

Mémoire de fin d'études

en vue de l'obtention du titre de

Bachelier en Informatique de Gestion

Année académique 2014-2015

Conception and implementation of an e-commerce module by using WebObjects

Pyramide Ingénierie

Rue de Liège, 188

B-4800 Verviers



Présenté par

Pierre NYSSSEN

Date de la défense

Juin 2015



Mémoire de fin d'études

en vue de l'obtention du titre de

Bachelier en Informatique de Gestion

Année académique 2014-2015

Conception and implementation of an e-commerce module by using WebObjects

Pyramide Ingénierie

Rue de Liège, 188

B-4800 Verviers



Présenté par

Pierre NYSEN

Date de la défense

June 2015



I want to thank all those who participated in any way in my internship and the writing of this memoir.

In particular, I would like to thank my supervisor, Mr. Bruno Wiesen, developer at Pyramide Ingénierie, for his availability and his advice that enabled me to progress continuously during those four months of internship.

I wish to express my gratitude to Mr. Luc Demarche, manager of Pyramid Engineering, who welcomed me into his team and allowed me to undertake this internship.

My thanks go to Mr. Louis Swinnen, Professor at the Haute Ecole Libre Mosane, for his valuable advice on writing this memoir.

I would also like to express my gratitude to the entire team of cipiyou for their support and encouragement which fostered my integration in the company.

Summary

Summary.....	5
Introduction	6
PART I: WebObjects & cipyou-Account	8
Chapter 1: WebObjects	9
1.1 History of WebObjects	9
1.2 Database	9
1.3 Application WebObjects.....	10
1.4 A WebObjects application	11
Chapter 2: cipyou-Account.....	15
2.1 Model	16
2.2 View.....	18
2.3 Controller and features	20
2.4 Deployment.....	22
2.5 PDF generation.....	23
2.6 Future of the application cipyou-Account.....	26
Conclusion Part I.....	27
PART II: Order Processing System.....	28
Introduction.....	29
Chapter 3: Delivery.....	32
3.1 Changing the framework cipyou-web-manage.....	33
3.2 Views of the delivery selection procedure	36
3.3 Conclusion.....	39
Chapter 4: Discount coupons.....	40
4.1 Database	41
4.2 cipyou-web.....	45
4.3 cipyou-manage	48
Chapter 5: Selecting a payment solution.....	52
5.1 Comparison criteria.....	53
5.2 Comparison.....	55
5.3 Choosing HiPay.....	58
Chapter 6: Integration of the solution “HiPay Direct”	60
6.1 Terms and conditions.....	61

6.2 Technical modifications.....	63
Conclusion - Part II.....	69
Conclusion.....	71
Bibliography.....	72
Attachments.....	A
1: Tariff Grid for parcel delivery from Belgium by GLS.....	B
2: Terms and conditions of Pyramide Ingénierie (in French).....	C
Table of figures.....	74

Introduction

Company Overview

My internship took place at Pyramide Ingénierie.

The company was founded in 1989. They began by programming on the Apple Macintosh. The first orders have steered the development of the company towards the use of interactive databases and complex geometry.

In 1995, Pyramide Ingénierie was the first company to offer internet hosting services in the region under the label “Datagram”.

In 1998, Pyramide Ingénierie got a new denomination (“Mac in’Shop”) to better identify its specialization in Apple Macintosh solutions. Mac in’Shop quickly became an “Apple Authorized Reseller” and, in 2003, was one to top 20 Apple resellers in Belgium.

In 2007, Mac in’Shop was renamed “cipiyou”, became an “Apple Premium Reseller” and moved to a new location.

In 2012, all Pyramide Ingénierie services are centralized at this new address. As well a training centre (cipiyou-school) was launched.

In 2014, Pyramide Ingénierie celebrated 25 years in business.

Tools

I worked on an iMac with OS X Yosemite, the eleventh and latest version of Apple’s operating system.

The development was done on WOLips. It’s an open-source version of Edipse (Juno), from Project Wonder. It facilitates the development of WebObjects applications.

I used several PostgreSQL databases.

Internship objective

During my internship, I had to design and develop an order processing system for the website cipiyou.

To do this, I first learned to use WebObjects in which I used different tools. To practice these new skills, I created the cipiyou-Account application. This enabled me to test the various tools I would use afterwards.

I was then ready to focus on the development of the order processing system for the cipiyou store.

The order processing system is separated into two parts. The first one concerns the choice of a delivery method and the second one the payment of the order. The delivery method affects the payment part as shipping costs must also be included in the total amount payable.

Document structure

This document is separated into three parts.

The first part focuses on WebObjects basics (Chapter 1) and on the cipiyou-Account application (Chapter 2).

The second part presents the ordering process. It talks about delivery choice (Chapter 3), the implementation of a system of discount coupons to use on the website (Chapter 4) before concentrating in the payment process with the choice of a payment solution (Chapter 5) and its integration (Chapter 6)

In this document, “He” refers to either “He” or “She”.

PART I: WebObjects & cipyoun-Account

I will start with the basics of WebObjects, focusing on concepts used in the various phases of development. I will then discuss the cipyoun-Account application, which is an application designed to take over the tools and WebObjects.

Chapter 1: WebObjects

1.1 History of WebObjects

WebObjects allows the creation of dynamic websites. Released in 1996, it is one of the first frameworks for web application. The development language is Objective C.

WebObjects was originally developed by NeXT. When Apple bought NeXT in 1997, the development was taken over by Apple Inc.

After the fourth version, Apple began to migrate from Objective-C to java. Compatibility with Objective-C is deleted in the fifth version. The current version is 5.4.3, available since September 2008.

Project Wonder is an open-source set of extensions, frameworks and WebObjects applications. The project increases the performance and possibilities of WebObjects.

1.2 Database

Enterprise Object Framework is an ORM (Object-Relational Mapping) integrated into WebObjects.

EOModeler

Enterprise Object provides a tool called EOModeler. This tool allows the developer to create a database by creating entities, their attributes and the relationships between these entities.

EOModeler automatically generates SQL code to create the tables. It can access to the database to run SQL automatically.

EOModeler also generates Java classes representing objects of the different tables.

EditingContext

Enterprise Object optimizes access to the database. To do so, it uses an EditingContext object. The EditingContext is a temporary picture of the database.

When the application accesses data present in the EditingContext, it is not reloaded from the database. If it is not present in the EditingContext, data is loaded from the database and is added to the EditingContext.

Data is maintained for some time in the EditingContext (twenty minutes by default). If it is not accessed during this time, it will be refreshed when accessed.

If data is retrieved from a database and that data is related to other objects, these objects are pre-loaded, which means that objects are created but not instantiated.

Modifications of the EditingContext are temporarily kept in a list of changes. When involving the "saveChanges()" method, changes are written to the database. With several applications accessing the same database, concurrent access can occur. A locking system must be put in place to prevent concurrent modifications. WebObjects offers two systems, an optimistic locking (which allows simultaneous access, the latest modification being the one saved) and a pessimistic locking (that prevents simultaneous access).

1.3 Application WebObjects

A web application generates dynamic web pages based on HTML. WebObjects applications are object-oriented, written in java and are deployed on a WebObjects application server. The application generates web pages by instantiating objects called “web components”.

A “web component” is the combination of the java class and the WOComponent.

A WebObjects application is event-driven, in which events are received from the application server. WebObjects also refers to the type of the application server. It is the first type of application server.

Figure 1 shows the architecture of a WebObjects website.

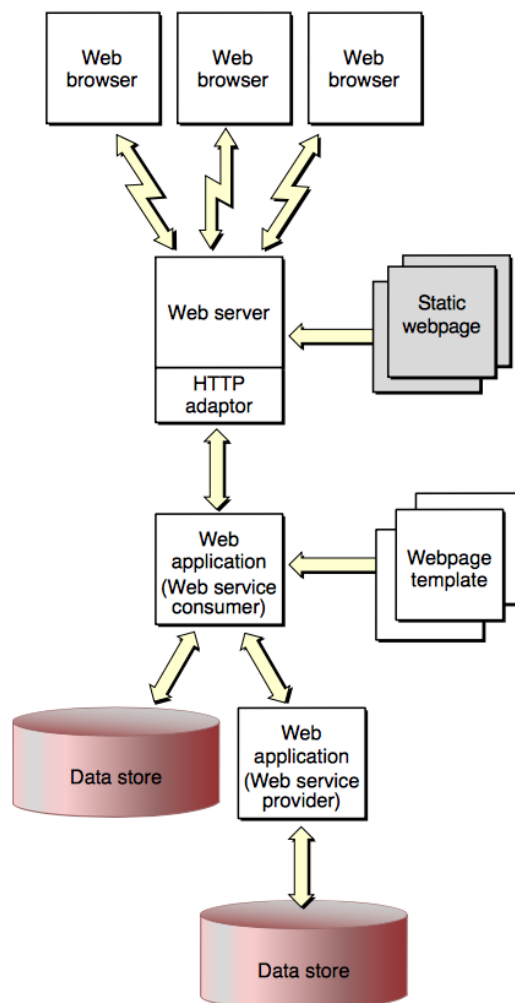


Figure 1 – Diagram of a WebObjects website

The request comes from a web browser (the query is a URL). The web server detects that the request must be processed by a WebObjects application and forwards the request to an HTTP adaptor. This adaptor translates the request so that the WebObjects application can understand it. The WebObjects application generates a webpage based on web components and data from the database (or from the EditingContext). The application transmits that page to the web browser via the HTTP adaptor. This cycle is called the request-response loop.

1.4 A WebObjects application

The project of a WebObjects application is separated into several parts. The main ones are:

- WOComponents;
- Sources;
- Resources.

Figure 2 shows the content of the cipyou-Account application.



Figure 2 – Content of the cipyou-Account application

1.4.1 WOComponents

WOComponents is used to describe a window of the WebObjects application. They are the visible part of the application. They are comprised of two files: an HTML file and a WOD file.

HTML

HTML files include <webobject> tags. These can be of many different types, depending on the data they contain. For example, WOString for strings, WOConditional to add display conditions to the page ...

WOD (WebObjects Declaration)

Each <webobject> tag is related to java code. The WOD file is the file that contains links between <webobject> tags and the java code. For example, a WOString tag takes a “value” parameter which can be a variable or a method in the java code.

Operation

When the WebObjects application generates a page that contains <webobject> tags, they are interpreted. The application will run the java code related to these tags through the WOD file. Once all tags are interpreted, the page will no longer contain <webobject> tags. The user never sees WebObjects code.

Figure 3 shows a portion of the web component called “Affichage”. It shows the HTML file on the top and the WOD file on the bottom.

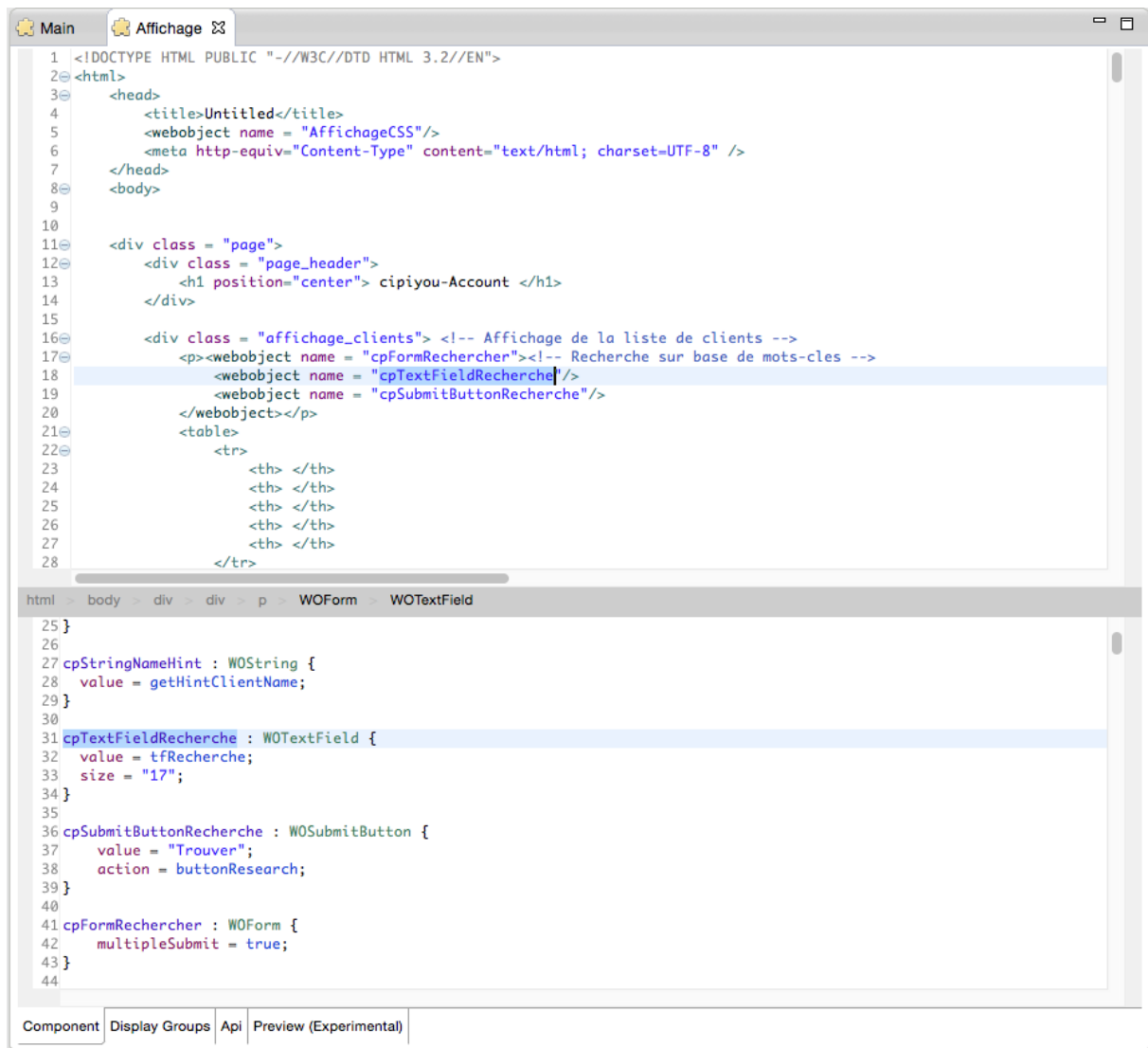


Figure 3 – View of the WComponent “Affichage” from the cipiyou-Account application

The <webobject> “cpTextFieldRecherche” is highlighted. It is visible in the HTML code and in the WOD file. When it is interpreted by the web application, it will be replaced by the content of the variable “tfRecherche”.

1.4.2 Sources

Figure 4 shows the content of the folder “Sources” of the cipiyou-Account application.

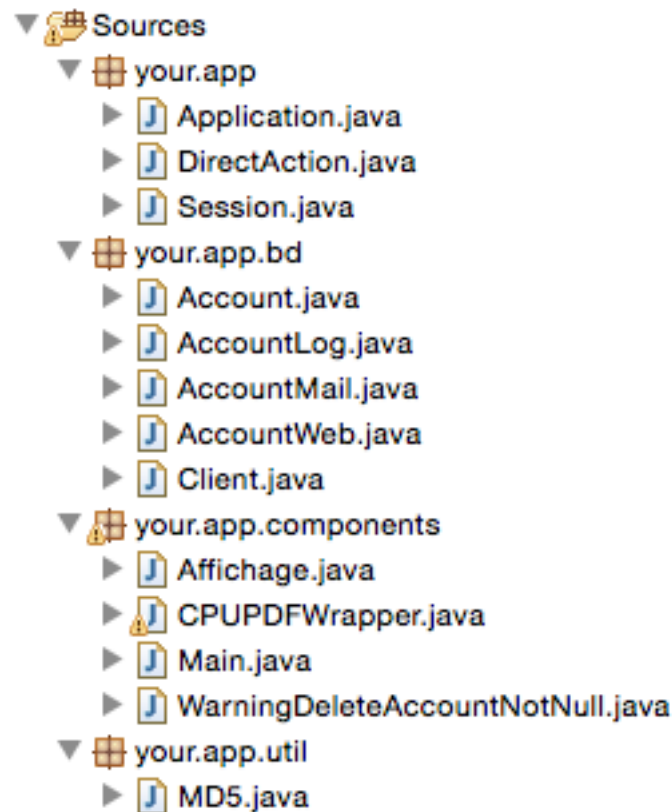


Figure 4 – Content of the folder “Sources” of the cipiyou-Account application

Session.java

WebObjects simplifies the management of user sessions. Each project contains a class called “Session.java”. Instances of this class are maintained on the server until expiration.

To keep the session between the different views, it is possible to put its id in the URL or to store it in a cookie.

If they contain the session id, URLs are not bookmarkable. The URL depends on the session id and it changes after expiration.

It is possible to store the session in a cookie. This makes the URL bookmarkable. This may seem trivial, but it becomes important in Chapter 6 with the online payment.

DirectAction.java

HTTP requests sent by the client web browser may be URLs. The DirectAction.java class contains all methods callable with URLs.

Application.java

This file is the entry point of the application.

Component.java

When a WOComponent is created, a java file corresponding to it is also created. It contains methods and variables related to this WOComponent within its WOD file.

Entities

The folder Entities (called “your.app.bd” in Figure 4) contains the classes of the ORM Enterprise Object Framework.

The different classes may be generated through EOGenerator. This creates two classes for each table. The first one is abstract and is designed to be modified by the developer while the second contains methods generated by the EOGenerator to manipulate objects.

These methods are automatically generated and used to access the values of the objects as well as all the other objects related to those objects.

1.4.3 Resources

Several files are placed in the Resources folder. When deploying a new version of the application, resources are deployed independently of the rest of the application.

Properties

This file contains all the project properties.

.jasper & .jrxml

These two files are used for PDF generation.

Images

Images can be placed in that folder.

CSS

It is possible to include a CSS file to a WOComponent with a <webobject> tag called “ERXStyleSheet”.

Chapter 2: cipyou-Account

To take over WebObjects and the different tools, I developed an application called cipyou-Account.

This application allows the user to administer (view, add and delete) the different clients related to Pyramide Ingénierie and their accounts. Each client has a customer account. A customer account contains user accounts. These user accounts can be of three types.

