YEROTH–ERP–3.0 Software–System Architecture

PR. XAVIER NOUMBISSI NOUNDOU

This document describes the thick–client software–system architecture of YEROTH–ERP–3.0. This document also explains the reasons for which we chose to design and implement YEROTH–ERP–3.0 as a thick–client software–system, as opposed to currently more popular web–browser–based software–system.

This document further demonstrates the superiority, in terms of simplicity, speed, maintenance, and low costs of development of thick–client software–system architectures over web–browser–based software–system architectures !

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *This introduction motivates why I created YEROTH–ERP–3.0, and why it uses the best software programming language of its time !*

1.1   Motivation

YEROTH–ERP–3.0 is an **Enterprise Resource Planing (ERP)** software–system that aims 'effectiveness' and 'simplicity', compared to other high ranked ERP software–systems (e.g.: 'Sage Gescom i7', 'SAP Business One', etc.).

We chose to design and implement YEROTH–ERP–3.0 as a thick–client software–system because of the following reasons:

1. The implementation language `C++` offers much flexibility:

   1. *MULTIPLE INHERITANCE:*

      It allows developers to abstract as much as possible business code upwards, away from downwards implementation classes. For instance, in YEROTH–ERP–3.0, GUI–Qt–windows inherits for instance *search filtering feature, and print capability from* 2 *different classes*.

      Print capability *couldn't be inherited from the same class where search filtering is abstracted and partially implemented (interface in* `Java` *for instance doesn't allow any method body code), because it works in its pure abstract class (*`C++` *class with at least* 1 *empty method body), together with feature* **database column filtering for viewing and printing.**

      The drawback of the multiple inheritance in `C++` *is: it sometimes can be very difficult to build it using "*`gcc (g++)` *[GCC]" !*

   2. *MACROS:*

      They enable developers to create TEXT TEMPLATE in their code.

      For instance, I use macros in some parts of my code to reduce execution time and stack activation records size for method or function calls in YEROTH–ERP–3.0 !

2. The availability of 'WHAT YOU SEE IS WHAT YOU GET' (WYSIWYG) tools for fast and useful user interface design (e.g.: `Qt designer` [Com20], `miniStudio (vxWorks)` [WEI20], etc.)

3. The low number of logical software–system architecture layer (i.e.: 2.) involved with the use of a thick–client software–system architecture, as opposed to a web–browser–based software–system (i.e.: 4, *client user interface*, *presentation layer*, *business logic*, and *data (DBMS)*).

Chapter 2

Thick−Client VS. Web−Browser−based Software−System Architecture

This comparison chapter tabular evaluates why thick−client are BETTER THAN web−browser−based software−system architectures !

2.1   Thick−client: 2 layers logical software architecture
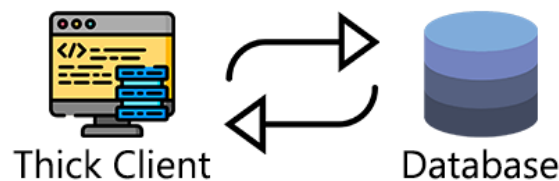
Figure 2.1: 2−layers logical architecture of thick−client software−system (Image copied from [sec20]).

Figure 2.1 illustrates an example of a thick−client software−system with a 2−layers logical architecture.

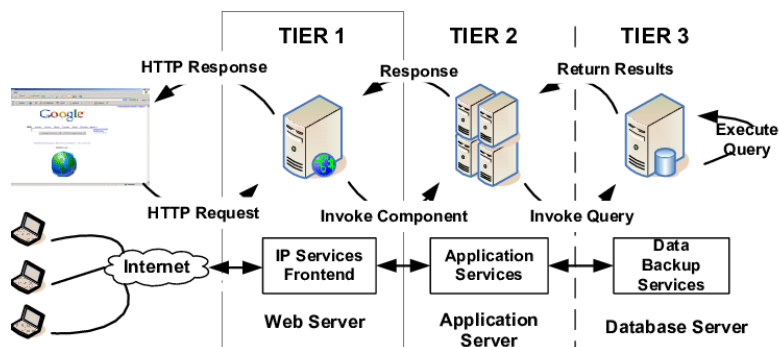## 2.2   Web−browser based: at least 4 layers logical software architecture

Figure 2.2: 4−layers logical architecture of web−browser−based software−system (Image copied from [KM06]).

