Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor **M^{r}.Ad Mohammed-Sahnoun** for the continuous support of my Master's degree project, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me a lot.

My sincere thanks also goes to **M^{r}. Dahmas**, **M^{r}. Benzekri**, and **M^{r}. Khaldoune**, for offering me the opportunity to defend this work.

I thank my fellow lab mates in SONELGAZ Agency Group: **Ramzi Hmida** and **Ferhat Youcef** for the stimulating discussions and the help they gave me. Also I thank my friends in INELEC.

Last but not the least, I would like to thank my family: my parents, for giving birth to me at the first place and supporting me spiritually throughout my life.

Abstract

The SONELGAZ division management of electricity distribution is facing many problems including:

- Lack of a web application for calculating and managing losses.
- Absence of a dynamic update of the database.
- Obligation of grasped static ways of calculating formulas.
- Access difficulty to data by the distribution service.

Our main work is to Design and implement a web application that manages these electricity losses through the following operations:

- The study and design of an application that automatically processes and analyzes the received XML files.
- The study and design of the database, which should contain all the data, needed for this web application.
- Calculating the consumption and waste of each agency.
- The design of a fluid and attractive model in an interactive interface that facilitates the update without going through the source code.

Contents

List of Figures

List of Tables

Introduction:

Sonelgaz commercial agencies today need to be profitable and generate substantial profit margins, thus they have to control losses and fraud, to do this, they have need automated information systems.

To calculate electrical losses, SONELGAZ retrieves data (counts) on consumption of the sale offices, data that is statically calculated by using Excel files. As part of our final year project study, we developed a web application that makes an evolution of technology over rates consumption that have moved from a manual technique which required the agent to move and capture data on paper, to a technique that makes remote reading (through a GSM chip), hence, the agency has seen obliged to develop a web application that manipulates XML files transmitted by the GSM chip to be able to do the required calculations in a dynamic way and avoid getting them done by Excel.

To achieve this We have collected the factures written in excel to implement our database using my sql platform, then implemented the application using PHP programming language, some HTML and CSS commands and javascript.

Chapter 1 :

Hardware System Description

Introduction:

To ensure the effective work of our new information system, we were obliged to use a special hardware that facilitate the readibility and data collection.

1. Hardware System presentation:

Our system compromises three essential parts:

- ACE6000 Electricity meter.
- FASTRACK modem M1206.
- A computer.

1.1 The electricity meter Itron ACE6000 Presentation: [1]

Technical specification

Ratings	Voltage: 3*57.5 /100V up to 3*277/480V auto ranging Direct Curent : Lb 5A, Imax 100A
Networks Types	Direct Connection: 4-wire meter, operational on 3- Connection. Wire
Frequency	50 / 60 Hz
Real Time Clock	Back up with external removable battery and internal super capacitor.
Communication	IR-port (IEC62056/21 and IEC 62056/42-46-53-61-62) RS232 or RS485 DLMS Cosem, Protocol (IEC 62056 / 42-46-53- 61-62)

Adding Values: [1]

Through the application of latest generation of metrological and communication technologies, ACE 6000 brings significant benefits to utilities and end-users alike.

- Utility Benefits:
  - Reduced inventory cost thanks to a wide measuring range and auto ranging power supply, industrial and commercial users need only one type of meter for many types of installations.
  - Reduced data collection cost read cycles are kept to a minimum by internal storage of all billing data, and powerful communications capabilities permit low- cost remote meter reading.
  - Integrated easily into standard data collection systems as well as future GPRS or IP based systems.
  - Reduced Non-Technical losses multiple safety features guard against technical problems introduced by human intervention.

- End-User Benefits:

- Excess Consumption Features ACE6000 can monitor consumption against configurable threshold and can activate contacts if consumption exceeds limits.
- Instantaneous values such as Power Factor, demand, Volts and Amps are made available for the end-user to help monitor personal consumption.

1.2 FASTRACK modem M1206 Description:

1.2.1 General presentation: [2]

FASTRACK M1206 modem is a self-contained E-GSM / GSM-GPRS 900/1800 dual-band modem and GPRS class 10 capable.

This modem supports the following transmission:

- Data,
- Fax,
- Short messages (Point to point and Cell Broadcast),
- Voice calls.

The modem comprises several interfaces:

- LED function indicating the operating status,
- External antenna (via SMA connector),
- RS232 Serial and control link (via 15-pin SUB HD connector),
- Power supply (via 4-pin Micro-Fit connector),
- SIM cardholder.

The main features of the modem are the following:

- 2 Watts E-GSM 900 radio section.
- 1 Watt GSM 1800 radio section.
- 32 Mbits of Flash memory and 4 Mbits of SRAM.
- Real Time Clock with calendar.
- Echo cancellation + noise reduction.
- Full GSM or GSM / GPRS software stack.
- Hardware GPRS class 10 capable.
- Complete shielding.
- A DC power supply,
- A RS232 serial link,
- Audio interface for:
    - microphone,
    - speaker.
- A 3V / 5V SIM interface.