The different types of accounts are:

Mail: Pyramide Ingénierie provides email addresses and hosts email servers;

Software: Pyramide Ingénierie has several programs (including “MacBuro” and “Proportions”) and sells licenses to its customers;

Server: Pyramide Ingénierie has web servers and allows its customers to host their website. Server-based accounts may include an email address.

Current application

There is already a standard application to manage accounts. However, it is no longer maintained and is no longer compatible with current versions of Mac OS X. A device is maintained on an older version of the operating system to continue to run this application.

The application does not use a database but a file to store data.

New implementation

The new application is made in WebObjects. It is thus a web application. This application may be placed on the server and accessed from any computer. The device used to maintain the old application will be upgraded.

Data will be stored in a database instead in a file.

Development approach

I created the project step by step. First, I realized a draft of the main view to place features. Then, I created features one by one. I could use the view to test the features and I had access to the database to verify data.

Once all features were correctly implemented, I produced a second version of the view. This version was submitted to the users who gave me feedback on changes and adaptations to be made, as well as ideas for modifications. This step was repeated several times until the application was appropriate.

In the rest of this chapter, I will discuss the data model developed for the application. I will continue by detailing the views, approaching controllers and features before approaching deployment of a WebObjects application, generation of PDF file and the future of cipyou-Account.

2.1 Model

The database is divided into five tables, shown in Figure 5. These tables contain all of the application data.

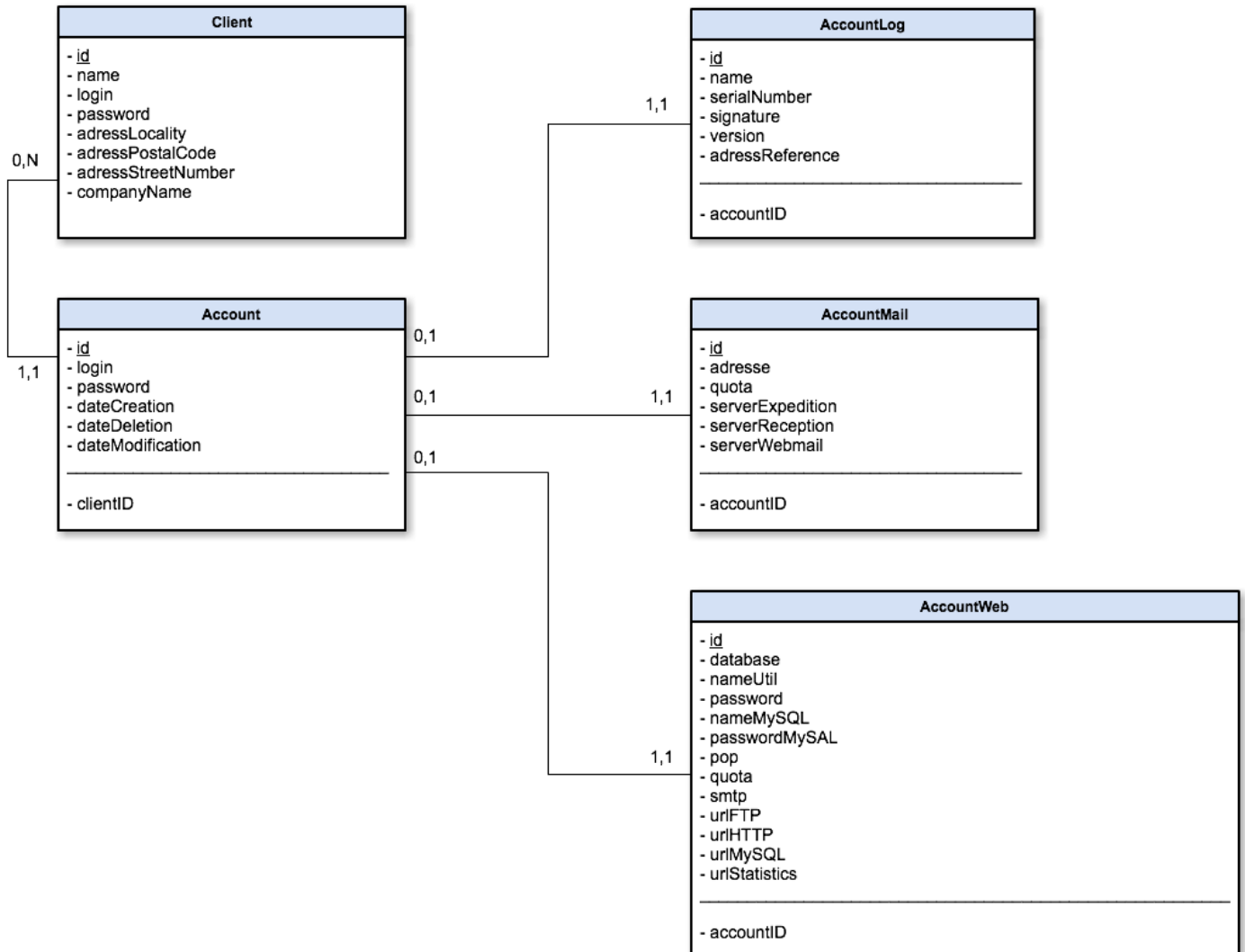


Figure 5 – Diagram of the database of the application cipiyou-Account

“Client”

The first table contains the customer information (name, address ...).

Global account (“Account” table)

This table contains data of a generic account. This generic account is linked to a client. A client may have several global accounts.

Specific account

A generic account must be one of these three types:

- Mail account ("AccountMail");
- Software account ("AccountLog");
- Server account ("AccountWeb").

To ensure data consistency in the application, some rules must be followed:

- Software account can't be linked to a global account if it's already linked to a specific account;
- A global account must be linked with a specific account.

In other words, there are four possibilities:

- An account with only one web account;
- An account with only one software account;
- An account with only one server account;
- An account with one server account and one mail account.

These constraints are managed in the application. A transaction is used to ensure that a global account and its specific account(s) is (are) created simultaneously. If an error occurred in the creation of accounts, it would be detected when showing account information and the user could correct the account.

2.2 View

The application is separated into three views.

2.2.1 Authentication

It allows the authentication of the administrator. It consists of a fieldset comprising two fields and two buttons.

2.2.2 Main View

The second view is the main view. All the information and functionality are included in this view.

Instead of multiplying screens, it was decided, in consultation with the future user, to concentrate the information on a single one, facilitating navigation and reading them.

This view is divided into four parts, visible in Figure 6:

- Customer display;
- Customer management;
- Multiple accounts management;
- Single account management.

Cipyou-Account

CLIENT : NYSSSEN PIERRE.

Login | Pierre Mot de passe | pasSecret eMail | pierre.nyssen@pyramide-ingénierie.be Nom Prénom | Nyssen Pierre
Code Postal | 4800 Localité | verviers Rue & N° | Rue du Brou, 83 Société | cipyou

Editer

LISTE DES COMPTES:

Type	Login	Créé le	Modifié le	Supprimé le
Mail	pierre_mail_1	14/04/15	14/04/15	
Mail	pierre_mail_2	14/04/15	14/04/15	14/04/15
Mail	pierre_mail_3	14/04/15	14/04/15	14/04/15
Mail	pierre_mail_4	14/04/15	14/04/15	
Mail	pierre_mail_5	14/04/15	14/04/15	
Soft	pierre_log_1	14/04/15	14/04/15	
Soft	pierre_log_2	14/04/15	14/04/15	14/04/15
Soft	pierre_log_3	14/04/15	14/04/15	
Soft	Pierre_log_4	14/04/15	14/04/15	14/04/15
Web	pierre_serveur_1	14/04/15	14/04/15	

Nouveau Compte

INFORMATIONS DU COMPTE:

- Compte

Login | Pierre_log_4 Mot de passe | pasSecret

- Logiciel :

Nom Logiciel | MacBuro Version | 3.0
Clé | XXXXXXXXXXXXXXXX Signature | YYYYYYYYYYYYYYYY
Contact Logiciel | macburo@cipyou.be

Figure 6 – Main view of the application cipyou-Account

Displaying customers

The searchable customer list is displayed on the left. The names appear in groups of fifteen.

This list is permanently visible for easy access to a customer.

Customer management

The second part, the upper frame, appears when a customer is selected. It contains the information of the customer. It is possible to edit the information.

This second part also appears when the administrator adds a client to the application.

Multiple accounts management

The third part, the middle frame, appears when a client is selected. It includes all user accounts of the customer. The administrator can select an account. The line of the selected account appears in green.

Single account management

The fourth part, the lower frame, appears when a user account is selected. It includes all of the account information. It is possible to edit information by entering edit mode.

This fourth part also appears when the administrator wants to add a new user account to the selected customer.

2.2.3 Warning Deletion

The third screen only appears when the administrator wants to delete a client that has still at least one active user account.

In this case, he is redirected to a page that indicates which accounts are still active and warns him/her that removing the customer will also delete all his/her user accounts.

2.3 Controller and features

WebObjects recommends the use of one controller per view. This controller contains the methods called from the view.

For cipiyou-Account, there are three windows and thus three controllers.

The application cipiyou-Account includes several features. Figure 7 shows the different possible actions at each state of the application. Each operation is possible at the next state, but is not repeated on the diagram to simplify it.

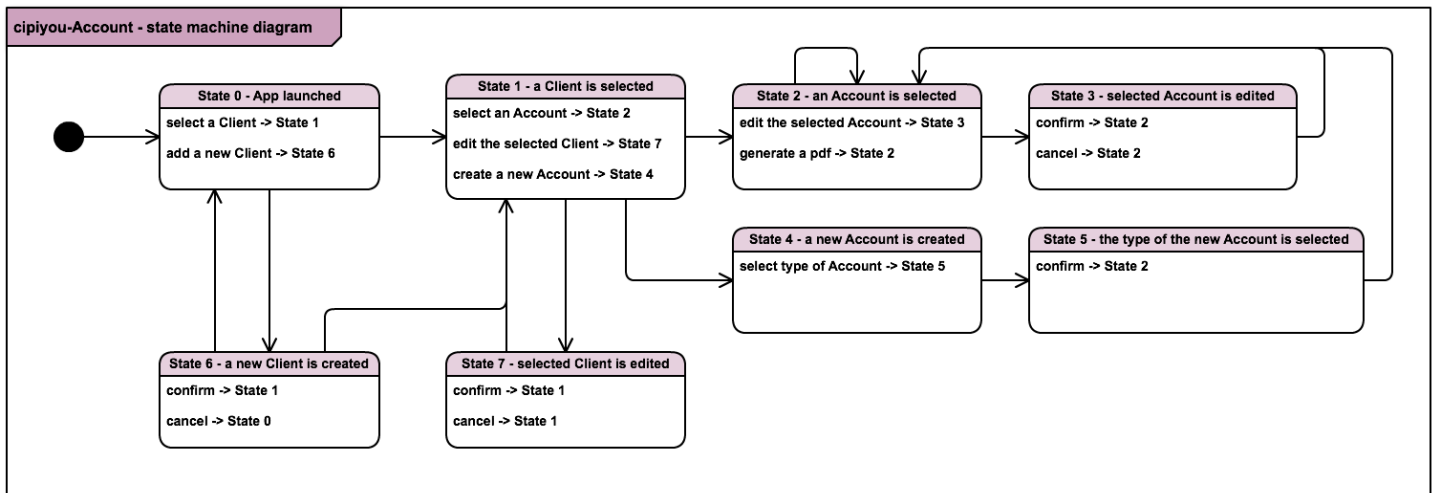


Figure 7 – State diagram of the application cipiyou-Account

2.3.1 Customer management

The first step to manage a customer is to create it (state 6) and to edit its information. Once in edit mode (state 7), the administrator can also manage customer information. He can:

- Delete the customer;
- Confirm the modifications;
- Cancel the modifications.

If the customer still has an active account and the administrator deletes it, he will be redirected to the "Warning Deletion" page.

2.3.2 User account management

To add a user account, its type must be selected (“Mail”, “Software” or “Server & Mail”; state 4). Then, the administrator may enter information of the account (state 5).

Once displayed in edit mode (state 3), the administrator can:

- Deactivate the account;
- Confirm the modifications;
- Cancel the modifications.

An active user account can be edited. An inactive account cannot be changed.

It is not possible to modify the type of an account.

2.3.3 PDF generation

When an account is selected, it is possible to generate a PDF file containing the account information. It is also possible to send it to the customer.

2.4 Deployment

The deployment of an application proceeds as follows.

WOLips (Eclipse) allows the creation of installation files for the application. The extension of these files is woa (for WebObjects Application). WOLips generates two files: one for the application and one for the resources.

These two folders are moved to the target server (in this case, it is a MacOS X server called “doc”).

Once placed on the server, permissions of the folder containing the application must be changed. The application must belong to the user “_appserverusr” (or “_appserveradmin”) and to the group “_appserver” with all rights (rwx).

The folder containing the application must be placed in the “/Library/WebObjects/Applications” folder. Resources must be placed in the “/Library/WebServer/Documents/WebObjects” folder.

The jars files added to the project must be added in the “/Library/Java/Extensions” folder.

Once the files are placed on the server, the application must be restarted. To manage the server, there is a web application called “WOMonitor”.

For a first installation, the monitor must be notified that there is a new application on the server by indicating to it the location of the woa file.

Then, a new instance of the application is created and started.

When dealing with an update of the application, it is enough to restart the instance of the application.

2.5 PDF generation

The application cipiyou-Account allows the generation of PDF files. WOLips uses two files (a “jasper” file and a “jrxml” file) to generate the PDF file.

These two files are created by an application called iReport. The jasper file is unreadable by the developer while the jrxml is readable and contains an XML description of the PDF. Both files are placed in the folder “Resources” of the application.

The application iReport allows the creation of a template (the content of the jrxml file). It is compatible with java application, but also JavaScript or Groovy application.

The jrxml file is created and edited by iReport. It is placed in the application project from iReport to link the files to the project. Before you can use this template, it must be compiled. This operation, which is done from iReport, generates the jasper file in the project (in the “Resources” folder).

Once compiled, the template is usable by the application.

It is possible to include basic java methods without connecting textfields to an object. For example, the method “new java.util.Date ()” is executed and displays the date of the creation of the PDF file.

The template is created by drag and drop. Elements are arranged on a page separated into several areas. These areas fit the content and each of these has their specificities. Figure 8 shows a template under construction from iReport. There are three distinct areas (header, body and footer). It includes elements of several types:

- Image (for the logo)
- Label (for fixed text)
- Textfield (for variable text)

When generating the PDF file, the variable texts are translated into text. A text “null” will be displayed as an empty string.

The screenshot displays an iReport template for software accounts, overlaid on a grid with a ruler. The template is divided into three main sections: header, body, and footer.

Header Section:

- On the left is the logo for "Pyramide Ingénierie", featuring the word "Pyramide" in a large serif font, "Ingénierie" in a smaller sans-serif font below it, and a stylized graphic of green triangles to the right.
- On the right, there is a date field: "Verviers, le new java.util.Date()".
- Below the date is a "Title" label followed by four text fields containing variable placeholders: "\${pdfGetCompanyName}", "\${pdfGetClientName}", "\${pdfGetStreetNumber}", and "\${pdfGetLocality}".

Body Section:

- It begins with a salutation: "Cher client," followed by a text field.
- Next is a paragraph of fixed text: "Nous vous remercions d'avoir commandé ce logiciel réalisé par notre société."
- Followed by another paragraph: "Vous trouverez ci-dessus les paramètres à encoder dans la fenêtre 'Licence d'utilisation...' pour utiliser pleinement le logiciel."
- Then, a longer paragraph: "Cette licence vous donne droit automatiquement à toutes les mises à jour du logiciel dont les 2 premiers nombres du numéro de version sont identiques à ceux du numéro de version repris sur ce document."
- Below this is a license information section with four rows of text fields:
 - "Licence pour : \${pdfGetName} v. \${pdfGetVersion}"
 - "Accordée à : \${pdfGetUsername}"
 - "Signature : \${pdfGetSignature}"
 - "Clé : \${pdfGetSerialNumber}"
- Then, a paragraph: "Pour toute question technique relative à ce logiciel, nous vous prions de vous adresser par email à l'adresse reprise ci-dessous:"
- Followed by a text field containing the variable "\${pdfGetReferenceAddress}" and a label "Soyez assuré de notre considération distinguée,".
- At the bottom of the body is a signature line: "Luc Demarche" followed by a text field.

Footer Section:

- On the left, two text fields: "tél (+32/0) 87 292130" and "fax (+32/0) 87 292139".
- In the center, a text field containing "Pyramide Ingénierie sprl", "rue du Brou 83", and "B-4800 Verviers".
- On the right, two text fields: "rpm Verviers" and "tva/id BE-0436.881.565".

Figure 8 – jrxml file created in iReport for software accounts

Figure 9 shows an example of a PDF file generated from the template of Figure 8. Variable text fields have been replaced by the contents of the objects to which they were connected. For example, the “\$F (pdfGetCompanyName)” field were replaced by “cipiyou”.

The method “new java.util.Date ()” executes and displays the date of the creation of the PDF file (here, “15 April 2015”).