Figure 2.2 illustrates an example of a web−browser−based software−system with a 3−layers logical architecture.

Table 2.1 compares thick–client software–systems against web–browser–based software–systems.

2.3   Tabular Comparison Between Thick−Client And Web Browser−based Architecture

	Thick−client application ✓	Web−browser−based application
business code	user interface	application server
co−related software−systems	1 (DBMS)	at least 3 (DBMS, web / application server)
number of logical layers	2 (client and data)	4 (client, presentation, logic, and data)
rapid prototyping (WYSIWYG tools)	yes	very limited
software security vulnerability	low (1 programming language)	VERY high (*several* programming languages)
user interface	all computers (GUI with *BUSINESS CODE*)	all computers (web−browser)

Table 2.1: Thick−client application VS Web−browser−based application.

Table 2.1 illustrates the advantages of thick−client software−system architecture over web−browser−based software−system architecture !

The common argument for web−browser−based software−system architecture is you update the business code just at 1 place: *the application server* !

I argue that thick−client architecture IS JUST AS WELL BEST UPDATED AT 1 PLACE: *the user's computer*.

FOR INSTANCE, UPDATE OR UPGRADE OF ENTERPRISE SOFTWARE−SYSTEM WEB−BROWSER−BASED REQUIRES ALLMOST AT LEAST 1−24−HOURS SHUTDOWN OF ALL INVOLVED WEB (apache tomcat, etc.) APPLICATIONS SERVERS.

The issue of automatic software upgrade in a computer network is best solved by the 'apt upgrade software−system of `Debian-Linux`', as an example !

Chapter 3

The Thick–Client Software–System Architecture of YEROTH–ERP–3.0

This chapter explains why YEROTH–ERP–3.0 is modular in its uses, and fits any industrial setting !

	Thick–client application ✓	Web–browser–based application
business code	user interface	application server
co–related software–systems	1 (DBMS)	at least 3 (DBMS, web / application server)
number of logical layers	2 (client and data)	4 (client, presentation, logic, and data)
rapid prototyping (WYSIWYG tools)	yes	very limited
software security vulnerability	low (1 programming language)	high (*several* programming languages)
user interface	all computers (GUI with *BUSINESS CODE*)	all computers (web–browser)

Table 3.1: Thick–client application VS Web–browser–based application.

3.1   Business and user interface code deployment

Table 3.1 depicts the issue of business and user interface code deployment on all computers participating in the functioning of YEROTH–ERP–3.0, as a software–system for a user.

We tackle the problem of automatic deployment of business and user interface code on all user computers by using the 'apt upgrade' software–system on 'Debian-Linux'.

3.2   Databases

DBMS MySQL is used for storing and managing huge data accross globe.

3.3   NUMBER OF LOGICAL SOFTWARE–SYSTEM LAYERS: E.G. Sample technical configurations

This section illustrates 2 different possible computer network configurations that could prevail in industry.