1.2.2 External Connections: [2]

FASTRACK M1206 modem has three external connections:

- Antenna connector: SMA connector to RF connection to the antenna,

- Sub High Density 15-pin connector for:
  - RS232 serial link connection,
  - Audio lines (microphone and speaker connection),
  - BOOT and RESET signals connection.
- Power supply connector: 4-pin Micro Fit connector for DC power supply [2].

- **Antenna Connector:**

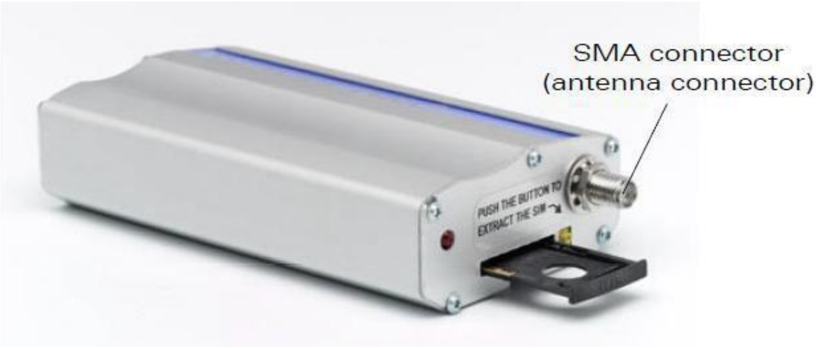


Figure2.1: Antenna connector [2]

- **Sub HD 15-pin connector:**

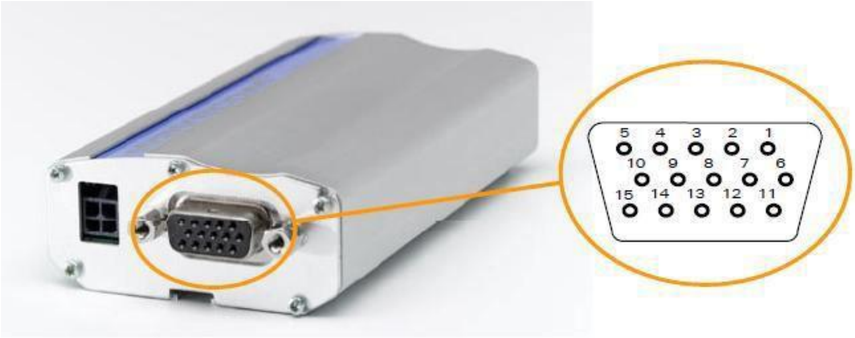


Figure 2.2: Sub HD 15-pin connector [2]

- **Power supply connector**

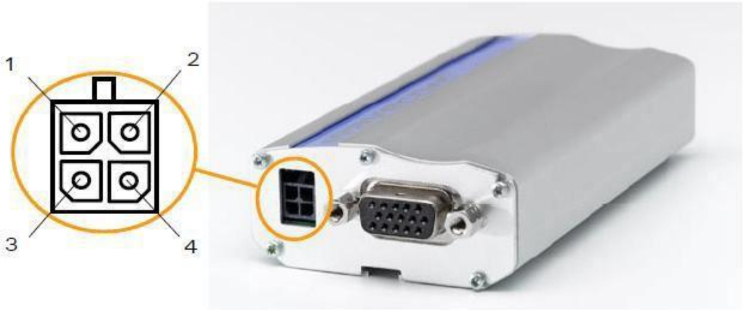


Figure 2.3: Power supply connector [2]

2. Setting up the System:

Setting up our system means set up the modem, doing the following:

- Insert the SIM card in the holder.
- Connect the antenna to the SMA connector.
- Connect both sides of the serial and control cable.
- Connect the power supply cable to the power supply source.
- Now the modem is ready to work.
- Connect the RS232 link between the DTE (port COM) and the modem (DCE).
- Configure the RS232 as follows:
    - Bits per second: **115.200 kbps,**
    - Data bits: **8 bits**,
    - Parity: **None,**
    - Stop bits: **1,**
    - Flow control: Hardware Flow control.
- Using a communication software like HyperTerminal program, enter the **AT** command.
- After getting started the modem the following commands are used:
    - **AT+CGMI**: modem answer is ”WAVECOM MODEM” when serial link is ok.
    - **AT+CPIN= <pin code>:** to enter a code (if activated).
    - **AT+CREG**: to verify the registration of the modem on the network.
    - **ATD <phone number of the Electricity meter targeted>:** to initiate a call.
    - **ATH:** to hang up (end call).

The data received is in XML files form, and then it is stored and treated to build our Database which is shown in the next chapter.

Chapter 2:

Project Design Using UML

1. Introduction:

This study is to move from the existing system to the future one, which must be consistent, reliable and objective, based on the diagnosis of the existing study.