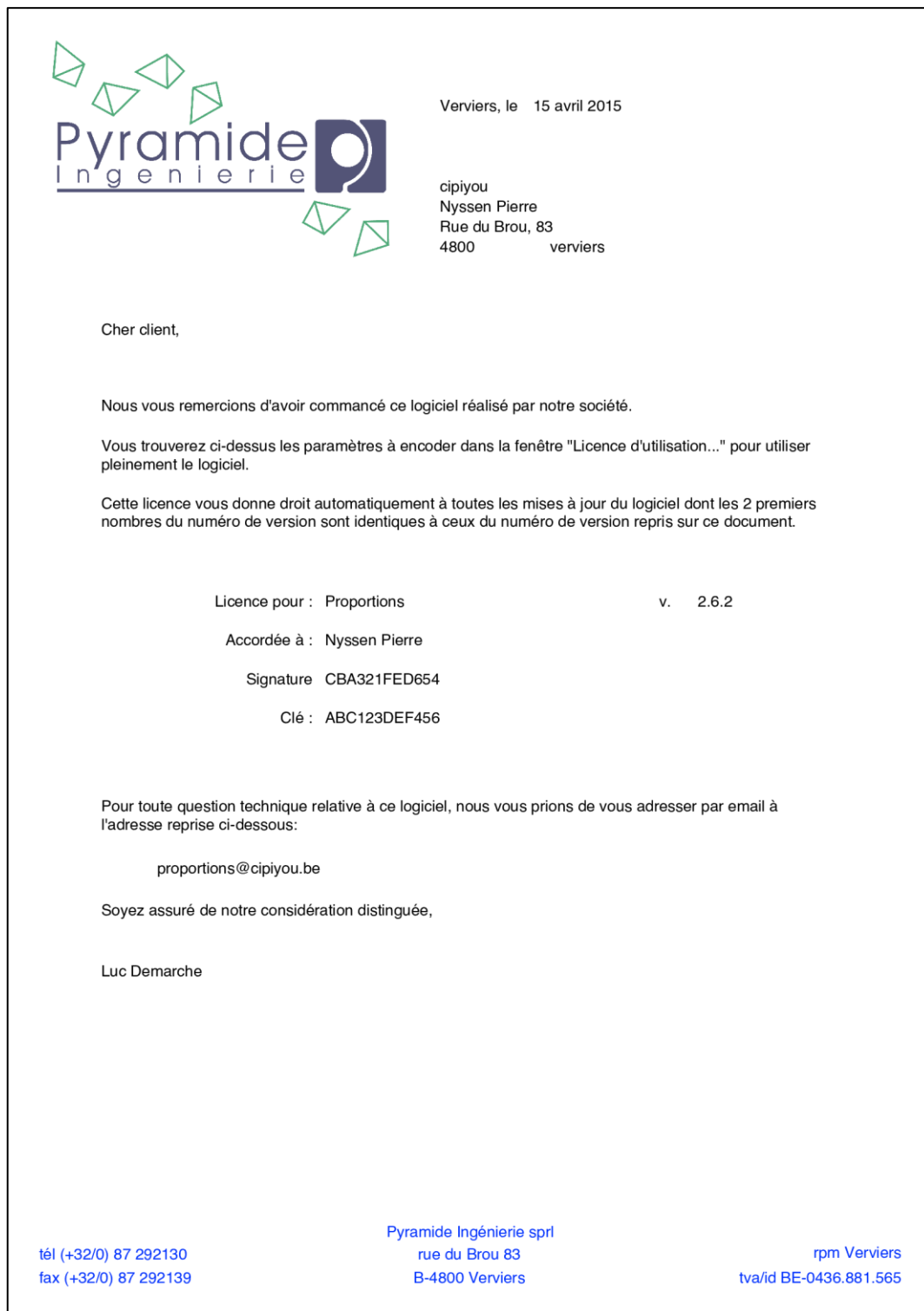


Figure 9 – Sample PDF file generated by the application cipiyou-Account for a software account

2.6 Future of the application cipyoun-Account

The old application stores its data in a file. Data must be transferred from the file to the new database. A program should be written since there is too much data to input manually to the database. This program has not yet been developed because it is not a priority at this time.

One suggested feature is to manage several accounts at once. For example, remove several accounts at once, or allow the administrator to generate all PDF files of a given type at once.

It should also be possible to allow clients direct access to the application. However, consideration would have to be given to things such as, client authentication, permissions and what they can see."

Another feature to consider would be to reactivate an account (as an administrator), for example if an account is inadvertently deleted.

Conclusion Part I

WebObjects is a tool for creating web applications that provide facilities.

For the generation of web pages, it uses a view (HTML file + WOD file) linked with java code.

To interact with the database, it includes EOFramework with:

- EOModeler to create the database and generate Entities classes;
- EditingContext to optimize access to the database.

I created the application cipiyou-Account in order to learn WebObjects. I learned to use most of the tools and features that I would need to use later.

PART II: Order Processing System

As an introduction, I will first talk about the architecture of Pyramide Ingénierie and about the order processes used by cipiyou.

I will continue by discussing the delivery solution from conception to integration and the creation of a discount coupons system.

I will end with the payment solution by describing the choice of the “HiPay Direct” solution and the integration of this solution.

Introduction

Architecture of the company system

Figure 10 shows a diagram of the architecture of Pyramide Ingénierie. It includes the WebObjects application server (called “Happy”) on which cipiyou-web and cipiyou-manage applications are running.

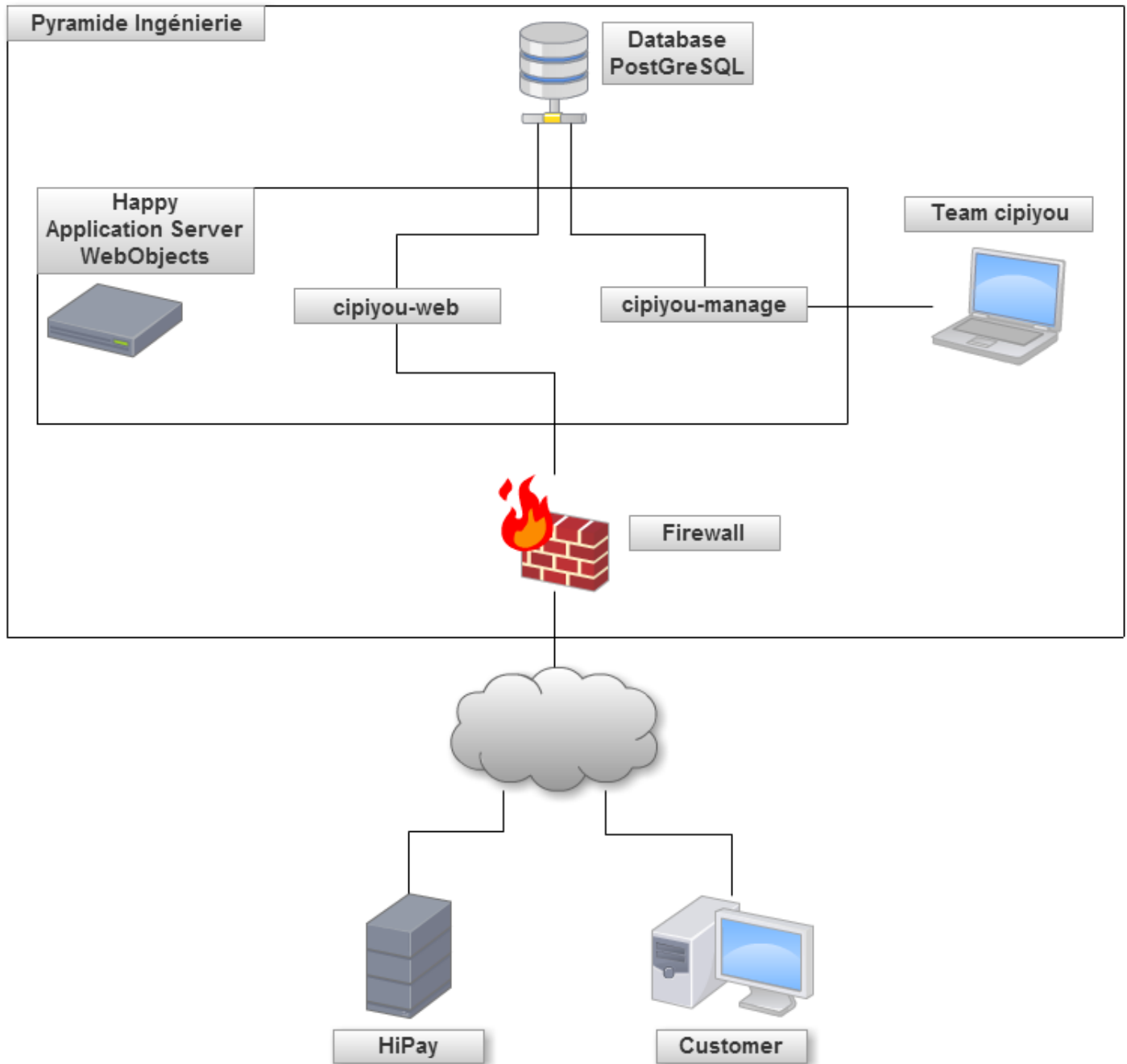


Figure 10 – Diagram of the architecture of Pyramide Ingénierie

Former order process

Figure 11 shows the old order process used by cipiyou. The transactions in blue occur without delay, while those in red require a delay.

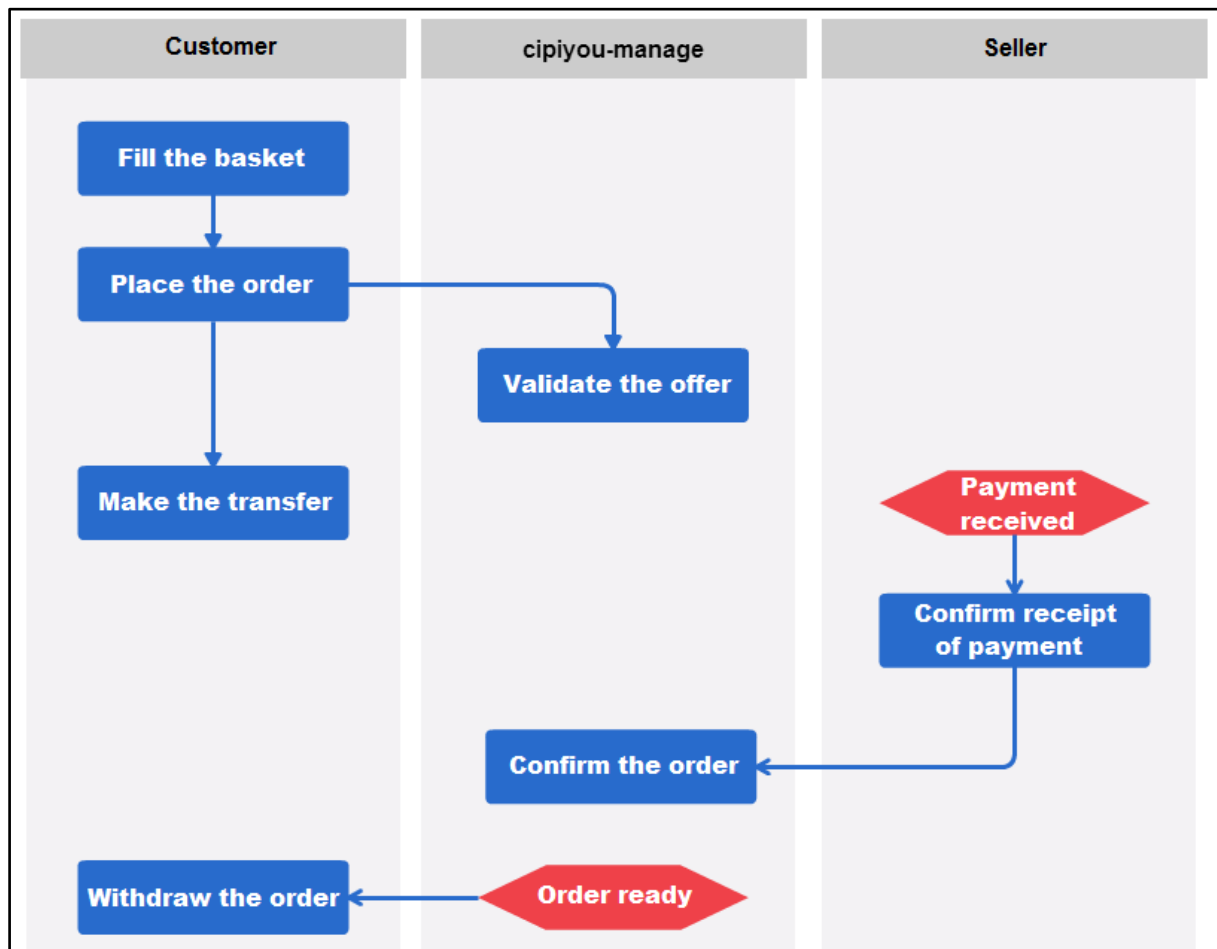


Figure 11 – Diagram of the old order process

When placing an order, the customer had to pay by making a bank transfer. Once received (several days later), a seller first had to verify payment data, then take the basket and manually to turn it into an order. Once the order was ready, the customer had to pick it up from cipiyou.

This caused several problems: a significant delay (several days for the transfer to be received), a risk of cancellation and refunding and the need for manual check of the payment information.

This system was almost never used, less than one sale in 500 was done this way¹.

¹ Data obtained from Mr. Demarche, manager of cipiyou.

New order process

Figure 12 shows the new order process used by cipiyou.

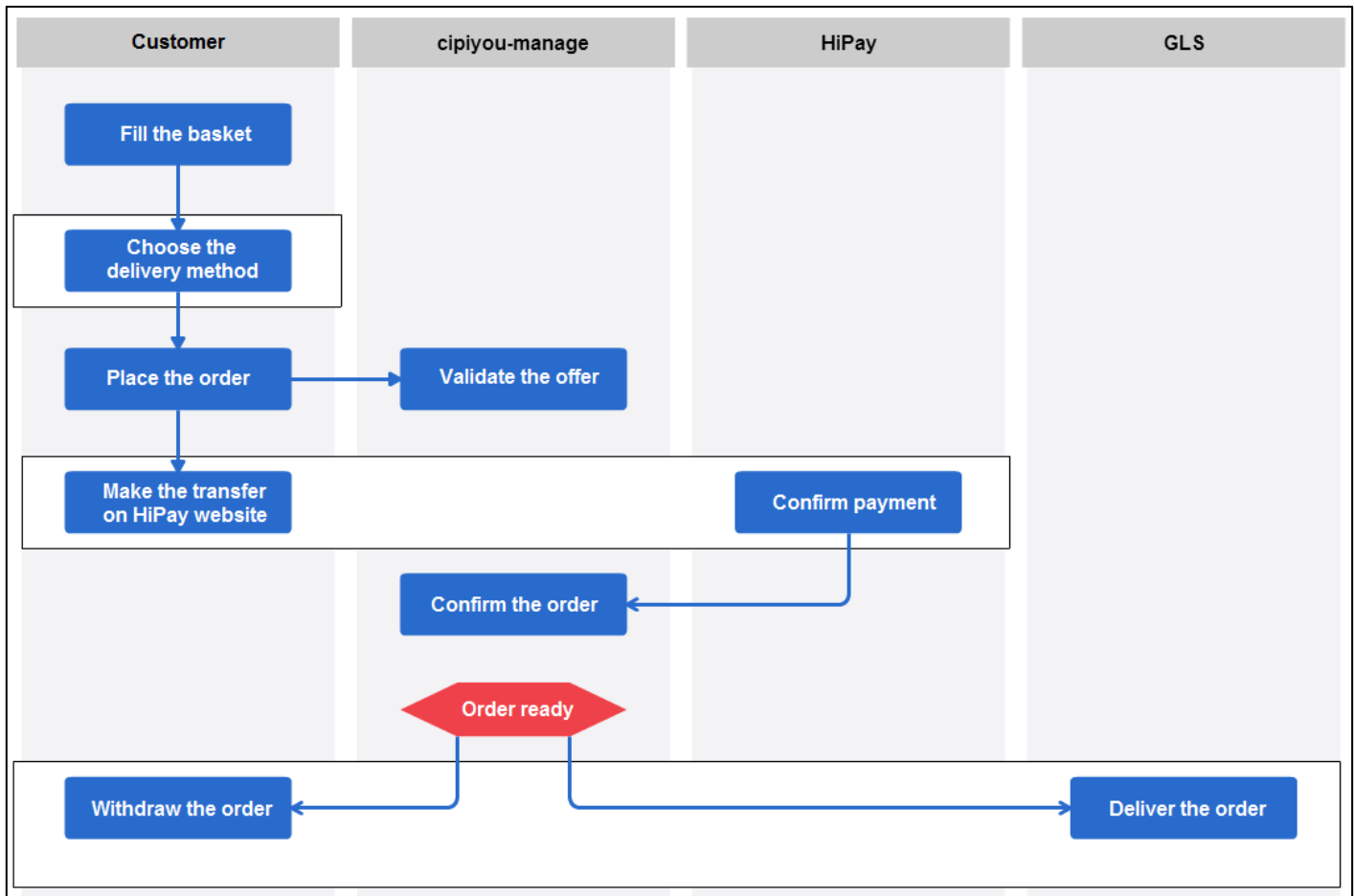


Figure 12 – Diagram of the new order processing method used by cipiyou

The order process was modified in several ways.

Firstly, the customer can now choose a shipping method when placing his order.

Secondly, the payment process is managed by HiPay and no longer causes a delay (we will discuss the choice of HiPay and its integration in Chapters 5 and 6, respectively).

Finally once the order is ready, the customer can withdraw it or it is delivered by GLS, depending on the choice of the delivery method (whose integration is explained in the next chapter).

Chapter 3: Delivery

When a client placed an order online, he could not select a delivery address. He had to pick up his order in-store.

I had to integrate the choice of delivery to a given address. This address may be entered in the customer profile or when choosing a delivery method.

The Pyramide Ingénierie partner for delivery is the company GLS. GLS is responsible for the delivery of orders to the cipiyou store. They deliver once per day. GLS uses this daily delivery to pick up orders to be shipped to clients.

3.1 Changing the framework cipyoun-web-manage

The choice of the delivery method is as follows (the customer must be connected and have at least one product in his cart).

The customer selects the one of the following three shipping methods: collect at cipyoun, delivery at the main address or delivery at another address. Once the method is selected, a transport product is added to the basket.

When the basket changes, transport products are removed from it. It is thus not possible to have several transport products in a basket.

Once the product is added to the cart, the customer can proceed to payment.

3.1.1 Pricing

The first step was to get a quote from GLS. That quote is in the form of a two-way table², the first indicating the destination country and the second showing the weight range for parcels. This grid is applicable from Belgium.

The pricing is done on these two criteria: the package weight and the destination country.

Each product has a weight; the total weight of the basket is calculated by adding the weight of each ordered product multiplied by the quantity ordered.

When indicating the delivery address, customer fills in the destination country.

We have thus all the necessary information to calculate the cost of delivery to charge the customer.

3.1.2 Transport product

The database contains a table "Product". These products belong to the family for transport products (number 65). Prices and related transport products were manually encoded in the database.

I added two columns to the table "Product": one for maximum weight and one for minimum weight. Each transport product has a minimum weight value and a maximal weight value and thus a weight range. This inclusion is the only change made to the database.