3.3.1　Sample 2–computers store
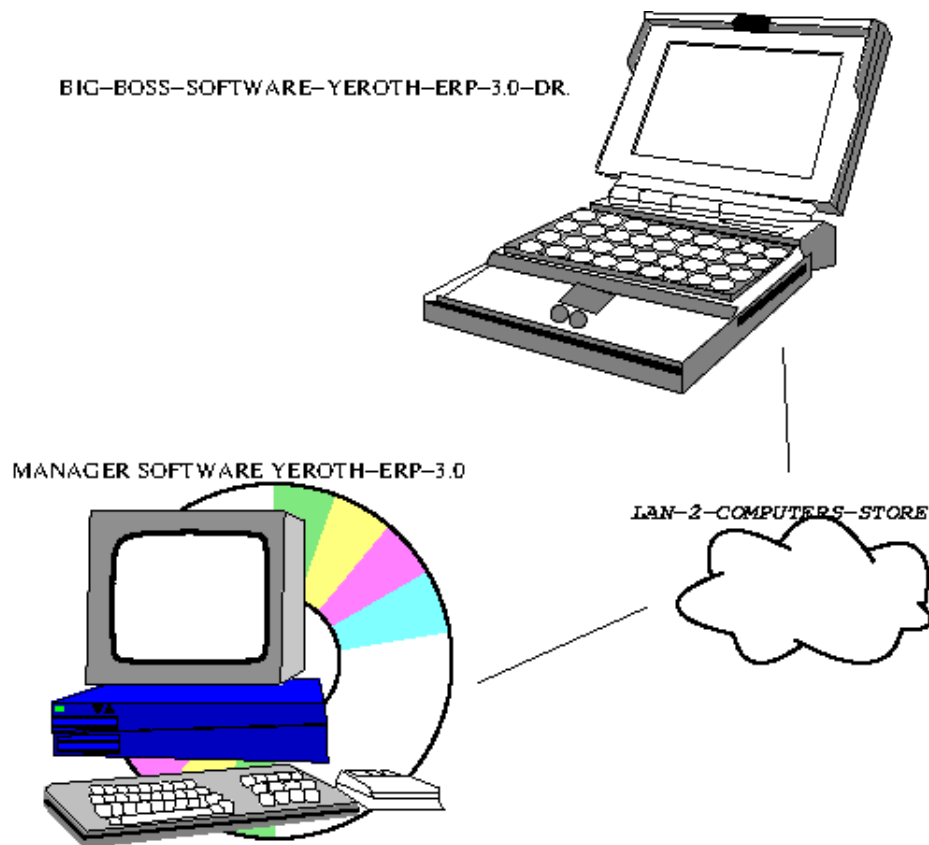
Figure 3.1: Sample 2–computers store.

3.3.2　Sample decentralized multi sites supermarket

YEROTH_{r&c}

YEROTH_{r&c}

FIGURE 2 – CONFIGURATION DE BASE SIMPLIFIÉE DE ETS-KENFACK (1 MACHINE MANAGER).

author: DR. XAVIER NOUMBISSI NOUNDOU　　　　　Version of − 22 décembre 2020 −

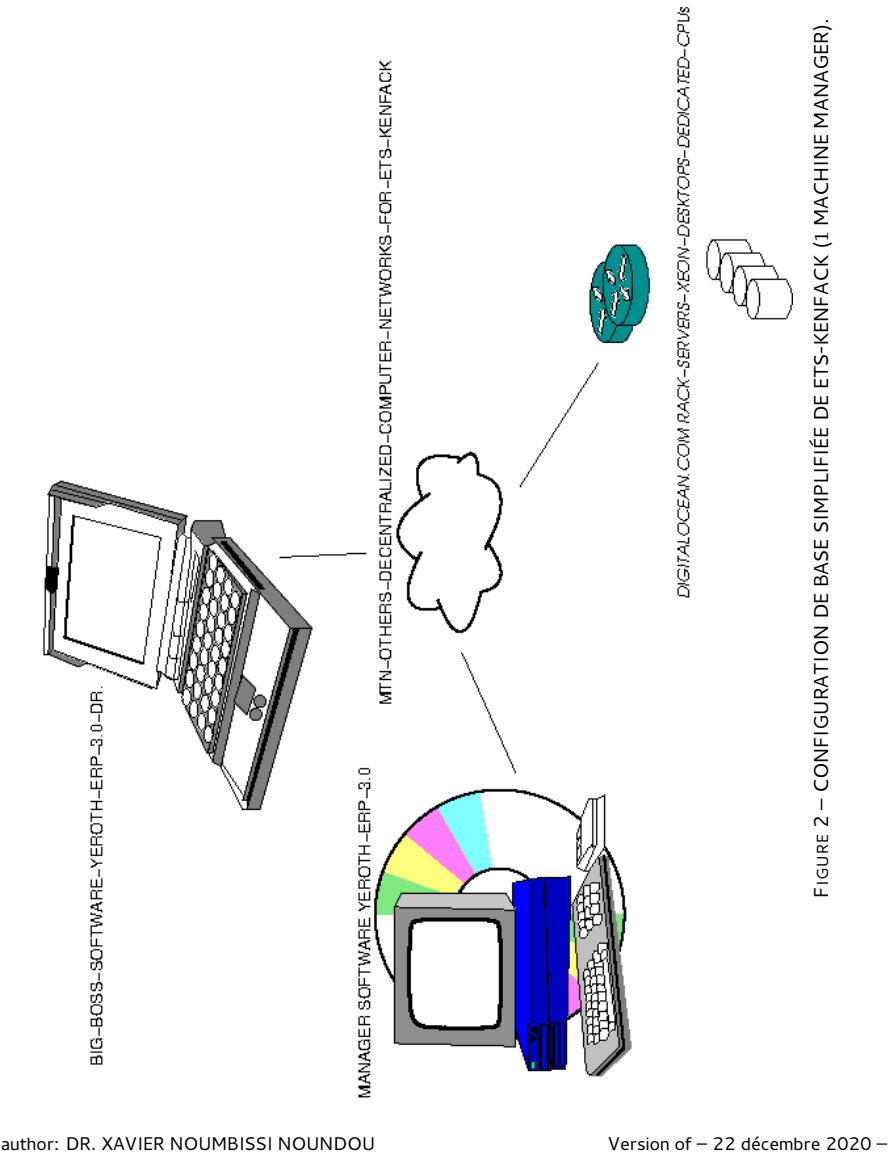author: DR. XAVIER NOUMBISSI NOUNDOU　　　　　Version of − 22 décembre 2020 −

