stock

Button update

Graphical user interface

Description automatically generated with low confidence

This code snippet shows the calculations to update stock, depending on the comboBox selection.

#### Validations

Email

Text

Description automatically generated

This is the email validation chosen. I tested many other options such as Regex, Apache commons validator but none of them worked properly. This one from Java mail does not works properly either, I accepts any input if there is a “@” and at a least three digits after. It does not validate domain or dot.

Text

Description automatically generated

Calling validation function when user types on JtextField and alerting user of incorrect input.

Text

Description automatically generated

Since Java Mail validation only alerts the user regarding the incorrect input, I added another function to make sure the wrong input does not go through. When the user selects some other component on the system, if the alert field still shows incorrect input message, the input will be cleared.

Graphical user interface, text

Description automatically generated

Case-insensitivity

Text

Description automatically generated

All fields containing letters (except email) were set to uppercase to make the system case insensitive.

Number validation

Graphical user interface, text, application

Description automatically generated

If the user tries to type any character that is not a number, the JtextField will be blocked, backspace and delete are also allowed so the user can correct any input mistakes.

Price validation

Graphical user interface, text, application, email

Description automatically generated

Only allows the user to input numbers, one dot, backspace and delete. Error message is displayed.

## Testing

Describe any testing tools, test plans and test specifications used in the project

## Graphical User Interface (GUI) Layout

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

Table

Description automatically generated

Graphical user interface

Description automatically generated

Provide screenshots of key screens and explain.

## Customer testing

Provide evidence for and results of customer testing. This may include ratings or quotes from the customer.

## Evaluation

How was the system evaluated and what are the results? In many cases this will include usage data and user feedback. It may also include performance evaluations, scalability, correctness, etc. depending on the focus of the project.

Quantative results may be reported in tables or figures. Note that tables have their caption above the table and need to be cross referenced in the text (see **Error! Reference source not found.**). In many cases, tables are better to read if you skip the vertical lines.

Table 1: Performance with and without caching

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nwithout** | **Nwith** | **Std.-Deviationwith** | **Std.-Deviationwithout** | **p** |
| Records | 100 | 200 | 2.54 | 3.97 | .002 |
| Data (GB) | 100 | 200 | 2.54 | 3.97 | .002 |
| Speed | 100 | 200 | 2.54 | 3.97 | .002 |

Figures have their caption below the figure as shown in **Error! Reference source not found.**. Make sure that if you use colour, the figure is still readable when printed in black & white, e.g., by using additional symbols, patterns, etc.



Figure 1: Learning gain across different experimental groups

# Conclusions

Describe the advantages/disadvantages, opportunities and limits of the project.

# Further development or research

With more resources, where could the results of this project lead to?

# References

The library provides a study guide on Harvard style referencing.

https://www.business.org/finance/inventory-management/why-is-inventory-management-important/

https://www.unleashedsoftware.com/blog/small-businesses-implement-inventory-management

<https://stackoverflow.com/questions/46326822/java-regex-first-name-validation>

<https://mailtrap.io/blog/java-email-validation/>

https://lucid.app/lucidchart/7a9b35a8-d03a-4e25-8651-859319110d9f/edit?beaconFlowId=5742BE28E167B437&page=0\_0#

# Appendix

## Project Proposal

## Project Plan

## Requirement Specification