When selecting the delivery method, the weight of the basket is compared with the weight range of transport product to select the correct product.

Products also have a description. The description of transport products contains the countries covered by the product. For example, the description of "Benelux Delivery" contains "Belgium Netherlands Luxembourg".

When encoding transport products, I fixed the amount to 999. It indicates that the product is always available and that its stock should not change. This feature was already present and used in the application.

² See Attachment 1: « Tariff Grid for parcel delivery from Belgium by GLS ».

3.1.3 Delivery method selection

There are three ways to delivery an order.

The first one is “pick-up at cipiyou”. The customer does not pay shipping costs for this method. This was the default delivery method of all orders.

The second method is to deliver to the main address of the customer. This address is the one in his profile.

The last one is to deliver to another address. The customer enters an address and validates it before placing his order. He also chooses a country from a list (Benelux, France or Germany).

When the second or the third delivery method is selected, a transport product is added to the basket. The correct product is selected by analyzing its description (which is compared to the selected delivery country) and its weight range (which is compared to the weight of the basket).

3.1.4 Order status

Once the order has been paid, it appears in the customer’s list of orders. Since WebObjects remembers the EditingContext for about twenty minutes without reloading the database, there may be a delay between the moment the customer pays for his order and the moment his order appears in the list.

An order is validated when it is paid. That means that an online order is automatically validated. Artides in stock are removed from it and artides that are not in stock are ordered from the supplier. This order status management was already present in the application cipiyou-manage, I added it to the application cipiyou-web.

Once all items are in inventory, the order is set in progress. Each morning, a report is produced with the orders in progress. I added two reports that contain orders to prepare for a pick-up at cipiyou store or for delivery by GLS.

At each step of the preparation, the customer can see the status of his order from his user space.

When an order is ready to be picked up or to be sent, the customer receives a notification by email or via Twitter, according to the preferences indicated in his user space.

3.1.5 Overview table

To facilitate the processing of orders in the application cipyounmanage, there are several dashboards showing various orders. I have separated the table “in process” in two parts.

The allocation of the various orders is based on a store identifier (“shopID”). There is an identifier for the technical service (2) and for the cipyoun store (3).

To separate orders placed on the web from those placed in-store, I created a store identifier for the web shop (4). To differentiate orders to be delivered from those to be picked up in-store, I created a store identifier for the customer’s home (5). By combining the different identifiers, all possibilities are covered (orders placed on the web or at the store and withdraw at the store or home delivered).

I separated the table of orders in progress into two tables:

- A table for orders that the customer will pick-up at the cipyoun store (3);
- A table for orders that GLS will deliver to the customer (5).

The addition of these two store identifiers has had other repercussions in the application cipyounmanage. Other summary tables also used the shopID and can display separately orders placed on the web, orders to be picked up at the store and orders to be delivered to the customer.

3.2 Views of the delivery selection procedure

The choice of delivery method is done via three pages that were added to the application.

3.2.1 First view: ContentShoppingCart

The first view is accessed by clicking on the “View Cart” button present on all pages.

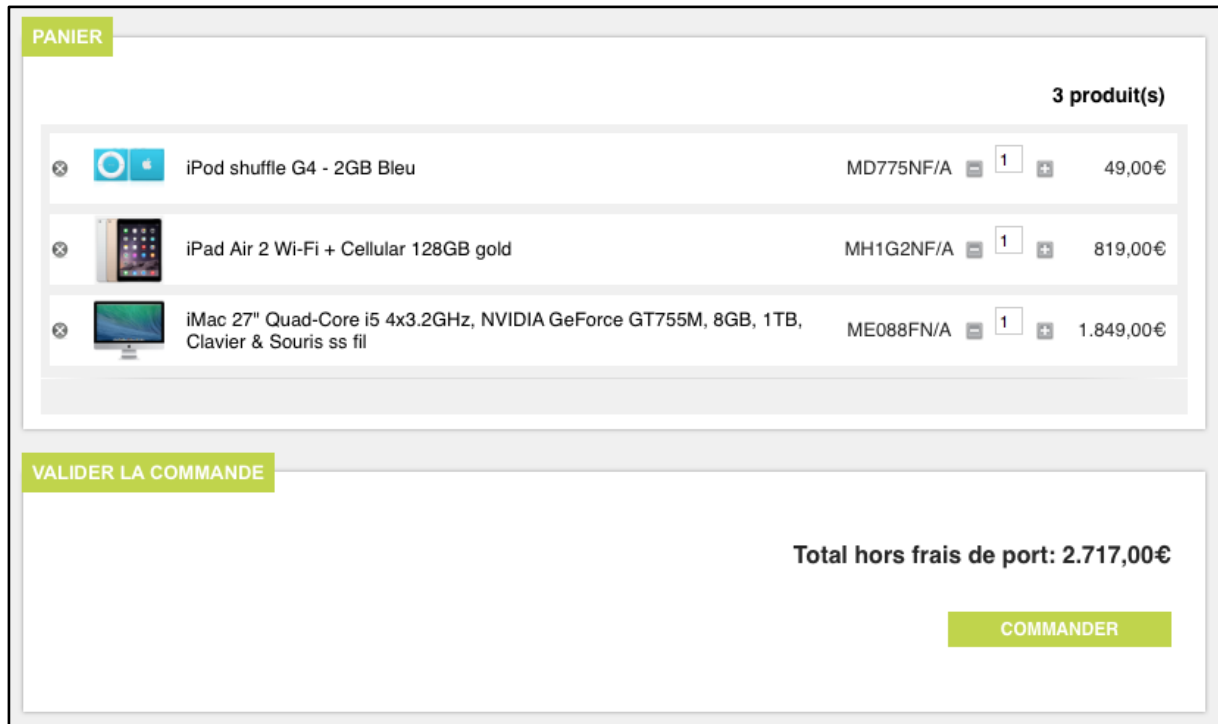


Figure 13 – Screenshot of the page “ContentShoppingCart”

Figure 13 shows that the component is separated into two fieldsets:





- Basket, in which its content is displayed;
- Confirm Order, used to display a total cost excluding shipping cost and continue the order.

The user needs to be identified to place his order.

3.2.2 Second view: ContentLieuxLivraison

PANIER

4 produit(s)

	iPod shuffle G4 - 2GB Bleu	MD775NF/A	1	49,00€
	iPad Air 2 Wi-Fi + Cellular 128GB gold	MH1G2NF/A	1	819,00€
	iMac 27" Quad-Core i5 4x3.2GHz, NVIDIA GeForce GT755M, 8GB, 1TB, Clavier & Souris ss fil	ME088FN/A	1	1.849,00€
	Frais de livraison GLS Benelux 10-15 kg	GLS-BNL-10-15	1	17,90€

MODE DE LIVRAISON

Veuillez choisir une méthode de livraison :

☐ Retrait chez cipiyou

☒ Livraison à l'adresse principale (Avenue de la belle Avenue, 42 4000 LIÈGE)

☐ Livraison à une autre adresse

RÉCAPITULATIF

Total hors frais de port:	2.717,00€
Frais de port:	17,90€
Total frais de port compris:	2.734,90€

ANNULER

CONFIRMER

Figure 14 – Screenshot of the page “ContentLieuxLivraison”






Figure 14 shows the second page. It consists of three fieldsets:

- Basket, identical to the previous view;
- Delivery method, in which the customer chooses the delivery address;
- Summary wherein different totals are displayed.

3.2.3 Third view: ContentConfirmToPayment

PANIER

5 produit(s)

	iPod shuffle G4 - 2GB Bleu	MD775NF/A	1	49,00€
	iPad Air 2 Wi-Fi + Cellular 128GB gold	MH1G2NF/A	1	819,00€
	iMac 27" Quad-Core i5 4x3.2GHz, NVIDIA GeForce GT755M, 8GB, 1TB, Clavier & Souris ss fil	ME088FN/A	1	1.849,00€
	Frais de livraison GLS Benelux 10-15 kg	GLS-BNL-10-15	1	17,90€
	Code : iPad5 : 50.0€		1	-50,00€

MODE DE LIVRAISON

Livraison à cette adresse :

Pierre NYSEN
Avenue de la belle Avenue, 42
4000 LIÈGE
BELGIQUE

CONCLURE LA VENTE

Total frais de port compris : 2.684,90€

Je dispose d'un code de réduction dont le code est :

J'ai lu et j'accepte les [conditions générales de vente](#) : ☒

ANNULER

PAYER

Figure 15 – Screenshot of the page “ContentConfirmToPayment”

The third page, shown in Figure 15, is also separated into three fieldsets:

- Basket, used to show the basket for the last time;
- Delivery method, to confirm the choice of delivery method;
- Finalize the sale, to read and accept terms and conditions and potentially enter a discount code.

3.3 Conclusion

The implementation of the delivery was separated into three parts.

For the first one, creating the delivery product, I added a column for a minimum weight and a column for a maximum weight in the table "Product".

In the second part, I added summary tables to the dashboard of the application cipiyou-manage. This led to the creation of two additional store identifiers (one for the web store and one for the customer's address).

The third part was adding interfaces to the application cipiyou-web. The old page was replaced by three pages for visualizing the cart, choosing the delivery method and to accept terms and conditions.

These pages have been submitted several times to the manager of the website to ensure visual consistency with the rest of the website.

Chapter 4: Discount coupons

In this chapter, we will see the creation of a discount coupon system and its integration into the cipyou-web-manage framework.

To refer to the ordering process, customer can use a coupon after selecting delivery method and right before proceeding to payment.

4.1 Database

Figure 16 shows the tables used for coupons. Tables in red (“Family”, “Product” and “Client”) were already in the database while tables in blue were added to the database.

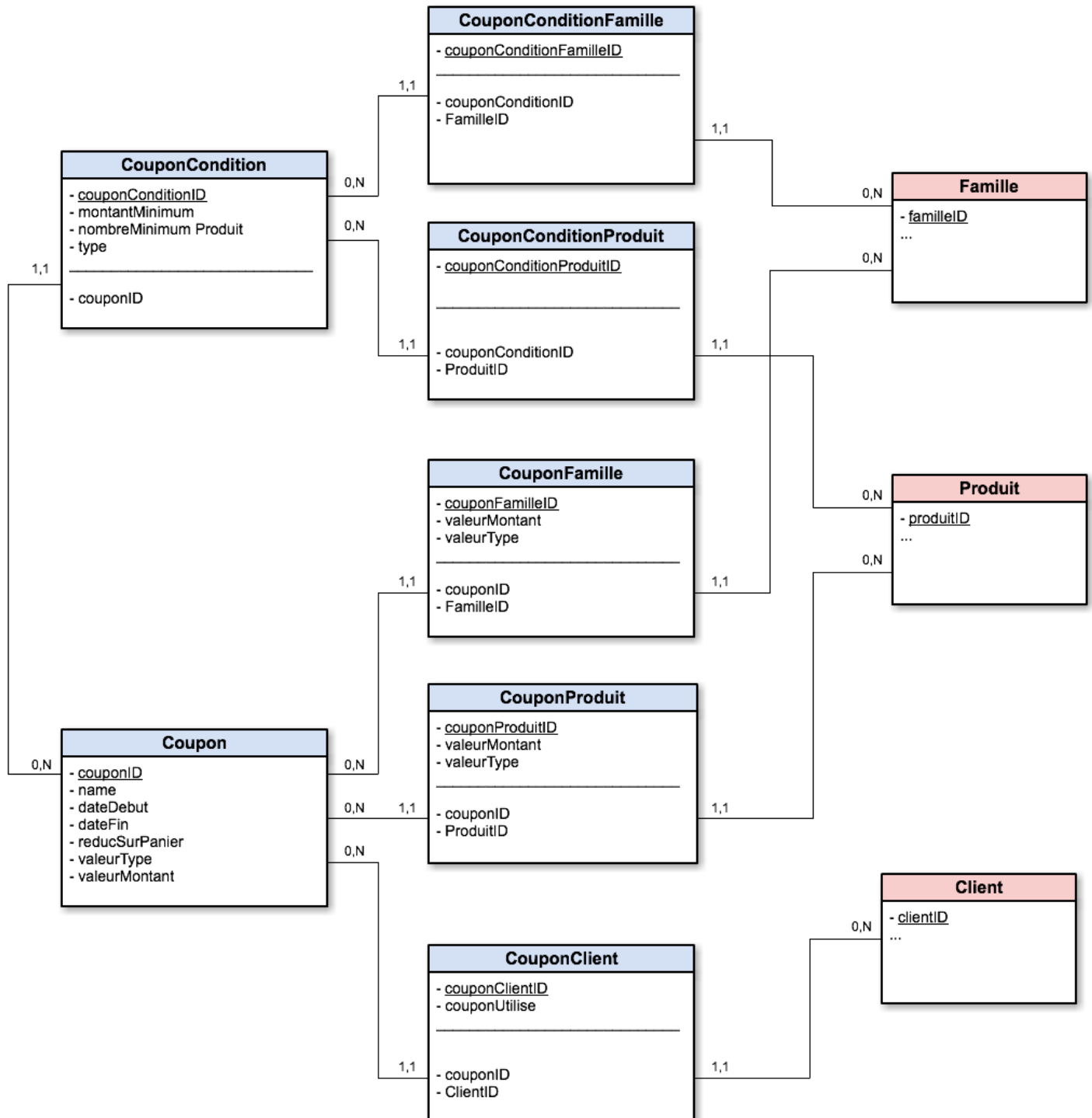


Figure 16 – Diagram of the tables used for discount coupons

Coupon

This table allows the user to create an overall price reduction by entering a code. For example, the code “gift5%” would offer a discount of five percent on all products in the basket.

The table “Coupon” contains the following values:

- A name (the code entered by the customer);
- A start date of validity;
- An expiration date;
- A global reduction.

It is possible to create two coupons with the same name but different validity dates. The name cannot be a primary key for this reason; therefore a coupon identifier (“couponID”) has been added.

The field “reducSurPanier” represent the type of reduction and can accept four values:

- 1 if the reduction is global;
- 2 if the reduction applies only to families of product;
- 3 if the reduction applies only to products;
- 4 if the reduction applies to both products and families thereof.

It is not possible to combine an overall discount with a reduction concerning families or products because some products could be subject to several discounts (with a risk of excessive discounts).

The double value “valeurMontant” indicates the amount of the transaction and the character “valeurType”, which can be ‘%’ or ‘€’, indicates the type of this amount (e.g. “5%” or “€ 40”).

CouponClient

This table is used to limit the use of a coupon to one or several selected customers. If the coupon is not linked to at least one customer, it is thus a universal coupon. It can be used several times by each customer. If a universal code was limited to one use per customer, customers could create fake accounts to use the code several times; that would have polluted the database.

This table includes an integer value “couponUtilise” and two foreign keys (one for the table “Coupon” and one for the table “Customer”).

This table is used to link several customers with a coupon.

Each pair of keys “clientID” and “couponID” is unique and could serve as primary key. However, WebObjects and EOFramework are limited and do not allow multiple primary keys. For this reason, an identifier was added.

The integer “couponUtilise” can take three values: 0, 1 or 2. 0 means that the coupon is not used 1 means that the coupon is blocked and 2 means that it is validated.

For example, the code “loyalty5%” could reward the customers who made a purchase in the last month.

CouponFamille & CouponProduit

These tables allow for the application of a reduction to certain products or product families.

For example, the code "DeliveryOffer" could apply a "100%" reduction on products from the family "Transport".

These tables contains a Double value "valeurMontant", a character value "valeurType" and two foreign keys, one to a coupon and one to a family (or a product).

As for the table "CouponClient", a pair of keys could have been primary keys but an identifier was added.

The field "valeurType" can accept two characters; '€' or '%'. The value "valeurMontant" is a Double. Together, these values indicate a reduction (e.g. "12.5%" or "€ 679.5"). This reduction is related to a family (or a product) and will be applied only on the articles of this family (or product).

CouponCondition

The table "CouponCondition" is used to apply activation conditions (minimal order amount and minimal number of products) on coupons. It contains:

- An identifier ("couponConditionID");
- A double "montantMinimum" indicating the minimum amount of the basket;
- An entire "nombreMinimumProduit" indicating a minimum number of products in the basket;
- An integer "type" indicating a minimum number of different products or products from different families.

For example, the code "Delivery150" would result in free delivery on any purchase of €150 or above.

CouponConditionFamille & CouponConditionProduit

These tables allow linking a condition to one or more families (or to one or more products). They include two foreign keys (“couponID” and “familleID” (or “produitID”)). An identifier was added.

Used with the CouponCondition table, it is possible to add specificities to a condition. For example, it is possible to specify a minimum number of products of a family (if a client purchase two iPods Shuffle, he will receive a reduction).

The value “type” specified in the table CouponCondition is important here. It is used to indicate the number of different families (or products) that must be present in the basket.

For example, “if a client purchases two iPods Shuffle” means that “nombreMinimumProduit” is 2 and that the type is 1.

But “if a client purchases two different iPods Shuffle” means that “nombreMinimumProduit” is 2 and that the type is 2, because the 2 iPods Shuffle have to be different.

Used together, these tables cover most conceivable coupons.

Compound primary keys

EOFramework has a limit that presents some problems: it is not possible to manage compound primary keys.

EOModeler can create tables with compound primary key Tables “CouponConditionFamille”, “CouponConditionProduit”, “CouponFamille”, “CouponProduit” and “CouponClient” could use a couple of keys as primary key. However, when creating and instantiating objects of these tables, it was not possible to insert them into the EditingContext.

The EditingContext is a representation of the database. When the application access a data present in the EditingContext, it is not loaded from the database. If the data isn’t in the EditingContext, it is retrieved from the database and placed in the EditingContext.

To access data in EditingContext and link it to the database, WebObjects uses a global identifier (globalID). This identifier links an object of the EditingContext to the primary key of the object in the database. The globalID can only be linked to one attribute serving as a primary key of one table. I had to add a primary key “ID” to each of these tables so that objects could be associated with a globalID and added to the EditingContext.

4.2 cipyou-web

4.2.1 Insert a code

In the application cipyou-web, the first step for the customer is to input his discount code.

I needed to create a new field in the interface. The customer enters his discount code (if he has such a code).

The code has to be validated. When the focus is lost by the field, the page is refreshed to verify the code.