Figure 3.2: Sample decentralized multi sites supermarket.

3.4   Software Security Vulnerability in YEROTH–ERP–3.0

I perform wecurity vulnerability detection for open source software YEROTH–ERP–3.0 by using memory leak detection tool 'valgrind' (ct. []).

Chapter 4

Conclusion

This conclusion explains why YEROTH—ERP—3.0 uses the BEST SOFTWARE TECHNOLOGY IN TERMS OF SOFTWARE—SYSTEM ARCHITECTURE !

	YEROTH—ERP—3.0	Odoo
libraries & programs	lxqt-sudo, etc.	python-lxml, etc.
user interface code TOOLS WYSIWYG	QT—DESIGNER	(CUSTOM BUILD) FRAMEWORKS
business code	C++	Python, JavaScript, XML
DBMS	MySQL	PostgreSQL
web—server		Werkzeug

Table 4.1: YEROTH—ERP—3.0 VS. Odoo.

YEROTH—ERP—3.0 has a thick—client software—system architecture because we found thick—client software—system architectures simpler than web—browser—based software—system architectures.

Thick—client software—system architectures is simpler because it requires less layers in its logical (or physical) software—system architecture, and is easier to develop and maintain as a software—system application.

Table 2.1 illustrates a thick—client software—system is SUPERIOR IN TERMS OF TOOLS FOR MAINTENANCE AND DEVELOPMENT than a web—browser—based software—system !

A web—browser—based software—system architecture has more drawbacks as follows:

1. it requires at least 2 other software—systems, *apart from the ones normally required by developed software—system itself, for instance libraries (e.g.:* Log4j*),* to fully operate (e.g.: web server, application server, etc.).

   Table 4.1 depicts this situation in the light of the open source ERP software—system Odoo.

   Accordingly, a thick—client software—system doesn't require any running and managing infrastructure such as for example an application server !

2. A web—browser—based software—system requires at least 4 layers in its logical system architecture (e.g.: client, presentation, logic, and data layers).

   Accordingly, a thick—client software—system only requires at least 2 layers !

3. A web—browser—based software—system potentially entails more software security vulnerabilities because its implementation requires the use of at least 2 different programming languages, and frameworks in combination.

   Accordingly, a thick—client software—system needs only the use of 1 homogeneous software programming language !

Chapter 5

Bibliography

[Com20] The Qt Company. Qt Designer Manual. http://doc.qt.io/qt-5/qtdesigner-manual.html, 2020. Last accessed on September 4, 2020 at 15:21.