Therefore, the study will allow the deployment of a set of data models and treatments to be used in the definition and implementation of a new system capable to fulfill the user expectations.

We pass to the modeling of a system via two aspects namely: the static aspect that represents the data of the system and the dynamic aspect that represents the treatments.

2. Presentation of the model:

UML (Unified Modeling Language) is a modeling graphic language originally designed to represent, specify, design and document software systems artifacts. Adopted by the Object Management Group (OMG) as a standard, it has become a reference in the field of software engineering. Its richness and power of expression also make it eligible for modeling concepts and processes "craft" ("business modeling") for engineering and non-software systems. [3]

UML is a result of the unification of proven technologies for the analysis and design of large and complex software systems.

UML is based on three types of bricks to allow the modeling of an architecture:

- Items which are the smallest vocabulary of UML;
- Relationships, which represent the project syntax;
- The diagrams, which contain the semantics of the project.

3. Analysis and specifications:

3.1. The Use Case Diagrams:

- **Presentation:**

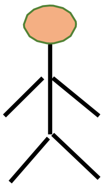
The use case diagram is the first diagram in the UML, it allows visualizing the behavior of a system so that:

- The user can understand how to use each item.

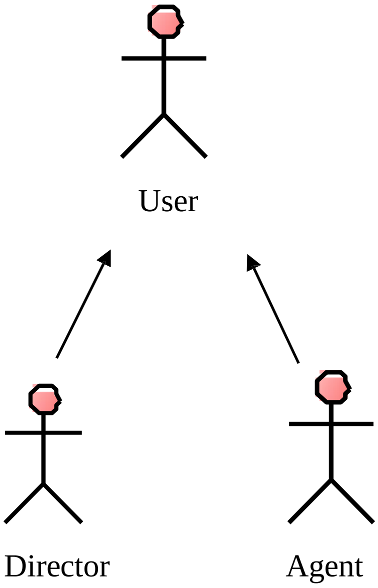
- The developer can implement these elements.

The elements of the use case diagram are:

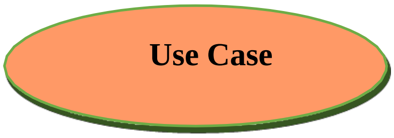
- Actor: is the role of application users. It is presented by:



- Generic actor: The generic actor is an actor who inherits one or more players who are charged with the same task.



- Use case: A use case describes what does an application, but does not specify how the application does. It describes a sequence of actions performed by the application.



- Relationship: describes the behavior of these classes. Three types of relationships exist:

  Association: Structural relationship between classes.

  Dependence: Use relationship which states that instances of a class are connected to instances of another element.

  Generalization: The relationship between a general class (parent) and a specific class (child) derived from it. [3]

———	**Communication path**
- - - - ▶	**Include**
◀———	**Extend**

N°	Acteur	Rôle
1	The Director	- Consult the final results (loss ratio)
2	Agent	-Consult Results - Edit Information - Enter the information in the event of failure automatic processing XML file
3	Computer Software	- Injects the XML files in the application