If the code is correct, the field is surrounded by green. If it is wrong, it will be framed in red and an error message appears.

4.2.2 Verify the code

The verification of the code is done in several steps.

Firstly, tables are reloaded from the database to ensure data validity.

Then, the name of the coupon is checked. If a coupon with this name is not found, the coupon does not exist.

If coupons with that name are found, validity dates are verified. If the date of utilization is not between the start date of validity and the expiration date, the coupon is rejected.

If a coupon is valid, the customer may use the coupon. If the coupon is related to a customer but not this one, the customer may not use it. If the coupon is related to this client but the value "couponUtilise" is not 0, then that means that the customer has already used his coupon. If the value is 0 or if the coupon is not related to any customer, then the activation conditions must be checked.

If all activation conditions are met, the coupon is validated and executed.

4.2.3 Execute coupon

This step transforms the coupon into an object that can be added to your cart. The cart can only contain products; coupons cannot be added directly in the cart. I had to create a product corresponding to a use of a coupon.

The choice to separate coupons and products has been done given the big differences between these entities. No data were shared, it was better to create new tables for coupons. So I had to find a way to add a coupon to the cart.

The `cipiyou-manage` application allows the creation of special products. These products are disposable; they allow selling products with specific conditions that will not recur. I used special products to create coupons uses.

A special product has a name. This name is composed of one coupon and its value.

A special product has a sale price, which is calculated based on the value of the reduction. The type of reduction depends of the value of `"reducSurPanier"`.

If the value is 1, then only the global reduction is applied.

If it is equal to 2 or 4, then each product in the cart is checked to see if a family reduction applies to it.

If it is equal to 3 or 4, then each product in the cart is checked to see if a product reduction applies to it.

Concurrent discounts can, in theory, occur if `"reducSurPanier"` is equal to 4 and if a product is concerned by a family reduction and a product reduction. If a product is subject to a reduction, the application `cipiyou-manage` does not allow the creation of another reduction for this product.

Once the amount of the reduction is calculated, the special product is added to the cart and the customer can proceed to payment

4.2.4 Ensuring limited use of coupon

A personal voucher cannot be used multiple times by the client. To ensure this, I added a status ("couponUtilise") to the CouponClient relation.

When should status be changed?

Status is modified, after verification, at the beginning of these three steps:

- Status is set to 0 when the coupon is created. Indeed, a new voucher will not be used;
- Status is set to 1 (blocked) when the payment process is initiated;
- Status is set to 2 (used) when the order has been paid.

When should status be verified?

Status is checked before each crucial step to buying. There are two important steps:

- When the client adds a coupon to the basket, the status must be 0;
- When the payment process is initiated, the status must also be 0.

Figure 17 shows the sequence of the verification status of the coupon when the customer inserts the code.

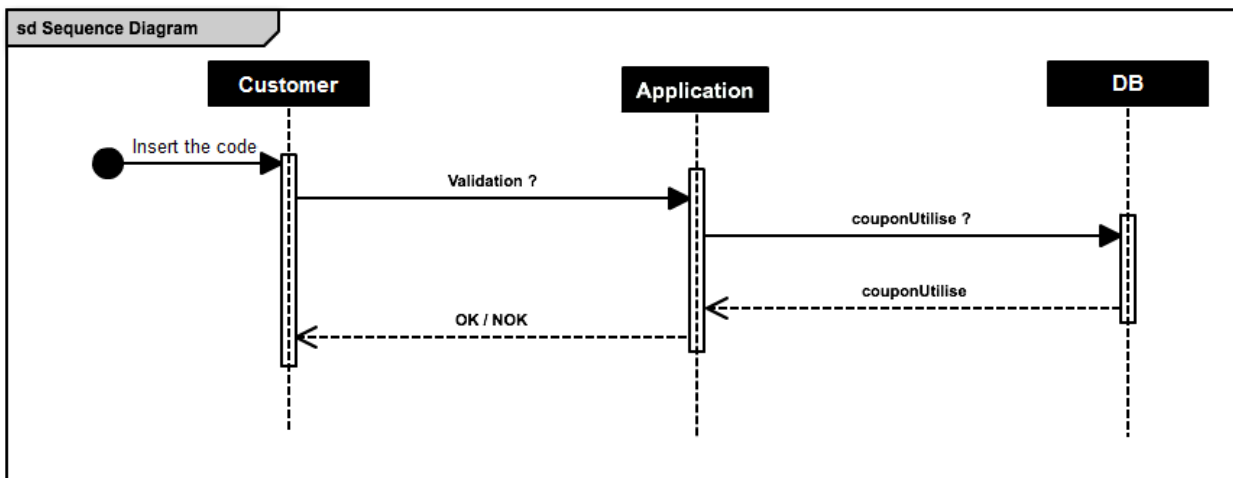


Figure 17 – Diagram of the validation process of a coupon

Checking the status before starting the payment process ensures that the customer has not validated the same code twice (for example by connecting multiple instances simultaneously).

WebObjects retains the EditingContext for 20 minutes. To prevent the user from using his code more than once in those 20 minutes, the validation reloads the coupons data from the database each time it is needed.

4.3 cipiyou-manage

Coupons are included in the cipiyou-manage application at several places to allow their management (creating coupons, viewing status, editing validity date, amount ...).

4.3.1 View of the edition of a coupon page

Figure 18 shows the editing interface while editing a coupon “livraison200” that offers free delivery on all orders over €200.

COUPON

Type: Produit

Réduction globale:

Code: livraison200

Valeur ("€" / "%"): 0.0€

Date de lancement: 15/05/15

Montant minimum: 200.0 €

Date de fin (non incluse): 30/05/15

Nombre de produit minimum: 0

RÉDUCTIONS <<

Familles affectées par le coupon:

Valeur: Famille(s):

100.0%	Transport	+
--------	-----------	---

Produits affectés par le coupon:

Valeur: Produit(s):

CONDITIONS <<

Conditions d'activation du coupon (familles):

Nombre minimum de familles différentes: Montant Minimum: Nombre minimum de produits: Famille(s):

1		1	Transport.	+
---	--	---	------------	---

Conditions d'activation du coupon (produits):

Nombre minimum de produits différents: Montant Minimum: Nombre minimum de produit: Produit(s):

CLIENTS <<

Clients concernés par le coupon:

Id Nom Statut

ACTIONS

Annuler

Supprimer

Confirmer

Figure 18 – Screenshot of the coupon editing interface

When creating a product, it is possible to select the product type. It can be “Product” (to create a single product) or “Bundle” (to create a package of products). I also added the type “Coupon” (to create a coupon). Three types of products require three different interfaces.

The “Coupon” interface is divided into several zones:

- “Coupon” to enter the general information of coupon;
- “Reductions” to add or remove reductions on a family or a product;
- “Conditions” to add or remove conditions on families or products;
- “Customers” to link the coupon with one or more customers.

When the user wants to enter a new discount or condition or to add links with customers, a modal window opens and the user can choose different values. Figure 19 shows the window used to add a reduction on a family.

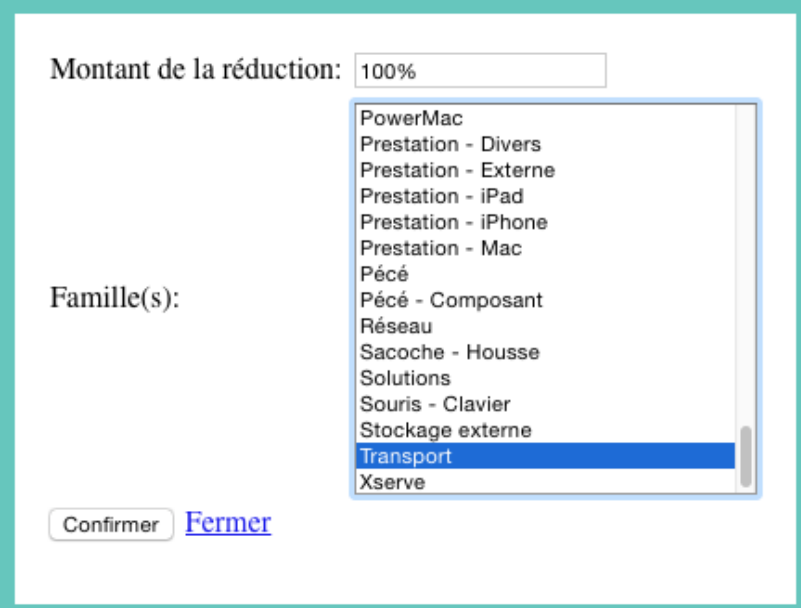
A modal window with a light blue border. At the top left, the text "Montant de la réduction:" is followed by a text input field containing "100%". Below this, the text "Famille(s):" is followed by a scrollable list box. The list box contains the following items: PowerMac, Prestation - Divers, Prestation - Externe, Prestation - iPad, Prestation - iPhone, Prestation - Mac, Pécé, Pécé - Composant, Réseau, Sacoche - Housse, Solutions, Souris - Clavier, Stockage externe, Transport (highlighted in blue), and Xserve. At the bottom left, there is a button labeled "Confirmer". At the bottom right, there is a blue text link labeled "Fermer".

Figure 19 – Modal window used to add a reduction on a family.

To link a customer to the coupon, the user can either enter the customer username or add all customers who made a purchase of a minimal given value from a given date.

4.3.2 How to create and to edit a coupon

When to create a coupon

A difficulty encountered in the coupon creation process was choosing the right time to create the coupon and to add it to the EditingContext. I had to avoid incomplete coupons in the database. To do so, the created coupon is kept in the session. When the user confirms the creation, the coupon is added to the EditingContext and to the database.

Lists of conditions, reductions and customers

Two lists of reductions (one for families and one for products), two lists of conditions (again one for families and one for products) and a list of customers are maintained in the session. When a coupon is added, lists are filled with different information.

Adding conditions, reductions and customers

Each time a condition, reduction or customer link is created, an instance of the object is created and added to the corresponding list. When the user confirms changes, all new instances (the ones in the lists but not yet in the EditingContext) are added to the EditingContext and linked with the coupon.

Deletion of conditions, reductions and customers

When the user deletes a condition, reduction or customer, the EditingContext should not be updated until the user confirms changes. The instance of the object is deleted from the corresponding list. If the user confirms changes, old instances (the ones in the EditingContext but no longer in the lists) are removed from EditingContext and deleted.

Cancel a modification

If the user cancels a modification, the lists are reloaded from the EditingContext. Doing so assures that displayed data are up to date. Each time the user refreshes the page, lists are updated.

Edition state

Lists are reloaded from the EditingContext every time the page is refreshed. However, they should not be updated when a modification is made. To refresh the page without actualising the list, I added a Boolean value, an edition state.

When modifying (adding or deleting) an object (condition, reduction or customer), the edition state is set to true and the page is refreshed.

When refreshing the page, if the edition state is true, then the lists are not updated and the edition state is set to false. If the edition state is false, the lists are updated.

This helps to keep changes despite page refresh.

4.3.3 Statistics and summary dashboards

Statistics

The cipiyou-manage application displays a dashboard containing sales statistics for any product. In this table, the different commands containing the product are displayed. I changed the statistical table so that coupons may be displayed in a different way.

Information about the coupon is displayed. Users can find a table listing the uses of the coupon, the number of coupons still not used is shown and the total of the amount of each utilisation is calculated.

Summary

When the user performs a search among products, he can ask for a summary of the selected products. If all the selected products are coupons, the summary screen is adapted to display other values.

The amount, the number of remaining utilisations and the number of utilisations of each coupon are displayed.

The total amount of selected coupons is also displayed.

Chapter 5: Selecting a payment solution

This chapter is devoted to the selection of a payment solution and the rationale that led to the selection of the solution “HiPay Direct”.

Existing solution

In the old order process, a customer could already place orders on the website. To for pay his order, he had to make a bank transfer. Once the payment was received, the order was processed.

Having to wait for the transfer added a delay of several days to the order treatment. It was necessary to manually check the amount and the completion of the transfer.

This process increased delays and workload for sellers.

New solution

The new payment solution needed to increase the automation of the order process; human intervention should no longer be necessary.

This solution will also diversify the payment options. Payments by credit cards (MasterCard & VISA) should be added. This is the most common method of payment used in Belgium (41%³). The next most common method of payment is by debit card (Bancontact & MisterCash). Both payment methods are preferred by the company.

The option to pay by bank transfer could then be deactivated. Accounting sees its work complicated by this payment option because it needs to continuously monitor the movements on the bank account, identifying web orders transfers and verifying information.

Integrated or rented system

There are two ways to manage online payments. The first one is to have your own management system, called an “integrated system”. This system, which requires high security standards (PCI and IPSC standards), was not an option for the company.

It is, however, possible to use service providers who themselves meet the high security standards. It’s called a “rented system”.

³ Comeos survey - June 2014.

5.1 Comparison criteria

There are many solutions in the highly competitive world of payment service providers (PSP). I compared these solutions.

5.1.1 Available payment methods

Payments can be made by credit card (VISA & MasterCard) or by debit card (MisterCash & Bancontact). All of the solutions, except Sofort, allow payment by credit card. Payment by debit card is available in all solutions except for PayPal because it doesn't integrate the Belgian market.

5.1.2 Prices

The prices charged by the various payment service providers (PSP) are divided into three categories.

The first one includes installation costs. These costs vary between € 0 (HiPay) and € 500 (Ogone).

The second category contains monthly fees, between € 0 (HiPay and Sofort) and € 180 (ATOS Sips Advantage). Some providers charge an annual fee (€ 300, Europabank).

The last category covers transaction costs. The payment service provider will charge a fee for each transaction. This amount is calculated by adding a fixed and a variable part.

The fixed part is an amount independent of the amount of the transaction. It is between € 0.17 (ATOS Sips Advantage) and € 0.40 (ATOS Sips Starter). To ensure that this fixed part is paid, there is a minimum amount per sale (€ 1, HiPay).

The variable component is a percentage of the total transaction amount. This percentage varies depending on the company revenue (the sum of sales amount realized with the payment solution) and the payment method used. The percentage decreases according to the revenue. This percentage varies between 3.4% (PayPal with revenue of up to € 2,000) and 0.9% (Sofort).

It would have been possible to combine several solutions. For example, Sofort (that offers the best tariff for payment by debit card) and another solution. However, the benefit on payments by debit card would have been negated by the credit card payments (lowering the total revenue realized on the solution and thus a less advantageous percentage).

Prices charged by the payment service providers are the decisive factor in choosing a solution or another. In the comparison I made, I didn't include installation costs because they are an investment that could be amortized over a shorter or longer period of time, making calculations more complex.

5.1.3 Integration constraints

Payment service providers also impose conditions on companies. The first one concerns URLs of the website. Pyramide Ingénierie had already been contacted by HiPay a few years ago. This spontaneous contact from HiPay could not reach an agreement for two reasons. Firstly, Pyramide Ingénierie was not ready for online sales, and secondly, URLs of the website weren't bookmarkable. Indeed, the session ID was included in the URL, which made them difficult to read for the user and he couldn't bookmark it. When I started to work on the website, URLs were already bookmarkable.

The second constraint is a requirement to display the payment solution. Some providers require that their solution be first on the list of payment solutions on the website. Others may require the use of their payment buttons to maintain consistency between different websites using the solution.

5.1.4 Contact

Some payment service providers I contacted didn't respond. Some responded with great delay.

I didn't have that much time to calculate the costs. I found some costs on internet, but without confirmation, I couldn't include these costs in my comparison.

I had the impression that some companies didn't want to give me all necessary information. I had to insist to be sure there were no hidden costs.

5.2 Comparison

The comparison is based on prices. I calculated costs of sales depending on several parameters. To do so, I made two dynamic arrays. Both tables include different types of costs (monthly fee and transaction fee). I didn't include installation costs because they are a one-time charge which would be amortized over the length of time the service is used.

For my calculations, the revenue is fixed (percentages charged for the transaction depends on the revenue). For the first calculation, the number of transactions varies. In the second, the average amount per transaction varies.

The tables include a lot of data, making them difficult to read. To simplify interpretation, I created two graphs taking each solution and its cost in terms of revenue.

The graphs include only solutions for which I received prices from the payment service provider itself (posted on their website or obtained in a phone or email). Thus, the solution "Ogone" and "Mercanet" are not included. In order to facilitate the reading of information, uncompetitive solutions have been omitted. The solution "ATOS Sips Advantage" is not included for this reason.

In both graphs, the horizontal axis is the total revenue achieved with the payment solution and the y-axis is the total cost per transaction in Euros.

For all solutions offering payment by credit and debit cards, cost for debit card is lower than for credit card.

Priority was given to credit cards because it is the most used payment method. When I compare tariffs of various solutions, I will compare their costs for payment by credit cards.

Figure 20 shows the change in total cost per transaction depending on the revenue for a total of 100 transactions. The solution “HiPay Direct”, in blue, is the most competitive (for payment by credit cards) with revenue under € 4,000. With higher revenue, the solution of PayBox is more competitive.

The solution of Sofort is the least expensive solution, but it is limited to debit cards.