[GCC]   THE COMPILER SUITE GCC. THE GCC (G++) COMPILER SUITE. http://www..org. Last accessed on December 29, 2020 at 12:00.

[KM06]   Taeho Kgil and Trevor Mudge. Flashcache: A nand flash memory file cache for low power web servers. In *Proceedings of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, CASES '06, page 103–112, New York, NY, USA, 2006. Association for Computing Machinery.

[sec20]   securityboulevard.com.   Thick   Client   Penetration   Testing   Methodology.   http://securityboulevard.com/2020/02/thick-client-penetration-testing-methodology/, 2020.   Last accessed on September 4, 2020 at 15:21.

[WEI20] Yongming WEI. miniStudio User's Guide. http://www.minigui.net/en/ministudio, 2020. Last accessed on September 4, 2020 at 15:21.

Index

Appendix A

Presentation Documents of open source software–system YEROTH–ERP–3.0

YEROTHr&d

YEROTH–ERP–3.0 Software–System Product Sheet

YEROTH–ERP–3.0 is an **ERP software–system** with 6 **user roles, and types**:

1. « Administrator »
2. « Business manager »
3. « Cashier »
4. « Seller »
5. « Stock manager »
6. « Storekeeper ».

YEROTH–ERP–3.0 **features**:

1. alerts over stock quantity, and, time period
2. business dashboard
3. customer relationship management (CRM)
4. sale management (e.g. point–of–sale)
5. stock management (e.g. check in)
6. user, and role administration.

YEROTH–ERP–3.0 **is**:

1. easier, and, intuitiver, in its use
2. lighter, and, faster, in memory usage
3. multi sites (enabled).

YEROTH–ERP–3.0's runtime memory usage test is realized using software analysis tool **valgrind**.
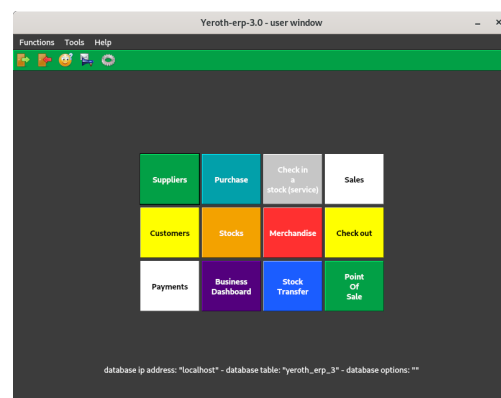
Business manager's main window
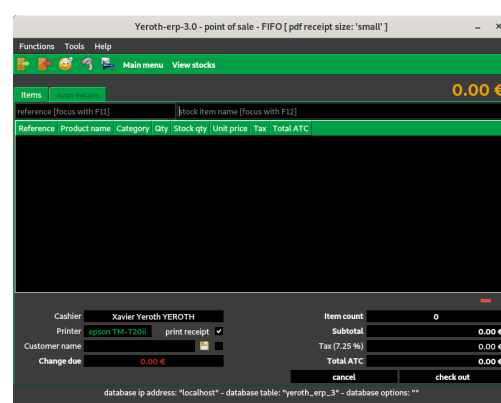
Cashier's main window

OPERATIONS

Point–of–Sale Hardware

✓ Barcode scanner
✓ Thermal printer, etc.

Database Management Systems

✓ MySQL

Operating Systems

✓ Debian–Linux

Advantages of YEROTH–ERP–3.0 Compared to other top ERP Software–Systems

YEROTH–ERP–3.0 is a very easy to use ERP (Enterprise Resource Planing) software–system because of its characteristics:

1. separate views for each user role
2. complete and fundamental training in 5 days
3. easy to use graphical user interface (GUI)
4. no college or university training needed
5. no formal business training needed
6. no financial accounting training needed
7. no internet connection needed.

Stock listing window

Table 1 pictures the 'effectiveness' and 'simplicity' of YEROTH–ERP–3.0, compared to other top ERP software–systems "Sage Gescom i7", and "SAP Business One".