3.1.1. General Use Case Diagram:

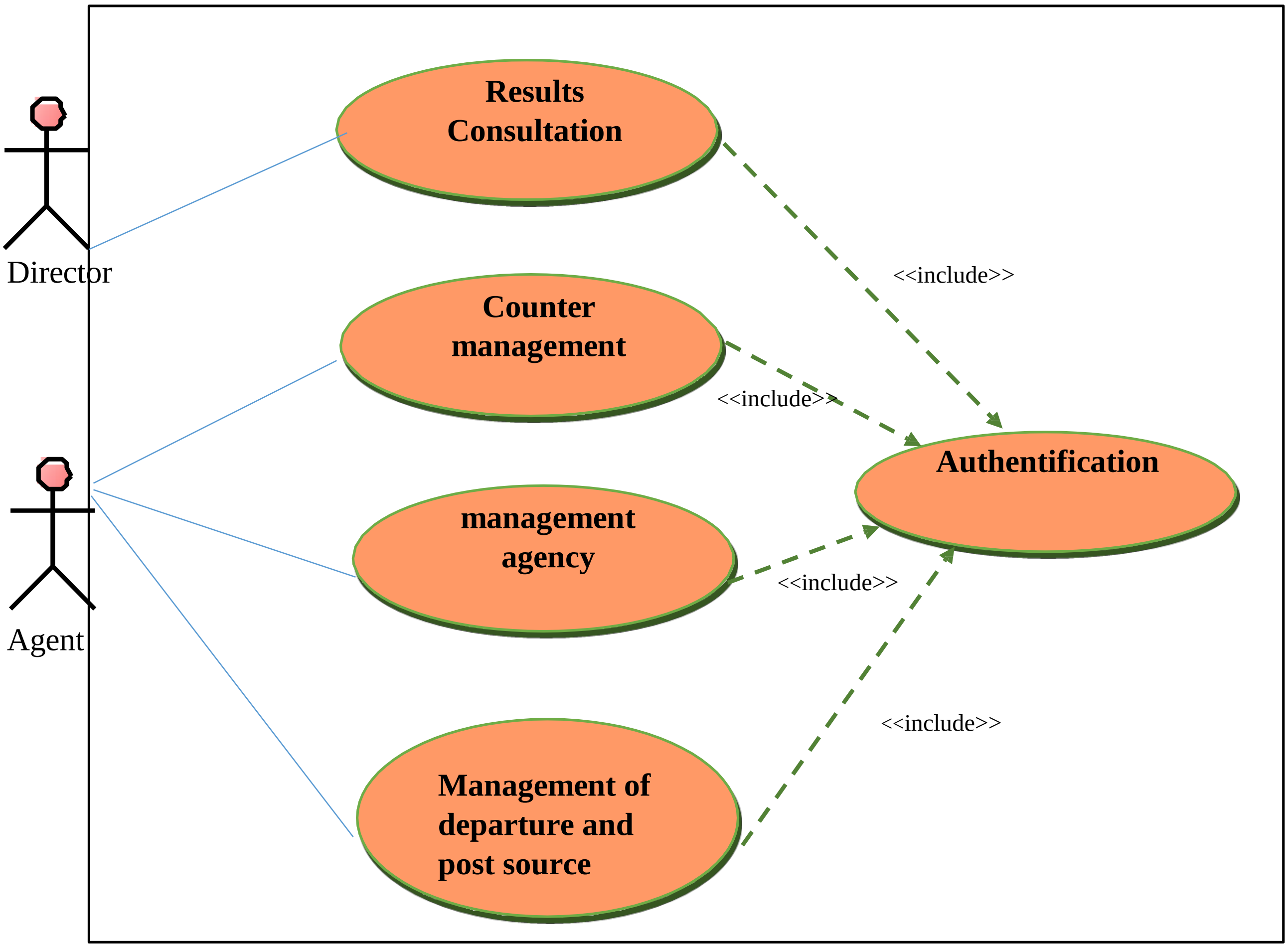


Figure 2.1: General use case diagram

3.1.2. Use Case Diagram <<view result>>:

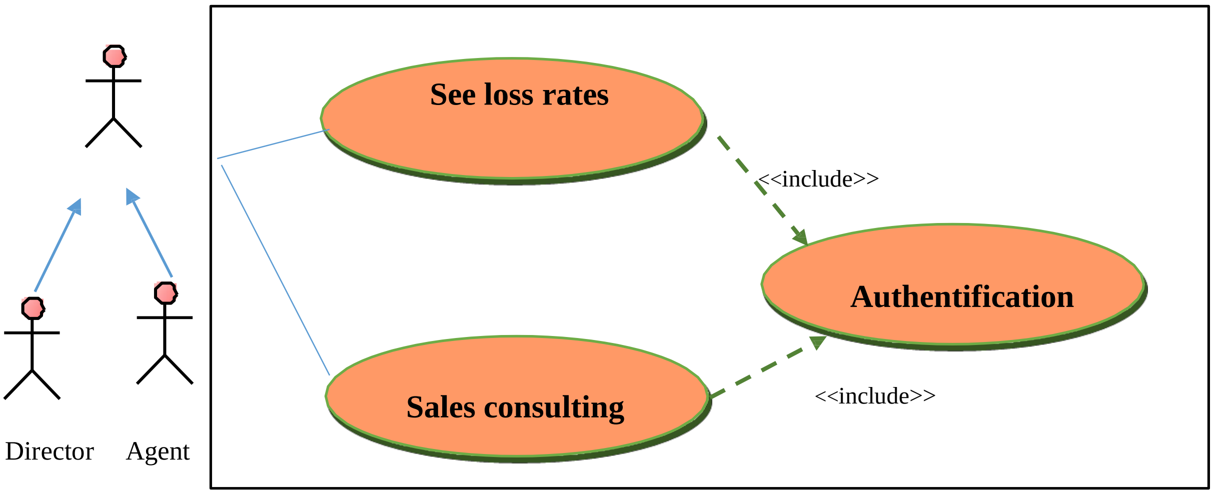


Figure2.2: Use Case Diagram <<View Result >>

3.1.3. Use case diagram <<Counter Management >>:

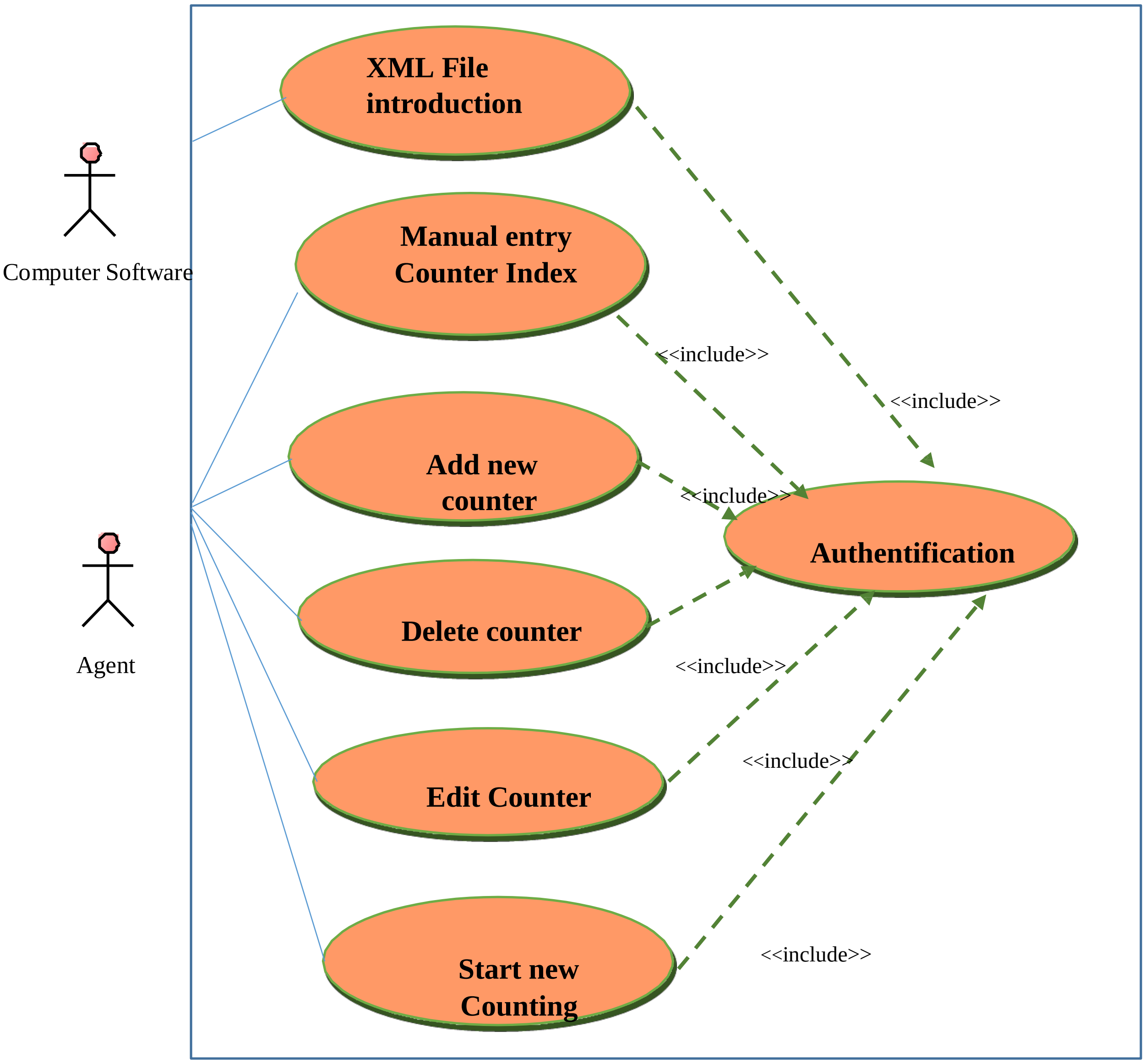


Figure 2.3: Use Case Diagram <<Counter Management>>

3.1.4. Use Case diagram <<Agency management>>:

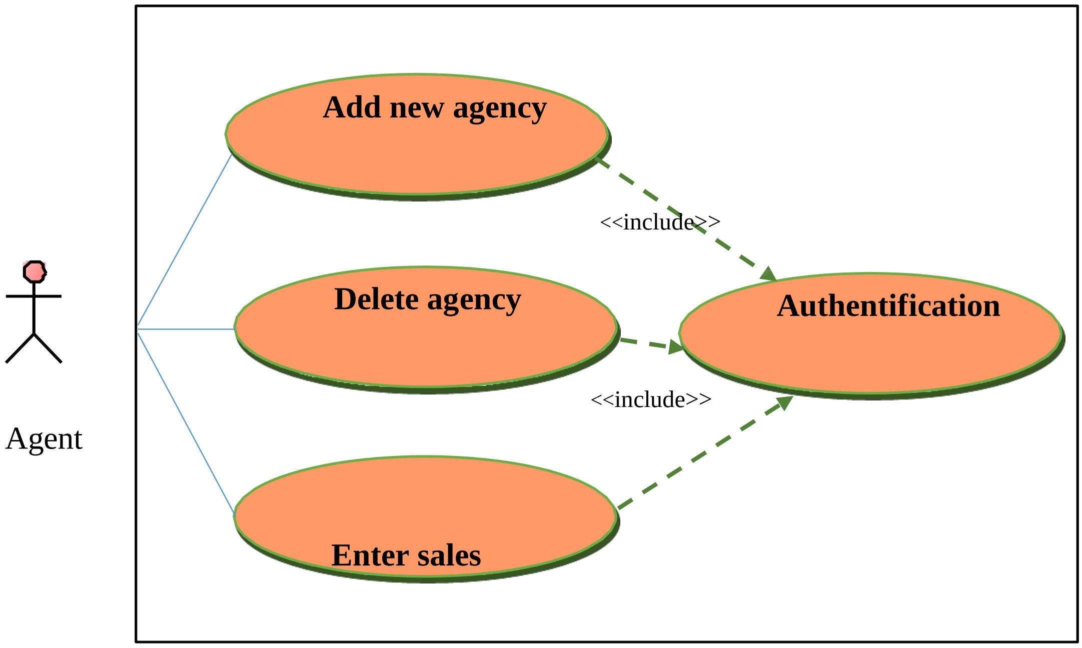