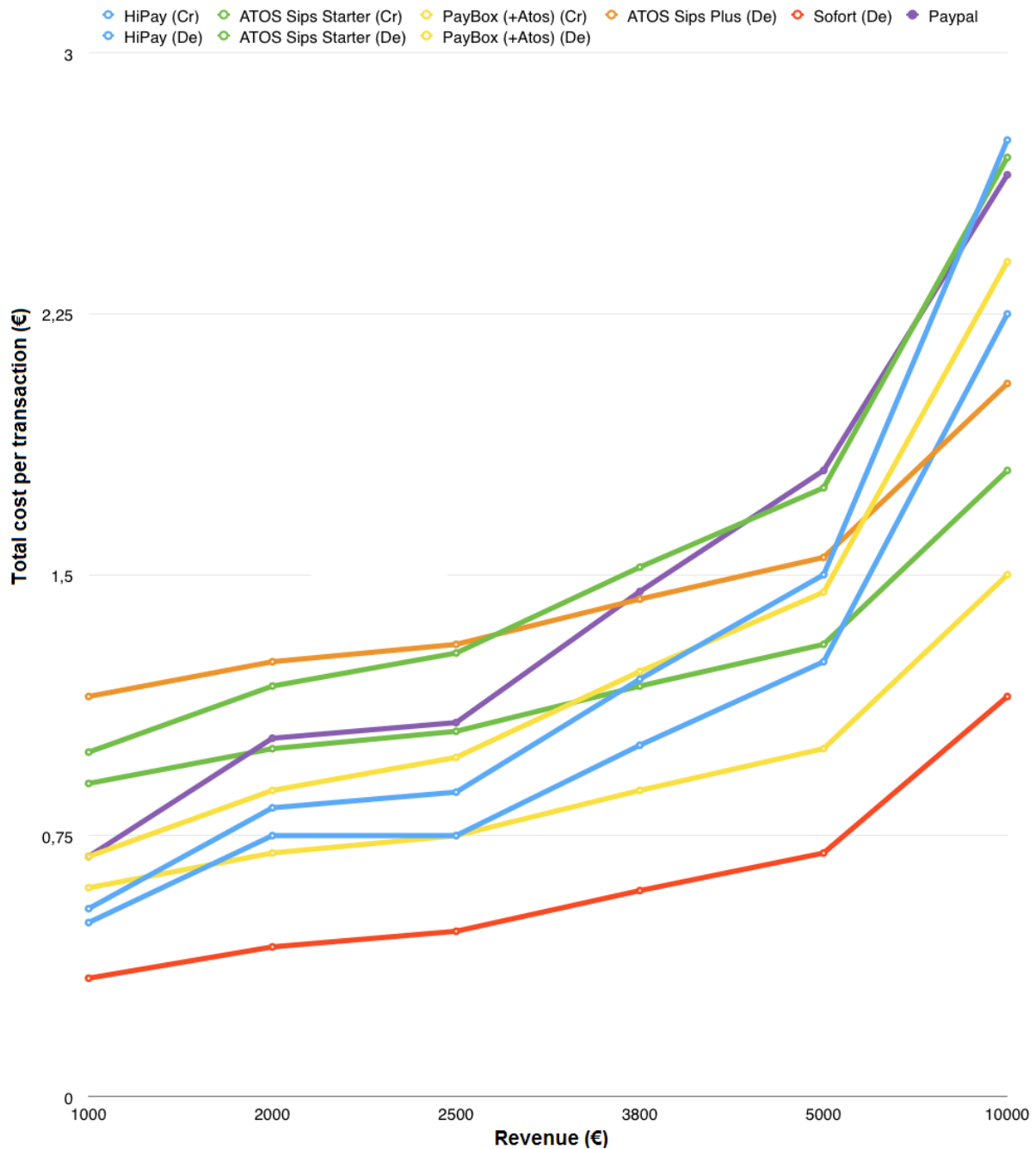


Figure 20 – Graph representing the change in the total cost per transaction depending on the revenue for a total of 100 transactions

Figure 21 confirms what the figure 20 said, that the solution “HiPay Direct” is the most attractive (for payment by credit cards) for revenue under € 4,000.

Figure 21 shows the comparison for a basket of an average amount of € 1,000. This amount corresponds to the sale of an iPhone⁴.

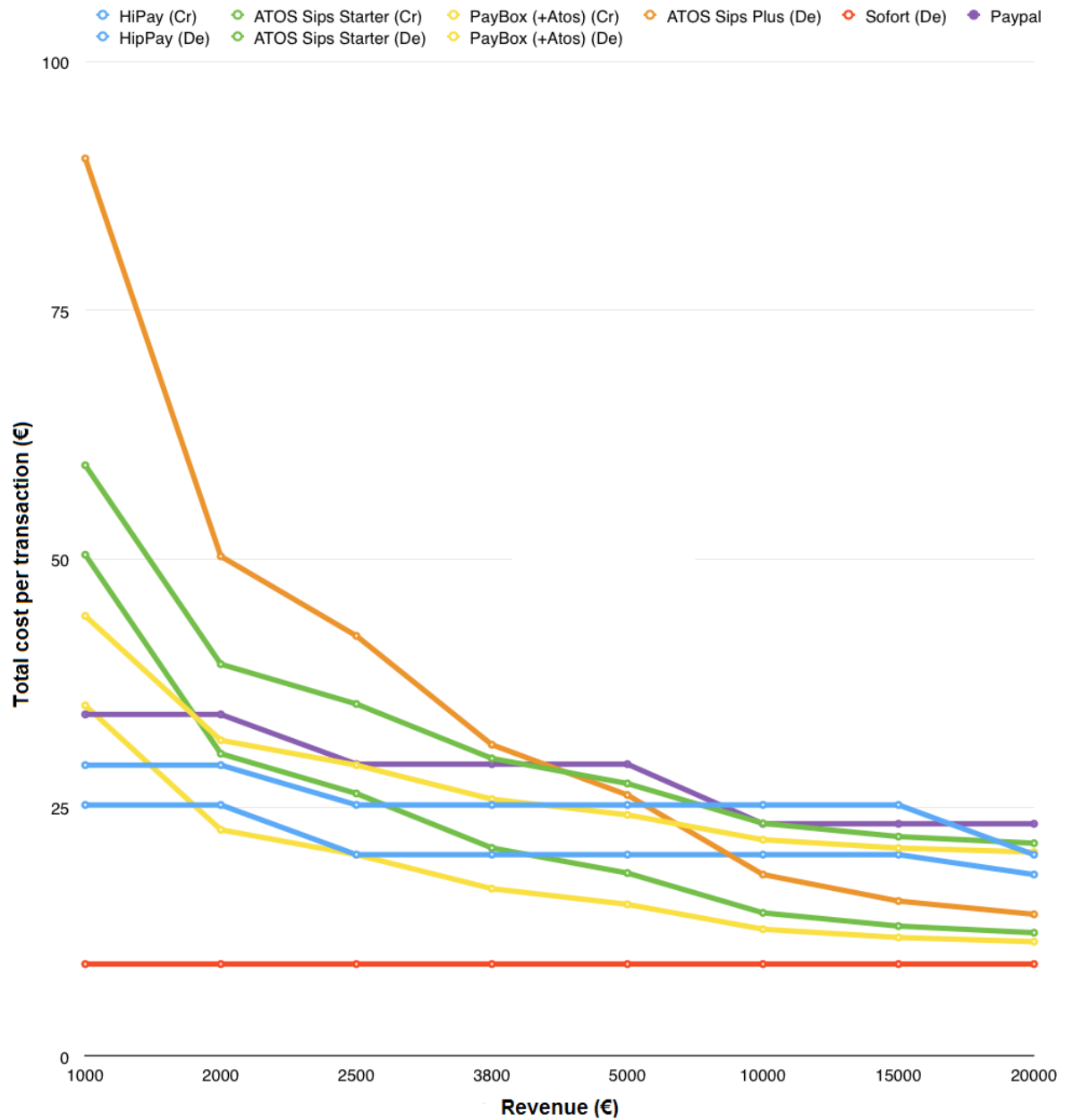


Figure 21 – Graph representing the change in the total cost per transaction depending on the revenue for a basket of an average amount of € 1,000

⁴ The price of the “iPhone 6 Plus 128GB” is € 999 on the cipiyou store website.

5.3 Choosing HiPay

5.3.1 HiPay Direct

The chosen solution is “HiPay Direct” proposed by the company HiPay.

HiPay has three solutions: “HiPay Wallet”, “HiPay Direct” and a premium offering.

The solution “HiPay Wallet” provides an electronic wallet. The money of the transaction is transferred to an account belonging to HiPay. The electronic wallet of Pyramide Ingénierie is then credited with the amount of the sale less the cost of the HiPay transaction. The money available on our electronic wallet can be transferred once a week to the company bank account. With the solution “HiPay Wallet”, the transaction amount is limited to €2,500.

The solution “HiPay Direct” also uses an electronic wallet, but without the restriction of maximum € 2,500 per transaction. It was not possible for us to accept such a limitation because the prices of some products exceed it⁵.

The Premium offering is only available after a first using one of the HiPay solutions a few times. HiPay would then contact certain companies to offer them the Premium option. I was not able to determine what criteria they use when offering this option.

The use of an electronic wallet allows HiPay to charge preferable tariffs. For solutions that do not use an electronic wallet, there is an additional cost of about 0.9% because an Atos solution is required (in addition to the chosen solution). HiPay solutions simplify the integration work.

⁵ The price of a MacPro is at least of € 3,049 on the cipiyou store website.

5.3.2 HiPay, a prudent choice

The revenue that will be achieved on the website is difficult to estimate.

There is no perfect solution, no solution is best for all revenues. HiPay solutions are less costly for small revenue and there are no monthly fees or installation costs. If the revenues greater than € 4,000, it will still be possible to change to another solution.

HiPay is thus a prudent choice made to assess the potential sales on the website based on statistics (that said that payment by credit cards is the most used method of payment in Belgium).

Furthermore, I got quick answers to my questions and no questions were evaded, either by email or phone.

Chapter 6: Integration of the solution “HiPay Direct”

In this chapter, I'll cover the integration of the solution “HiPay Direct” in the cipiyou-web application.

I will start with the terms and conditions, which are essential for online selling. I will then discuss the life cycle of a payment and the various steps of the communication between the application and HiPay.

6.1 Terms and conditions

In addition to the technical changes (discussed in the next section), the integration of online sales involves changes to the cipiyou store. The main change concerns terms and conditions. These must respect consumer rights while defining those of the seller.

To develop the terms and conditions of Pyramide Ingénierie⁶, I compared those of several other businesses practicing online sales. I defined several categories and important points, and I wrote new terms and conditions. I couldn't use the terms and conditions of other websites because it would be plagiarism. It would make our terms and conditions unusable and the company could be sued.

This section presents the various categories that emerged from the comparison.

General terms:

This first part sets the context of the following conditions. It is possible to report that they apply to online sales, the customer agrees to read these and they may change without notice.

The terms and conditions that apply to a sale are those that were valid when the customer placed his order.

If the terms and conditions are not provided in the customer's language, they will still be valid.

Validity of offers:

This section sets the limits of the validity of prices on the website. If a human error or technical failure occurs in the price, the seller may cancel the concerned orders.

Delay:

This part allows the seller to specify the normal delivery times and to protect him if the delay is exceeded.

Pickup in-store:

The customer is informed of the requirement to pick his order up in-store. His identity will be verified and a copy of his documents may be used to increase security (data won't be used for other purposes than security).

Delivery - transportation:

Almost the same as "pickup in-store", but for delivery at a customer address.

In addition, the customer is informed that any apparent defect must be reported immediately to the delivery service.

⁶ See Attachment 2: "Terms and conditions of Pyramide Ingénierie".

Security interest clause:

The seller keeps ownership of the product until the customer has paid for his order in full. However, the customer becomes responsible for any damage to the product upon receipt of the order.

Prices:

This section is used to define labels and taxes (VAT, Recupel ...) applying to the displayed prices.

Payment:

The different payment methods are described in this part.

Cancellation right:

This section is only valid for online sales. The right of cancellation allows the customer to cancel a purchase if he couldn't examine the product prior to purchase. This right can be exercised for all online sales, even if the product is available in a physical shop. The customer has 14 days to claim his cancellation right.

This right does not apply to certain products (software and audio or video recordings whose seal has been broken, damaged items, clearly personal items ...). This list of product types is displayed in this section.

This section also informs the customer revocation process.

Warranty:

All products purchased for a private use come with a two-year warranty.

This section informs the customer about the limits of this warranty and the actions that could lead to a cancellation of it (non-authorized modifications, unauthorized use, water damage ...).

Conflicts:

In case of disputes, the seller advises that the terms and conditions are subject to Belgian law and must be resolved in the courts that have jurisdiction at the location where the company is based.

Severability clause:

This clause says that if a clause was invalidated, this would not cause the invalidation of others.

6.2 Technical modifications

6.2.1 Payment Life cycle

The payment process can be divided into several stages.

0. Fill the basket

During this stage, the customer fills his basket, selects a shipping method⁷ and possibly uses a coupon⁸. The customer can launch the payment process by clicking on the “Pay” button.

1. Call to HiPay web service

Figure 22 shows a sequence diagram representing the call to the HiPay service web.

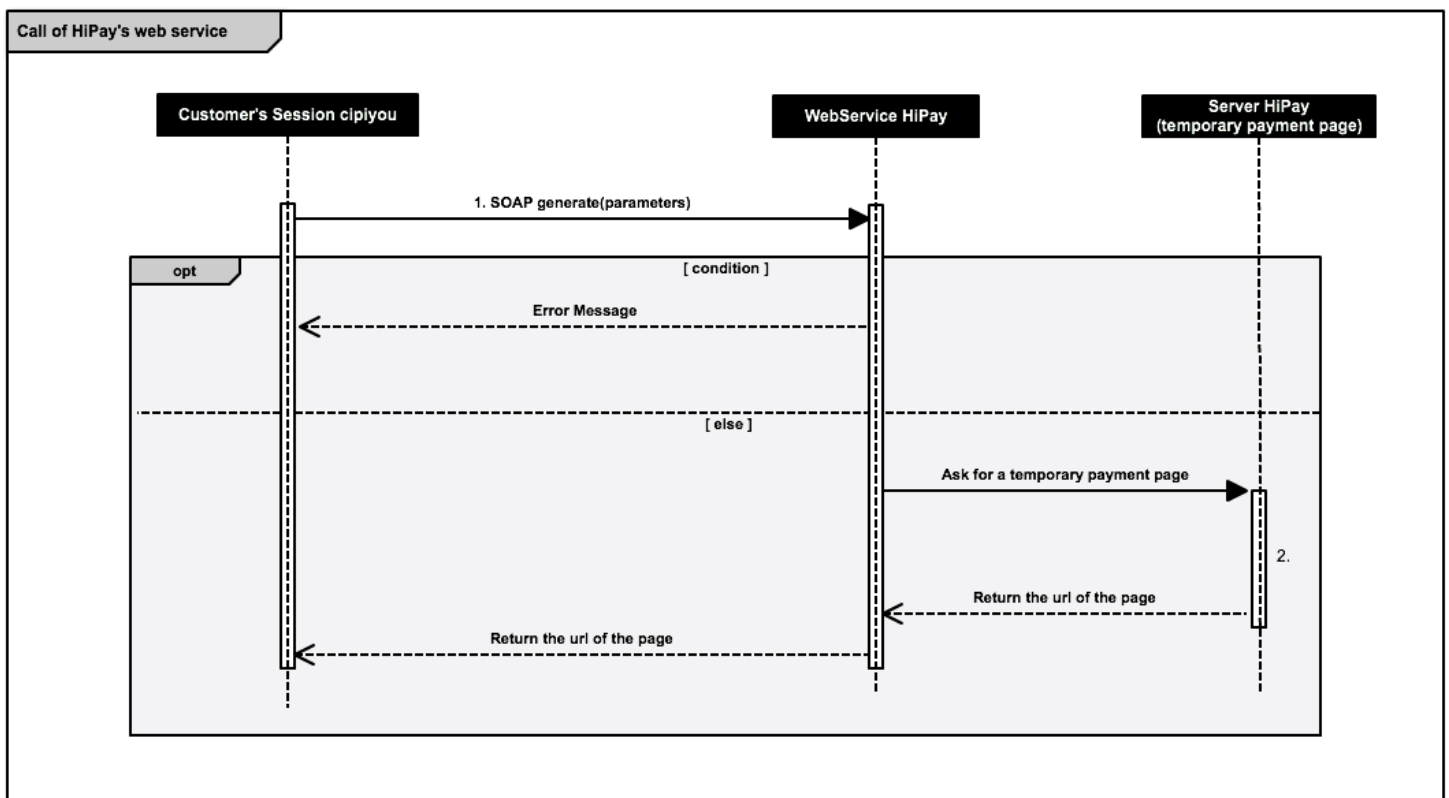


Figure 22 – Sequence diagram representing the call to the HiPay service web

⁷ See Part II, Chapter 3: « Delivery ».

⁸ See Part II, Chapter 4: « Discount coupons ».

A SOAP (Simple Object Access Protocol) request “generate” is sent to the HiPay server. Figure 23 provides an example of a SOAP request provided by HiPay in its documentation.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns1="https://test-ws.hipay.com/soap/payment-v2">
  <SOAP-ENV:Body>
    <ns1:generate>
      <parameters>
        <websiteId>345</websiteId>
        <categoryId>225</categoryId>
        <currency>EUR</currency>
        <amount>20</amount>
        <rating>ALL</rating>
        <locale>fr_FR</locale>
        <customerIpAddress>127.0.0.0</customerIpAddress>
        <merchantReference xsi:nil="true"/>
        <description xsi:nil="true"/>
        <executionDate>2012-12-10T18:00:00+00:00</executionDate>
        <manualCapture>false</manualCapture>
        <customerEmail xsi:nil="true"/>
        <merchantComment xsi:nil="true"/>
        <emailCallback>callback@mydomain.com</emailCallback>
        <urlCallback>http://www.exemple.com/callback.php</urlCallback>
        <urlAccept>http://www.exemple.com/ok.php</urlAccept>
        <urlDecline>http://www.exemple.com/ko.php</urlDecline>
        <urlCancel>http://www.exemple.com/cancel.php</urlCancel>
        <urlLogo>http://www.exemple.com/logo.jpg</urlLogo>
        <freeData>
          <item>
            <key>keyOne</key>
            <value>aValueOne</value>
          </item>
          <item>
            <key>keyTwo</key>
            <value>aValueTwo</value>
          </item>
        </freeData>
        <wsLogin>4f8c3e80fd4e476f1e667c68fefxx3e5</wsLogin>
        <wsPassword>2f0x4099x4254f819xx7246d6202d192</wsPassword>
        <wsSubAccountLogin xsi:nil="true"/>
      </parameters>
    </ns1:generate>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 23 – Example of a SOAP request “generate” provided by HiPay in its documentation.

It includes, among other information:

- Customer information (his/her email, address, name ...);
- Payment information (amount, currency, order number ...);
- Information specific to Pyramide Ingénierie (access code (hashed with MD5⁹) to the web service, merchant account information, redirection URLs ...).

The “freeData” contains specific information for the current payment, such as the shipping address and order contents.

⁹ HiPay uses MD5 to hash the data and don’t provides alternative (such as SHA-256 for example).

Figure 24 shows a sequence diagram of the payment process once the application has received the redirection URL from HiPay. Steps 2 to 6 are indicated therein.