	YEROTH–ERP–3.0	Sage Gescom i7	SAP Business One
separate views per user role	YES	NO	YES
complete training (or solution)	5 days	at least 2 months	at least 3 months
difficulty in navigation	very easy	difficult	very difficult
usage language in software	easy everyday English	simple	technical
financial accounting knowledge	no	no	useful
advanced marketing knowledge	no	useful	useful
internet connection	optional	mandatory	optional

Table 1: Comparison between YEROTH–ERP–3.0 and 2 top tier–1 full featured ERP software–systems

OPERATIONS

Point–of–Sale Hardware

✓ Barcode scanner
✓ Thermal printer, etc.

Database Management Systems

✓ MySQL

Operating Systems

✓ Debian–Linux

YEROTH–ERP–3.0 Point–Of–Sale Recommended Hardware

1 Barcode Scanner

We recommend, but not exclusively, the use of barcode scanner: **" Xfox FJ–5 USB Plug and Play Automatic Barcode Scanner "** (approx. 17 €).

Barcode Scanner

2 Thermal Printer

We recommend, but not exclusively, the use of thermal printer: **" Epson TM–T20ii Point of Sale Thermal Printer "** (approx. 100 €).

Thermal Printer

3 Cash Drawer

We recommend, but not exclusively, the use of cash drawer: **" HP QT457AT "** (approx. 90 €).

Cash Drawer

4 Touch Screen Monitor

We recommend, but not exclusively, the use of touch screen monitor: **" ASUS 15.6" LCD Monitor (VT168H) "** (approx. 155 €).

Touchscreen Monitor

5 Computer

We recommend, but not exclusively, the use of desktop computer: **" Lenovo Thinkcentre M720 Small Form Factor (SFF) "** (approx. 450 €).

Computer

Information Brochure of the ERP software–system YEROTH–ERP–3.0

DR. XAVIER NOUMBISSI NOUNDOU

Tasks	« Business manager »	« Seller »	« Stock manager »	« Storekeeper »	« Cashier »
insert stock (or service)	✓	✓ (SERVICE)	✓ (STOCK)		
delete stock	✓				
view stock	✓	✓	✓	✓	✓
modify stock	✓		✓		
transfer stock	✓		✓	✓	
modify stock management strategy (e.g.: « FIFO », etc.)	✓	✓ (NO PERMANENT)	✓ (NO PERMANENT)		
point–of–sale	✓	✓			✓
view stock transfers	✓		✓	✓	
purchase management	✓	✓	✓ (PARTIAL)		
supplier management	✓	✓			
customer relationship management (CRM)	✓	✓			
business dashboard	✓				
sale return	✓				
view sales information	✓	✓ (SELF)			

Table 1: YEROTH–ERP–3.0 functions–tasks, and associated users–roles.
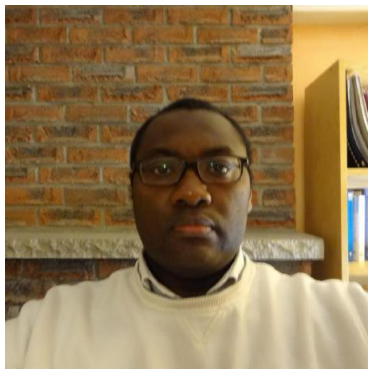
1 Developer Biography

Figure 1: Portrait of DR. XAVIER.

DR. XAVIER NOUMBISSI NOUNDOU is a CHRISTIAN BY FAITH, Cameroonian, born on September 16 1983 in DOUALA (LITTORAL region, CAMEROON).

Xavier has a *"Diplom–Informatiker (Dipl.–Inf.)"* qualification from the **University of Bremen, Bremen, Bremen, GER-MANY** (May 25, 2007).

Xavier is a *PH.D. in Software Engineering* (software construction, and testing) since November 18, 2020 because of his academic research, and professional engineering contributions as follows:

1. 'Context-Sensitive Staged Static Taint Analysis For C using LLVM'

   1. source code:
      http://github.com/xnoumbissinoundou/yeroth-saint
   2. full text (published on July 1st, 2015): http://archive.org/details/saint_201507.