Figure 2.4: Use Case diagram <<Agency Management>>

3.1.5. Use Case Diagram <<management of departure and post source>>:

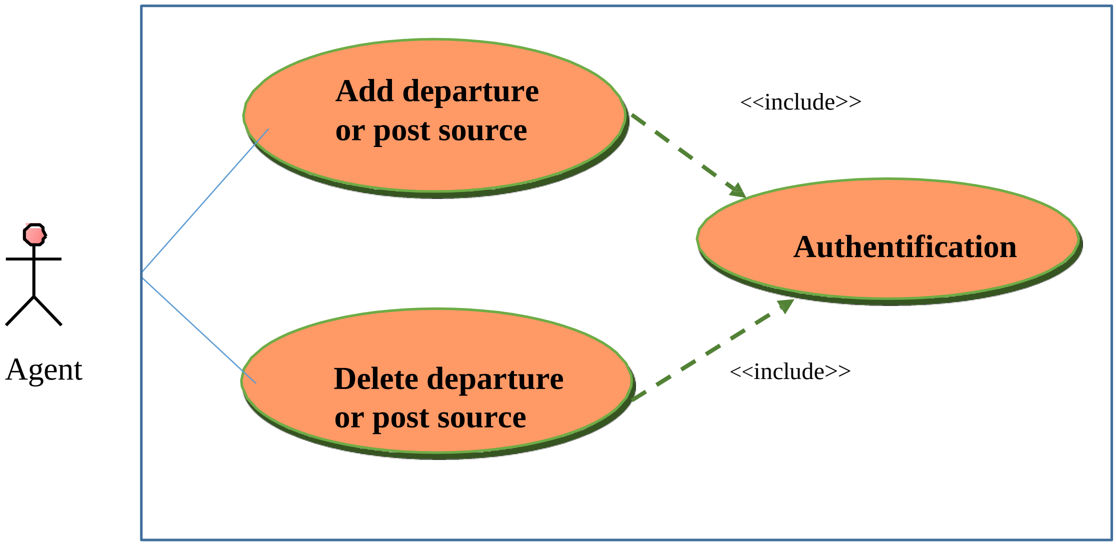


Figure 2.5. Use Case Diagram <<Management of Departure and Post source>>

3.2. Sequence Diagram:

- **Presentation:**

A sequence diagram represents the chronological order of messages sent and received by a set of objects.

- **Sequence diagram elements**

nnA sequence diagram consists of the following elements:

• **Object**: represents the different objects used. Each object is represented by a square above a dotted line. This line represents the lifetime of the object.

• **An object activation period**: On the lifeline of an object, it is possible to insert activation periods of the object.

• Message is through horizontal arrows, the messages exchanged between the different objects.

• **Package**: divides and organizes the representation of the diagram.