2. Creation of a temporary payment page

When the HiPay receives the SOAP request (and if the information is consistent), a temporary payment page is generated by HiPay. This page has a unique URL and will expire after a few minutes. The URL of this temporary payment page is returned in response to the SOAP request.

3. Redirection of the customer to the temporary payment page

The application receives the redirection URL and, using a WOREdirect, directs the customer to the payment page. At this time, the customer is no longer on the cipyoy store website. The application has no knowledge of data entered by the customer.

4. Payment

On the temporary payment page, the customer can proceed to payment. Three possibilities can occur.

Payment can be abandoned or cancelled. For example, the customer can cancel the transaction or the temporary page can expire.

Secondly, the payment can cause an error. For example, if the customer card is blocked or he doesn't have enough money, it will cause an error.

Finally, the payment can proceed normally.

5. Redirection on the cipyoy store website

The customer is redirected from the HiPay website to the cipyoy store website. The different redirection URLs generated in step 1 are used at this time.

If payment has been abandoned or causes an error, the customer is redirected on an error page. If the payment was successful, the customer will be redirected to a confirmation page. I created those two pages when I began to use redirection URLs.

The customer session is passed to the redirection URLs so that he's still connected when he is redirected to the cipyoy store website.

6. Calling the call-back URL

HiPay sends two HTTP request (POST) with an XML stream to the call-back URL (passed in step 2). This URL is a DirectAction on the website and an action is effected when the request is received.

Figure 25 shows an example of XML stream sent by HiPay after payment.

```
<?xml version="1.0" encoding="UTF-8"?>
<map>
  <mapversion>1.0</mapversion>
  <md5content>b8d242380bd35ba37e78cc46d41fc112</md5content>
  <result>
    <operation>capture</operation>
    <status>ok</status>
    <date>2015-05-27</date>
    <time>09:00:26 UTC+0000</time>
    <origAmount>1.00</origAmount>
    <origCurrency>EUR</origCurrency>
    <idForMerchant>OC-256236</idForMerchant>
    <emailClient>pierre.nyssen@pyramide-ingenierie.be</emailClient>
    <idClient>173328</idClient>
    <merchantDatas>
      <_aKey_idOffreCli>OC-256236</_aKey_idOffreCli>
      <_aKey_adrNom>NYSSSEN</_aKey_adrNom>
      <_aKey_adrPrenom>Pierre</_aKey_adrPrenom>
      <_aKey_adrRue>Rue du Brou</_aKey_adrRue>
      <_aKey_adrNumero>83</_aKey_adrNumero>
      <_aKey_adrCodePostal>4800</_aKey_adrCodePostal>
      <_aKey_adrLocalite>VERVIERS</_aKey_adrLocalite>
      <_aKey_adrPays>BELGIQUE</_aKey_adrPays>
      <_aKey_liv>CPU</_aKey_liv>
    </merchantDatas>
    <cardCountry>N/A</cardCountry>
    <ipCountry/>
    <transid>5565879F5FC01471</transid>
    <is3ds>No</is3ds>
    <paymentMethod>DEXIA</paymentMethod>
    <refProduct0/>
    <refProduct1>MD818ZM/A</refProduct1>
    <customerCountry>BE</customerCountry>
    <returnCode/>
    <returnDescriptionShort/>
    <returnDescriptionLong/>
  </result>
</map>
```

Figure 25 – Example of XML stream sent by HiPay after payment

The XML stream is received via the method "request()". It will be analyzed to obtain certain information.

There are two types of requests: "authentication" and "capture". The authentication request is sent when the customer's bank authorizes the transaction. The capture request is sent when the transaction is executed. The order is executed by the application if the status is "OK".

The authentication request is ignored by our application. The capture request initiates the action.

The treatment of the request consists of retrieving the customer's address and the offer number (passed with freeData in the XML stream), and creating an order in the database. This order is automatically validated, which means the cipiyou staff can now treat it.

During tests, the delay between payment of the order and the receipt of the call-back URL generally took a few seconds, but for one order took four weeks.

I couldn't determine why it took so much time, but such a delay could occur again. A confirmation email is sent to the customer when the capture request is received by the server. If the customer doesn't receive this email, he can contact the cipiyou team.

It is possible to check payment status on the HiPay website. It is also possible to verify if the order has been paid or not and, if necessary, to validate the order without the HiPay confirmation.

6.2.2 Difficulty: Merchant kit in PHP

HiPay provides a merchant kit to easily integrate its solutions in website, but it is written in PHP and it wasn't possible to use it on our website.

To create an equivalent service, I contacted the HiPay web service to get a basic code using WSDL2Java. After adapting the code, I could communicate with the web service and create correct queries.

6.2.3 Difficulty: Test ⇔ Production

Once we were sure the order system was operational, we could put the application into production. HiPay provided a new version of the documentation for their production server. This new version was much more complete than the old one, and if I had obtained it earlier, the work would have been simplified.

Unfortunately, there were differences between the two versions of the documentation, particularly in data transmission by HiPay. The biggest change was in the content of the MD5 verification. The content of the hashed data was supposed to be different. After one test of a real payment (and thus a loss of money to proceed), I realized it was not the case and that the new version contained some mistakes.

Conclusion - Part II

Improvements in delivery

Several ideas have been proposed for improving the delivery part. The main part of this improvement concerns the management of customer addresses.

The idea consists of keeping all the addresses used by the customer. These addresses can be active or inactive. The customer could manage his addresses by deactivating an address from his profile.

When choosing a delivery method, the customer could choose:

- Pickup at cipiyou;
- Delivery to the main address;
- Delivery to a secondary address.

Selecting the third choice would allow him to choose one of his active addresses or to enter a new one. One option might allow him to make the selected secondary address his new main address.

To allow this advanced management, the database would have to be adapted to include the address table and the relationship between this table and the others. All references to addresses should be adjusted to point to this new table.

Figure 26 shows a diagram of the database including changes to make in it.

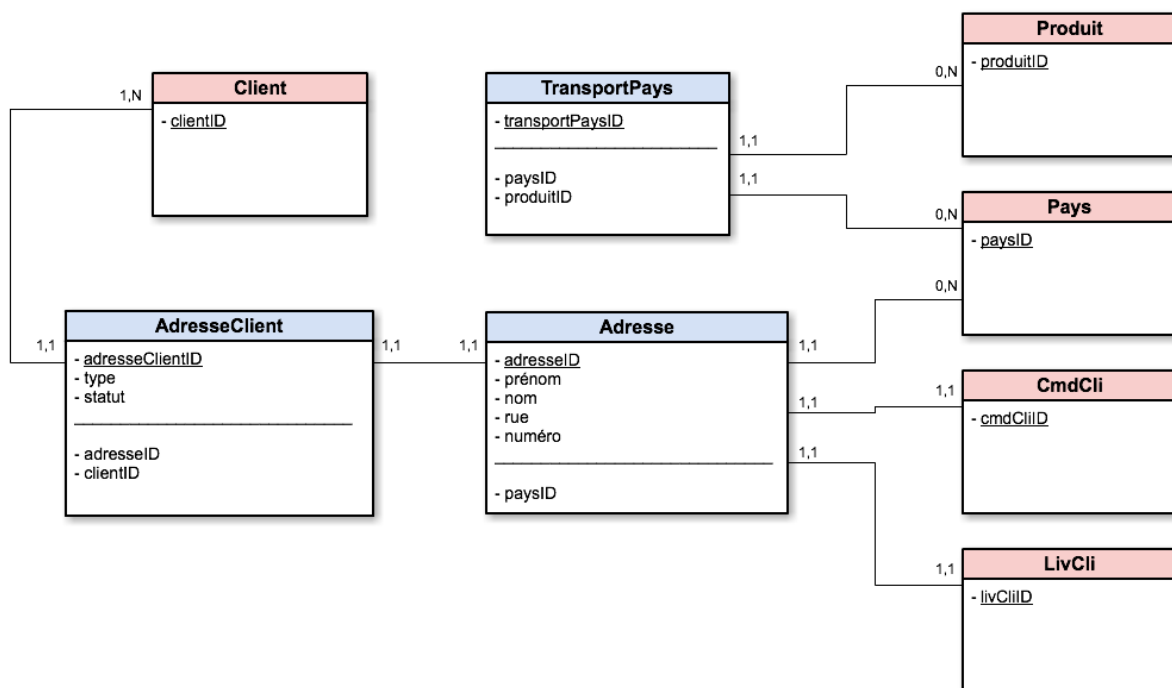


Figure 26 – Diagram of the database adapted for the addition of an Address table

Another improvement would be to change the management of delivery country. For now, the description of a transport product contains the country names but it would be more convenient for product management to create a table of country.

Improvements of coupons

There is a type of coupon I couldn't implement: coupons with a number of uses (no matter which customer uses it).

For example, a coupon could be used by the next ten customers. A column for a number of utilizations could be added.

But the verification of this condition is very complicated. If a client validates the code, he should be able to use it, but if he doesn't, the code can't be blocked.

It's still a problem to be solved.

Selecting a payment solution

To choose the solution "HiPay Direct", I had to compare many solutions on several criteria. The most important criterion was the cost of the service.

I had the opportunity to communicate with Customer Service Representatives of several companies; I had to be very patient to get the answers I wanted.

I made a comparison between the solutions and we choose the solution "HiPay Direct".

Integration of the payment solution

Once the solution was selected, I had to integrate it into the website. HiPay provides a merchant kit, but it is coded in PHP and it wasn't possible to use it on our website.

I created an equivalent service in java to communicate with the HiPay web service.

I also received several versions of the documentation for HiPay web service.

Tests of payment were performed to ensure that the payment procedure was correctly developed.

Conclusion

During my internship, I managed to create an order processing system for the online sales of cipiyou.

I evolved in different environments from those I knew because I was able to work in a company, outside the school context (although it remained present) and because I did not know about the Apple environment.

I discovered new technologies (mainly WebObjects) and I became more and more proficient at it during the four months of internship. I started by writing a modest application – cipiyou-Account – and ended up working on more complex applications – cipiyou-web and cipiyou-manage.

I was able to design and integrate different features. I have developed an application from a limited one to a complete solution. I enabled delivery to different addresses; I replaced the payment by bank transfer system to a complete solution with many payment methods and I did create a discount coupons system.

I grew in a professional setting of SMEs that inspired me to move forward and encouraged me to take part in business life. I felt part of a team during this internship. This experience has reinforced my desire to move forward by joining a company. I would like to join a larger company, where I might have the opportunity to work on more projects, grow my knowledge and acquire more experience.

Bibliography

Book:

HILL, Charles MALLAIS, Sacha, 2004. Practical WebObjects. New-York: Apress. 364p.

Blog posts:

LEBERT, David, 2004. Using WebObjects DirectActions pt.1 [online]. 14-10-2004 10:33. [Accessed 02-03-2015]. Available at: <http://davidleber.net/using-webobjects-directactions>

LEBERT, David, 2004. Using WebObjects DirectActions pt.2 [online]. 22-10-2004 16:46. [Accessed 02-03-2015]. Available at: <http://davidleber.net/using-webobjects-directactions-pt-2/>

LEBERT, David, 2004. Using WebObjects DirectActions pt.3 [online]. 25-11-2004 13:06. [Accessed 02-03-2015]. Available at: <http://davidleber.net/using-webobjects-directactions-pt-3/>

LEBERT, David, 2004. Using WebObjects DirectActions pt.4 [online]. 26-11-2004 14:44. [Accessed 02-03-2015]. Available at: <http://davidleber.net/using-webobjects-directactions-pt-4/>

LEBERT, David, 2004. Using WebObjects DirectActions pt.5 [online]. 15-12-2004 15:50. [Accessed 02-03-2015]. Available at: <http://davidleber.net/using-webobjects-directactions-pt-5/>

PDF File:

Apple, 2007. WebObjects Web Applications Programming Guide (Legacy) [PDF file]. 2007-07-11. [Accessed 04-02-2015]. Available at:

https://developer.apple.com/legacy/library/documentation/WebObjects/Web_Applications/Web_Applications.pdf

BNP PARIBAS, 2013. Guide des principaux produits, services et tarifs [PDF file]. 11-2013. p. 10. [Accessed 23-02-2015]. Available at:

http://www.bnpparibas.net/banque/PA_CanalnetApp/documentum/canalnet/public/PDF/

Tarif_pro_2014_20131230035126.pdf AWT, 2012. E-Commerce Quels moyens de paiement choisir ? [PDF file]. 22-10-2012. [Accessed 23-02-2015]. Available at:

http://www.awt.be/contenu/tel/ebu/221212_quelsmoyensdepaiement.pdf

AWT, 2012. Commerce électronique: réglementation et contrôle [PDF file]. 28-11-2012. [Accessed 23-02-2015]. Available at:

http://www.awt.be/contenu/tel/ebu/SPF_economie_e-commerce.pdf

CM CIC, 2014. Paiement sécurisé sur Internet [PDF file]. 03-2014. [Accessed 23-02-2015]. Available at:

https://www.cmcicpaiement.fr/fr/info/documentations/CMCIC_paiement_documentation_technique_v3_0b.pdf

Web pages:

ATOS WORLDLINE, 2013. Sips: la plateforme de paiement en ligne pour votre e-boutique. Worldline e-Payment services [online]. [Accessed 23-02-2015]. Available at:

https://masolutiondepaiement.be/index/fr_FR/0000/6777814/Sips:-la-plateforme-de-paiement-enligne-pour-votre-e-boutique.htm

PAYPAL, 2014. PayPal Intégral, la plateforme de paiement complète. PayPal [online]. [Accessed 23-02-15]. Available at: <https://www.paypal.com/fr/webapps/mpp/standard>

PAYPAL, 2014. Existing Developer Products - PayPal Developer. PayPal [online]. [Accessed 23-02-2015]. Available at: <https://developer.paypal.com/webapps/developer/docs/classic/products/>

EUROPABANK, 2015. E-commerce - Europabank. Europabank [online]. [Accessed 26-02-2015]. Available at: <https://www.europabank.be/ecommerce-professionnel>

INGENICO PAYMENT SERVICES, 2015. De quoi ai-je besoin pour démarrer la vente en ligne. Ingenico Payment Services [online]. [Accessed 23-02-2015]. Available at:

<http://payment-services.ingenico.com/be/fr/support/besoins-commencer-vente-en-ligne>

INGENICO PAYMENT SERVICES, 2015. Ingenico Payment Services - Acceptez les paiements à l'international Belgique. Ingenico Payment Services [online]. [Accessed 23-02-2015]. Available at:

<http://payment-services.ingenico.com/be/fr/solutions-paiement-en-ligne/solution-moyen-paiementen-ligne-international#>

Solution SOFORT. SOFORT, 2015. Tarifs - SOFORT AG. SOFORT AG [online]. [Accessed 24-02-2015]. Available at: <https://www.sofort.com/fre-BE/commerçants/sb/tarifs/>

AGENCE WALLONNE DES TELECOMMUNICATIONS, [sans date]. E-Commerce. Rappel des réglementations économiques à respecter. awt.be [online]. Mis à jour le 08-02-2013. [Accessed 23-02-2015] <http://www.awt.be/web/ebu/index.aspx?page=ebu,fr,foc,100,202>

Websites:

SPF ECONOMIE, 2015. SPF Economie [online]. [Accessed 13-03-2015]. Available at: <http://economie.fgov.be/fr/>

JACOB, Damien, 2015. Retis [online]. [Accessed 24-02-2015]. Available at: <http://www.retis.be>

PCI Security Standards Council, 2015. PCI Security Standards Council [online]. [Accessed 15-03-2015]. Available at: <https://www.pcisecuritystandards.org>

VISA, 2015, Visa [online]. [Accessed 15-03-2015]. Available at:

<http://usa.visa.com/merchants/protect-your-business/cisp/merchant-pci-dss-compliance.jsp>

Attachments

1: Tariff Grid for parcel delivery from Belgium by GLS

2: Terms and conditions of Pyramide Ingénierie (in French)

1: Tariff Grid for parcel delivery from Belgium by GLS

EuroBusinessParcel		Tarif 2015						GLS	
Offre du: 01/01/2015		à: PYRAMIDE INGENIERIE-7269856							
Tarif valable jusqu'au: 31/12/2015		Point de départ: Verviers							
Prix/Colis	5 kg*	10 kg*	15 kg*	25 kg*	30 kg*	40 kg*	50 kg*	RTT ¹⁾	
Luxembourg (G.D. de)	9,48	11,92	14,35	18,11	20,89	26,59	29,87	1	
Pays-Bas	9,48	11,92	14,35	18,11	20,89	26,59	29,87	1	
Allemagne	9,97	12,55	15,12	21,97	28,00	31,43	35,70	2-3	
Autriche	13,43	15,11	17,64	21,67	28,20	33,42	37,78	2	
Bulgarie ²	82,12	120,43	160,32	238,50	291,73	354,80	446,47	4	
Chypre (UE) ²	83,30	85,43	89,14	131,04	152,87	222,98	261,31	5	
Danemark (exclu Îles Féroé et Groenland)	11,68	14,26	15,96	22,82	28,85	32,28	39,16	2	
Espagne	36,58	39,59	43,94	48,49	53,03	57,54	62,09	3	
Baléares	68,03	73,18	80,82	97,82	101,39	118,42	135,61	4	
Îles Canaries, Melilla, Ceuta ^{2 3}	104,68	106,39	119,17	150,61	167,64	199,02	230,14	5	
Estonie	33,10	36,72	39,86	44,27	48,66	51,96	57,26	4-5	
Finlande	36,08	39,67	42,90	47,22	51,64	54,98	60,23	4-5	
France	16,75	18,11	21,13	26,29	31,10	42,36	44,80	2-3	
Corse	81,58	99,16	133,09	145,94	150,25	154,55	158,83	3	
Grande-Bretagne	17,52	22,32	24,23	28,94	31,94	41,20	44,65	2-3	
Îles du canal/Irlande du Nord/Îles de Man ³	55,80	60,08	64,40	72,97	78,98	84,14	88,44	3-4	
Grèce ²	68,02	71,44	74,89	116,10	137,57	206,22	244,88	4-5	
Îles	68,02	71,44	74,89	116,10	137,57	206,22	244,88	5-7	
Hongrie	28,92	33,10	37,46	46,09	50,63	53,63	55,90	3	
Irlande	34,34	38,63	42,90	47,22	49,80	54,08	58,38	3	
Italie (Sicile et Sardaigne inclus) ³	30,92	34,34	37,78	40,34	44,65	47,22	49,80	3-4	
Lettonie	33,10	36,72	39,86	44,27	48,66	51,96	57,26	4-5	
Lituanie	33,10	36,72	39,86	44,27	48,66	51,96	57,26	4-5	
Malte ²	53,06	92,16	132,88	212,63	251,69	331,45	411,22	4	
Norvège ³	36,08	39,67	42,90	47,22	51,64	54,98	60,23	3-6	
Pologne	26,63	30,66	34,91	43,25	47,68	50,60	52,80	3	
Portugal	27,40	31,56	35,96	44,56	49,12	52,10	54,40	3-4	
Açores	81,37	89,18	102,85	126,90	141,48	152,68	177,49	5	
Roumanie	39,01	42,10	45,17	49,43	55,58	60,00	67,76	3	
Slovaquie	28,92	33,10	37,46	46,09	50,63	53,63	55,90	3	
Slovénie	28,92	33,10	37,46	46,09	50,63	53,63	55,90	3	
Suède	36,08	39,67	42,90	47,22	51,64	54,98	60,23	3-6	
Suisse ³	14,24	15,78	17,93	26,14	29,75	34,27	40,56	3	
Tchéquie	27,40	31,56	35,96	44,56	49,12	52,10	54,40	2	
Turquie ^{2 3}	46,26	66,97	90,14	127,92	150,25	188,04	225,80	5	

* Le poids du paquet le plus lourd défini la catégorie du poids appliqué.