2. 'YEROTH-ERP-3.0':

   1. source code:
      http://github.com/xnoumbissinoundou/yeroth-erp-3.0
   2. full text (ongoing publication): http://archive.org/details/yeroth-erp-3-0-info-english.

2 Introduction

YEROTH–ERP–3.0 is an **Enterprise Resource Planing** (ERP) software–system.

Users of YEROTH–ERP–3.0 could have the following roles:

1. « Administrator »
2. « Business manager »
3. « Cashier »
4. « Seller »
5. « Stock manager »
6. « Storekeeper ».

YEROTH–ERP–3.0 allows for business management tasks listed in Table 1, depending on user role.

3   Advantages   of   YEROTH–ERP–3.0

1. YEROTH–ERP–3.0 is 100% stable

2. YEROTH–ERP–3.0 has an alert system with two types of alerts: alerts based on stock–quantity, and time–period alerts

3. users have the choice between small size receipts, and, bigger size receipts ("A4")

4. YEROTH–ERP–3.0 runs on the Linux operating system, because Linux is stable, performant, and less vulnerable to security breaches in comparison to other operating systems ('`Windows 10`')

5. YEROTH–ERP–3.0 has an user interface "Sales" to view sale information (Figure 2), and thus enables users to make managerial decisions

6. YEROTH–ERP–3.0 has an interface "Business dashboard" that generates financial accounting reports, from sale and payment information, to help managers to make "business decisions".

Figure 2: Sale–information window.

4   Alert System

Users with roles « Administrator » or « Business manager » are the ones able to create alerts.

YEROTH–ERP–3.0 allows its users to create two types of alerts:

1. alerts over stocks–quantities
2. alerts over time intervals (this helps for perissable articles and for sales discounts over a period of time).

4.1   Alerts over Stock–Quantity

An alert aver a stock–quantity is a message that is sent to a pre–determined user whenever "pre–determined" stock–quantity (X) of a specific article–stock is reached.

For instance, Xavier (« Business manager ») could create an alert for stock "mango" that will be trigerred whenever stock "mango" quantity reaches 100; An alert–message is sent to user John (« Storekeeper »).

4.2   Alerts over Time–Period

A time–period is defined by a starting–date and an ending–date (dates are from the "gregorian" calendar).

An alert aver a time–period (T) is a message that is generated, sent to a pre–determined user, and kept within YEROTH–ERP–3.0 from T's starting–date up to T's ending–date.

For example, an alert with a message has to be sent to Paul (« Cashier ») when the date of May 05th is reached. The alert message specifies that a rebate of 20% has to be applied on every sale of yoghourt 'trèsbon' during a time interval of 2 weeks.

5   Database   Management   System

YEROTH–ERP–3.0 uses 'MariaDB' as the standard DBMS. 'MariaDB' is very stable, very performant, and free–software.

6   Conclusion

Figure 3: Point–of–sale window.
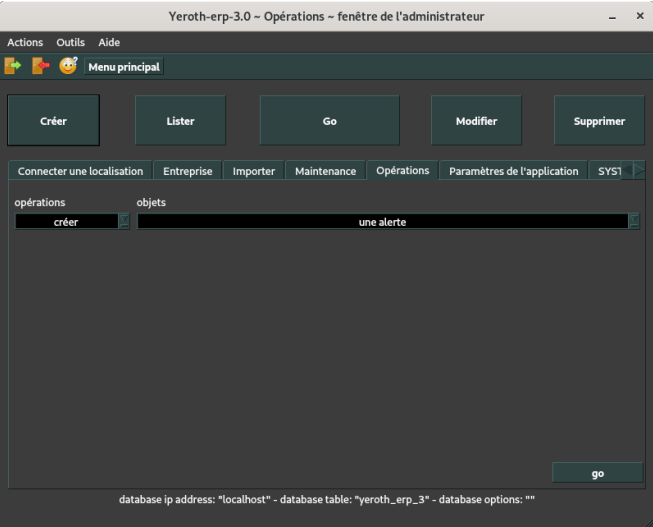
Figure 3 illustrates the window for selling articles.

Figure 4: Administrative window for business manager.

Figure 4 illustrates the administrative window for business managers.