Figure 27 – Tariff Grid for parcel delivery from Belgium by GLS

2: Terms and conditions of Pyramide Ingénierie (in French)

Les conditions générales suivantes sont consultables sur le site « cipiyou.be ».

Les conditions de ventes citées ci-après sont d'application pour toute commande en ligne passée sur le site Web. Le client est tenu de lire ces conditions de vente avant de passer une commande sur le site. Il devra accepter expressément ces conditions lors de toute commande, sans quoi la transaction ne saurait être finalisée.

1. Dispositions générales

1.1 Identification et siège social du représentant :

Pyramide Ingénierie sprl dénommée ci-après « Pyramide Ingénierie ».

Rue du Brou 83, 4800 Verviers (Belgique)

Tel. 087/29.21.20

E-mail shop@cipiyou.be

Pyramide Ingénierie est enregistrée à la Banque Carrefour des Entreprises au numéro 0436.881.565.

1.2 Sauf dérogation expresse écrite convenue entre les parties, les relations entre Pyramide Ingénierie et ses clients sont soumises aux présentes conditions de vente. Ces conditions sont réputées admises par le client, en ce compris dans l'hypothèse où elles entreraient en contradiction avec ses propres conditions générales ou particulières d'achat.

1.3 Pyramide Ingénierie se réserve le droit d'apporter des modifications à ces conditions de vente sans en avertir personnellement ses clients. Il est de la responsabilité du client de vérifier les conditions de vente avant chaque commande. Sur des contrats en cours, les conditions en vigueur lors du moment de la vente resteront en vigueur.

1.4 Le fait pour le client de ne pas avoir reçu les présentes conditions générales dans sa langue maternelle ne le dispense aucunement de leur application.

2. Validité des offres

Sauf mention contraire, les offres et listes de prix sont valables aussi longtemps qu'elles figurent dans notre catalogue online. Pyramide Ingénierie se réserve le droit de modifier à tout moment les caractéristiques de ses offres et listes de prix. Pyramide Ingénierie se réserve le droit de refuser une commande si le prix et/ou les spécifications d'un produit mentionné(s) sur le site Web s'avère erroné suite à une erreur humaine ou une défaillance technique.

Pyramide Ingénierie se réserve le droit de refuser des commandes dans les cas suivants :

- en cas de graves soupçons de mauvaise foi ou d'intention d'escroquerie;
- en cas d'abus répété du droit de rétractation;
- en cas de suspicion d'abus (de droit) répété ou de fraude dans le chef d'un client;

- en cas de communication de données erronées, suspectes ou incorrectes;
- si Pyramide Ingénierie constate qu'une personne fournit des données fictives à plusieurs reprises;
- si Pyramide Ingénierie constate une quantité de commande anormale — éventuellement dans plusieurs ordres d'achat — qui dépasse une utilisation strictement personnelle;
- s'il peut être présumé que des commandes ont été effectuées à des fins de revente;
- en cas de force majeure.

3. Commandes

Pour commander un produit en ligne, le client doit l'ajouter à son panier. Ensuite, il choisit la méthode de livraison. Il peut alors procéder au paiement.

Toute commande qui nous est confiée ne nous engage qu'après réception de l'e-mail de confirmation.

4. Délais

Les délais de livraison ne sont pas contraignants pour le contrat.

Sauf si les parties en disposent autrement concernant le moment de la livraison, notamment dans le cas d'une précommande, commande ou lorsqu'un des produits de la commande n'est pas en stock, Pyramide Ingénierie aspire à expédier la commande endéans les 2 jours ouvrables. En cas de manquement de Pyramide Ingénierie à l'obligation de livraison dans ce délai, un délai supplémentaire, adapté aux circonstances, sera convenu avec le consommateur.

5. Retrait chez cipiyou

Les commandes « Retrait chez cipiyou » sont à retirer chez cipiyou (Rue du brou, 88, 4800 Verviers (Belgique)). Le client accepte de montrer sa carte d'identité ou passeport avant le retrait. Pyramide Ingénierie se réserve le droit de transmettre les produits uniquement si le client montre sa carte d'identité ou passeport valable, si nécessaire avec une procuration originale et signée par la personne qui a fait la commande. En principe, c'est la carte d'identité de la personne qui a fait la commande qui doit être montrée ou celle du représentant légal si la commande a été faite par une personne morale. Afin de prévenir les fraudes de paiement et pour la vérification de fraude dans le cas d'une enquête judiciaire, Pyramide Ingénierie est autorisée à prendre une copie de la carte d'identité ou preuve d'identité.

Les produits restent disponibles pendant 14 jours calendrier à compter du moment où la commande est prête en magasin, après quoi la commande sera annulée. Un remboursement pourra être effectué à la demande du client. Des frais administratifs seront déduits du montant à rembourser.

6. Livraisons - transports

La livraison a lieu à l'adresse indiquée par le client lors de la commande. Notre expéditeur (GLS) ne délivrera les marchandises que contre production d'un document d'identité officiel et signature par le client d'un bon de livraison pour réception conforme. Tout vice apparent du produit doit être signalé sur le bon de livraison et celui-ci doit être renvoyé à Pyramide Ingénierie (shop@cipiyou.be) endéans les 2 jours.

7. Clause de réserve de propriété

Par dérogation à l'article 1583 du Code civil, Pyramide Ingénierie reste propriétaire des produits vendus jusqu'au jour du paiement total du prix en principal et/ou intérêt(s) et/ou indemnité(s) forfaitaire(s). Tous les risques de perte et de dégâts relatifs aux biens concernés sont transférés au client dès la prise en possession.

Par la présente, le client fait cession au vendeur de toutes sommes qui lui seraient redevables par des tiers. Le client décharge Pyramide Ingénierie de toute formalité de signification et supportera les frais éventuels y afférents. En outre, le client autorise Pyramide Ingénierie à reprendre le matériel à chaque moment, tant qu'il n'est pas intégralement payé, et ce, où qu'il puisse se trouver.

8. Prix

Les prix fixés sont en Euro/€, TVA comprise, Recupel, Auvibel et toutes autres taxes comprises.

9. Paiement

Le paiement des commandes en ligne doit être effectué en ligne au moment de la commande par :

- E-Banking
- Visa ou MasterCard

10. Droit de révocation

Le client a le droit de communiquer à Pyramide Ingénierie qu'il renonce à l'achat, sans paiement d'amende et sans indication de motif, endéans les 14 jours calendrier à compter du jour qui suit la livraison de l'article.

Le droit de révocation ne s'applique pas aux personnes morales ni aux produits achetés à des fins professionnelles par une personne physique. Pour exercer le droit de révocation, vous devez nous informer, par une déclaration sans équivoque, de votre décision de révoquer le contrat. Vous pouvez pour ce faire nous contacter par email (shop@cipiyou.be) ou en vous présentant chez cipiyou. A l'égard de l'exercice du droit de révocation, la charge de la preuve incombe au client. Le client est responsable de toute perte de valeur des produits résultant de manipulations dépassant ce qui est nécessaire pour en déterminer la nature, les caractéristiques et le fonctionnement.

Le droit de révocation s'applique uniquement aux produits dans leur emballage d'origine accompagnés de leurs accessoires, modes d'emploi, ainsi que de la facture et de la preuve d'achat. Les produits qui ne seront pas rendus à Pyramide Ingénierie dans cet état resteront à la disposition du client. Le délai de mise à disposition est de 1 mois, délai après lequel Pyramide Ingénierie devient propriétaire du produit.

En cas d'exercice du droit de révocation, le client renverra les produits à Pyramide Ingénierie par tous les moyens et à ses propres risques et frais. Les envois « port payé par le destinataire » et « contre remboursement » seront refusés.

Le droit de révocation ne s'applique pas aux produits suivants :

- Articles installés, endommagés ou incomplets;
- Enregistrements audio et vidéo et logiciels pour ordinateurs dont le scellé a été rompu;
- Articles à caractère clairement personnel;
- Articles ne pouvant être renvoyés, de par leur nature;

Dans les 14 jours suivant la communication à Pyramide Ingénierie de la décision du client de rétraction, tous les montants versés seront remboursés (diminués des frais de transport et de la dépréciation susmentionnées, le cas échéant). Lors du retour d'appareils en magasin, le client ne pourra obtenir aucun remboursement en espèces.

Pyramide Ingénierie se réserve le droit de différer le remboursement jusqu'à récupération du bien et vérification de son état.

11. Garantie

Tous les produits, achetés ou livrés, pour usage privé, bénéficient d'une garantie de 2 ans sauf pour les produits reconditionnés sur lesquels il y a une garantie d'1 an. La garantie offerte par Pyramide Ingénierie correspond à la garantie légale et n'affecte pas les droits légaux de l'acheteur. Si le produit est utilisé à des fins non privées, les conditions limitatives de garantie du fabricant/ fournisseur sont en vigueur.

Nous garantissons que nos produits sont conformes à votre commande et répondent aux attentes normales que vous pouvez avoir tenant compte des spécifications du produit. Nous garantissons également que nos produits répondent à l'ensemble des lois existantes au moment de votre commande.

Des défauts ou des dégâts dus à une mauvaise utilisation, tels que dégâts d'eau, oxydation, chute ou choc, négligence, mauvais entretien, usage en contradiction avec les instructions du fabricant, tout comme l'usure, ne sont pas couverts par la garantie.

Les réparations effectuées par des techniciens non agréés par le fournisseur entraînent la déchéance de la garantie si cette réparation est afférente à ce vice. Pour plus d'informations, veuillez consulter le mode d'emploi livré avec le produit.

La note d'envoi que vous recevrez à la livraison fera office de certificat de garantie. Veuillez donc la conserver soigneusement.

Pour bénéficier de votre garantie, veuillez vous rendre chez cipiyou avec votre appareil défectueux ou contacter notre service après-vente par téléphone (au 087/29.21.20) ou par e-mail (shop@cipiyou.be).

12. Risques Internet

Le client connaît les restrictions et les risques liés à l'utilisation d'internet ou de tout autre moyen par lequel le site Web est actuellement ou sera à l'avenir mis à disposition. Le client connaît aussi les risques de stockage et de transmission d'informations par voie numérique ou électronique. Le client accepte que Pyramide Ingénierie ne peut être tenu responsable pour tout dommage causé par l'utilisation du site Web de Pyramide Ingénierie ou d'internet, suite aux risques précités.

Les parties acceptent les moyens de preuve électroniques (email, backup, etc.).

13. Occasions

Les appareils d'occasion sont uniques.

Sauf mention contraire, les conditions de garantie et de réparation restent d'application pour ces appareils. De par leur nature unique, ils ne sauraient être échangés.

14. Réparations

Lors de la réparation d'appareils électroniques, il est possible que les données qui ont été stockées dans ou par la marchandise à réparer soient endommagées ou perdues. Seul le client peut être tenu responsable de la sauvegarde des données précitées ou de leur reconstitution, à l'exclusion de toute responsabilité de Pyramide Ingénierie à cet égard. Pyramide Ingénierie ne peut en aucun cas être tenu responsable de la perte ou de la reproduction de données stockées dans ou par ces appareils électroniques.

15. Clause salvatrice

La non validité ou l'illégalité d'une des clauses prévues dans les contrats (conditions spécifiques et générales) convenues entre parties, n'entraîne aucunement une invalidité ou une nullité des autres conditions du contrat conclu entre parties - clauses restant intégralement valables.

16. Litiges

Pour toute plainte éventuelle, le client peut s'adresser au Service Consommateur par téléphone (087/29.21.20) ou par email (shop@dpiyou.be).

En cas de contestation entre parties ou de poursuites en paiement, sont seuls compétents les tribunaux de Verviers.

Table of figures

Figure 1 – Diagram of a WebObjects website.....	10
Figure 2 – Content of the cipyoun-Account application.....	11
Figure 3 – View of the WOComponent “Affichage” from the cipyoun-Account application	12
Figure 4 – Content of the folder “Sources” of the cipyoun-Account application.....	13
Figure 5 – Diagram of the database of the application cipyoun-Account.....	16
Figure 6 – Main view of the application cipyoun-Account	18
Figure 7 – State diagram of the application cipyoun-Account.....	20
Figure 8 – jrxml file created in iReport for software accounts.....	24
Figure 9 – Sample PDF file generated by the application cipyoun-Account for a software account	25
Figure 10 – Diagram of the architecture of Pyramide Ingénierie.....	29
Figure 11 – Diagram of the old order process.....	30
Figure 12 – Diagram of the new order processing method used by cipyoun	31
Figure 13 – Screenshot of the page “ContentShoppingCart”	36
Figure 14 – Screenshot of the page “ContentLieuxLivraison”	37
Figure 15 – Screenshot of the page “ContentConfirmToPayment”.....	38
Figure 16 – Diagram of the tables used for discount coupons.....	41
Figure 17 – Diagram of the validation process of a coupon	47
Figure 18 – Screenshot of the coupon editing interface	48
Figure 19 – Modal window used to add a reduction on a family.	49
Figure 20 – Graph representing the change in the total cost per transaction depending on the revenue for a total of 100 transactions.....	56
Figure 21 – Graph representing the change in the total cost per transaction depending on the revenue for a basket of an average amount of € 1,000	57
Figure 22 – Sequence diagram representing the call to the HiPay service web	63
Figure 23 – Example of a SOAP request “generate” provided by HiPay in its documentation.....	64
Figure 24 – Sequence diagram of the payment process once the application has received the redirection URL from HiPay	65

Figure 25 – Example of XML stream sent by HiPay after payment.....	67
Figure 26 – Diagram of the database adapted for the addition of an Address table	69
Figure 27 – Tariff Grid for parcel delivery from Belgium by GLS.....	B

Table of contents

Summary.....	5
Introduction	6
Company Overview.....	6
Tools.....	6
Internship objective	7
Document structure.....	7
PART I: WebObjects & cipyoy-Account	8
Chapter 1: WebObjects	9
1.1 History of WebObjects	9
1.2 Database	9
1.3 Application WebObjects.....	10
1.4 A WebObjects application	11
1.4.1 WOComponents.....	11
1.4.2 Sources.....	13
1.4.3 Resources	14
Chapter 2: cipyoy-Account.....	15
2.1 Model	16
2.2 View.....	18
2.2.1 Authentication	18
2.2.2 Main View.....	18
2.2.3 Warning Deletion	19
2.3 Controller and features	20
2.3.1 Customer management	20
2.3.2 User account management	21
2.3.3 PDF generation.....	21
2.4 Deployment.....	22
2.5 PDF generation.....	23
2.6 Future of the application cipyoy-Account.....	26
Conclusion Part I	27
PART II: Order Processing System.....	28
Introduction.....	29
Architecture of the company system.....	29
Former order process.....	30

New order process.....	31
Chapter 3: Delivery.....	32
3.1 Changing the framework cipyou-web-manage.....	33
3.1.1 Pricing.....	33
3.1.2 Transport product	33
3.1.3 Delivery method selection.....	34
3.1.4 Order status.....	34
3.1.5 Overview table	35
3.2 Views of the delivery selection procedure	36
3.2.1 First view: ContentShoppingCart	36
3.2.2 Second view: ContentLieuxLivraison	37
3.2.3 Third view: ContentConfirmToPayment	38
3.3 Conclusion.....	39
Chapter 4: Discount coupons	40
4.1 Database	41
4.2 cipyou-web.....	45
4.2.1 Insert a code	45
4.2.2 Verify the code.....	45
4.2.3 Execute coupon.....	46
4.2.4 Ensuring limited use of coupon.....	47
4.3 cipyou-manage	48
4.3.1 View of the edition of a coupon page.....	48
4.3.2 How to create and to edit a coupon.....	50
4.3.3 Statistics and summary dashboards	51
Chapter 5: Selecting a payment solution.....	52
Existing solution.....	52
New solution	52
Integrated or rented system.....	52
5.1 Comparison criteria.....	53
5.1.1 Available payment methods.....	53
5.1.2 Prices.....	53
5.1.3 Integration constraints.....	54
5.1.4 Contact	54
5.2 Comparison.....	55

5.3 Choosing HiPay.....	58
5.3.1 HiPay Direct	58
5.3.2 HiPay, a prudent choice	59
Chapter 6: Integration of the solution “HiPay Direct”	60
6.1 Terms and conditions.....	61
6.2 Technical modifications.....	63
6.2.1 Payment Life cycle.....	63
6.2.2 Difficulty: Merchant kit in PHP.....	68
6.2.3 Difficulty: Test ⇔ Production.....	68
Conclusion - Part II.....	69
Conclusion.....	71
Bibliography	72
Attachments.....	A
1: Tariff Grid for parcel delivery from Belgium by GLS.....	B
2: Terms and conditions of Pyramide Ingénierie (in French)	C
Table of figures.....	74