3.2.1. Authentication sequence diagram:

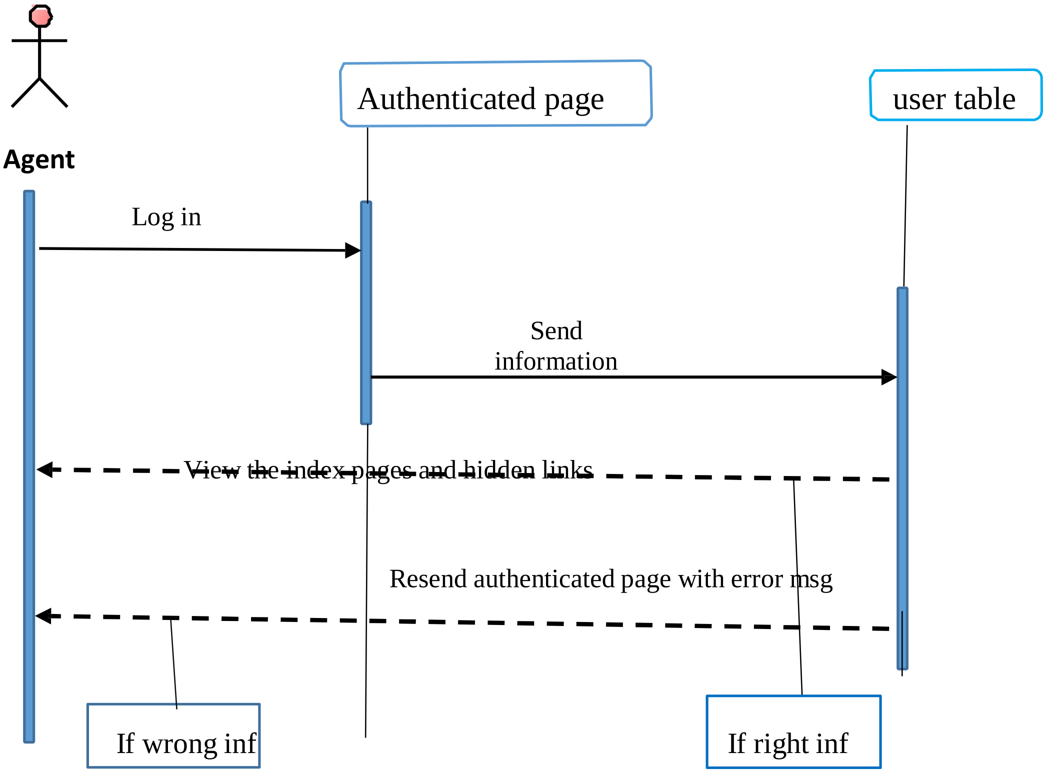


Figure 2.6: Authentication Sequence Diagram

3.2.2. Sequence Diagram <<add new counter>>:

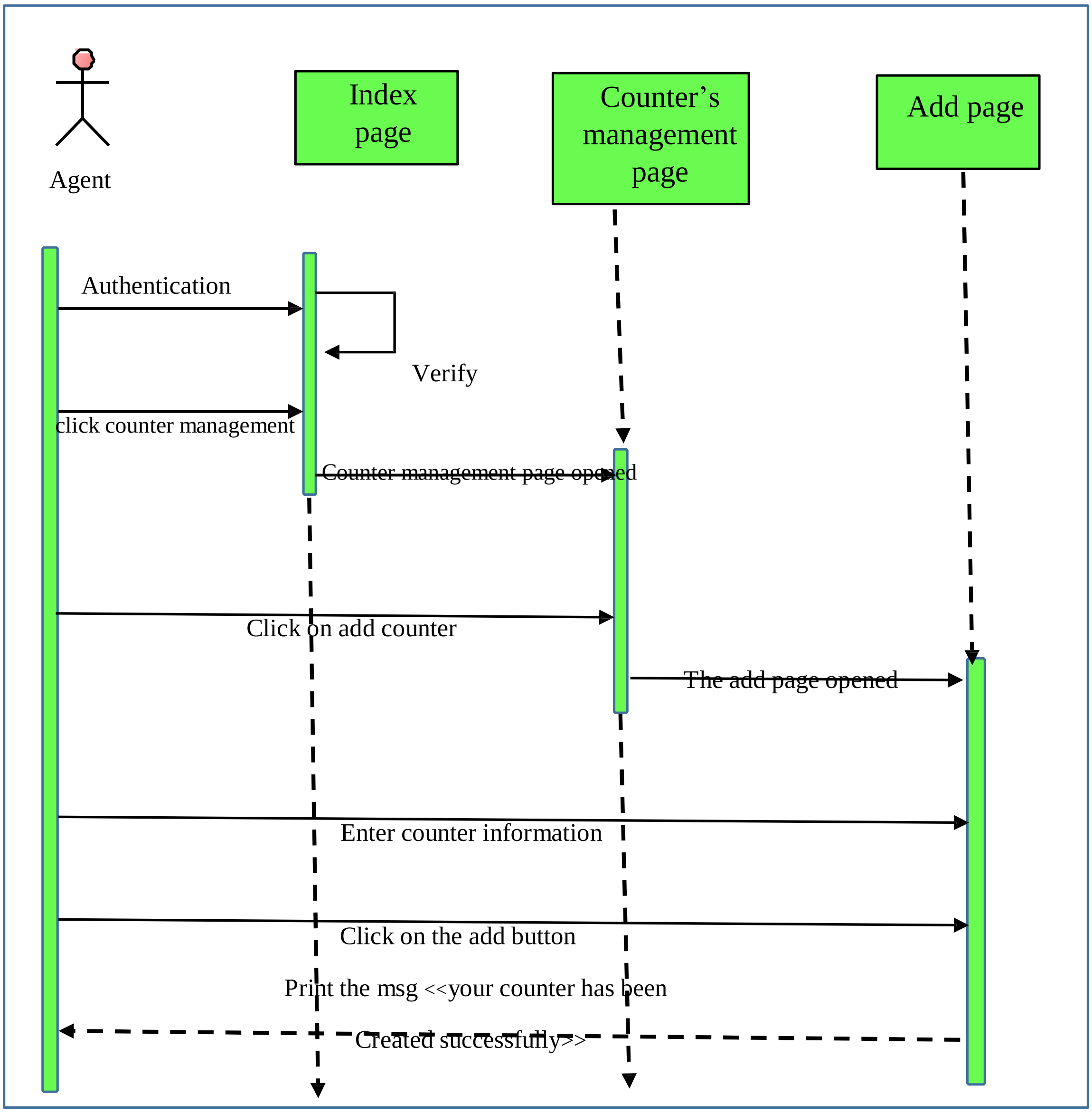


Figure 2.7: Sequence Diagram <<Add new Counter>>

3.2.3. Sequence diagram <<Delete Counter>>:

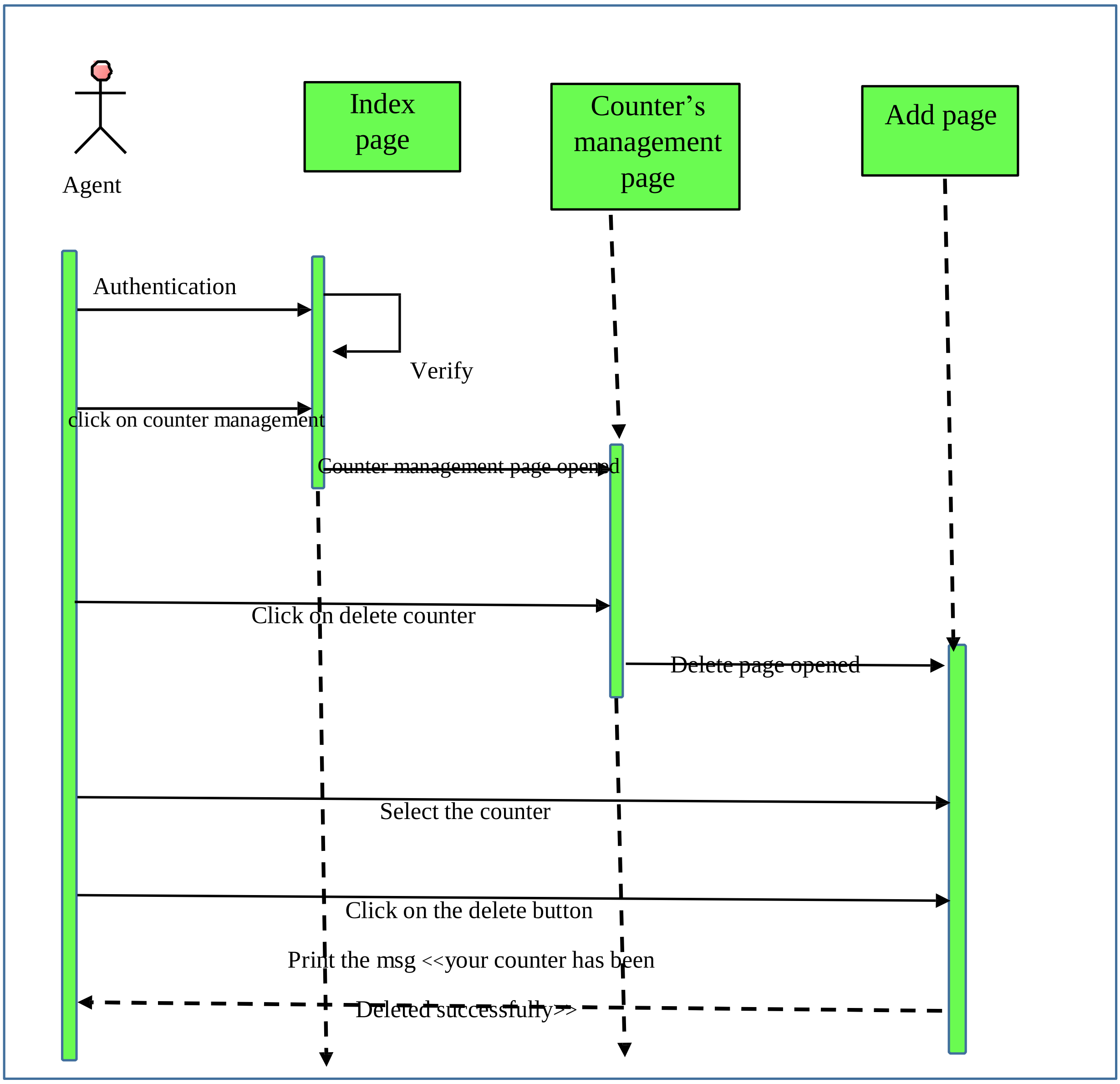

Agent
Index page
Counter's management page
Add page
Authentication
Verify
click on counter management
Counter management page opened
Click on delete counter
Delete page opened
Select the counter
Click on the delete button
Print the msg <<your counter has been
Deleted successfully>>

Figure 2.8: Sequence Diagram <<Delete Counter>>

3.2.4. Sequence diagram <<Edit Counter>>:

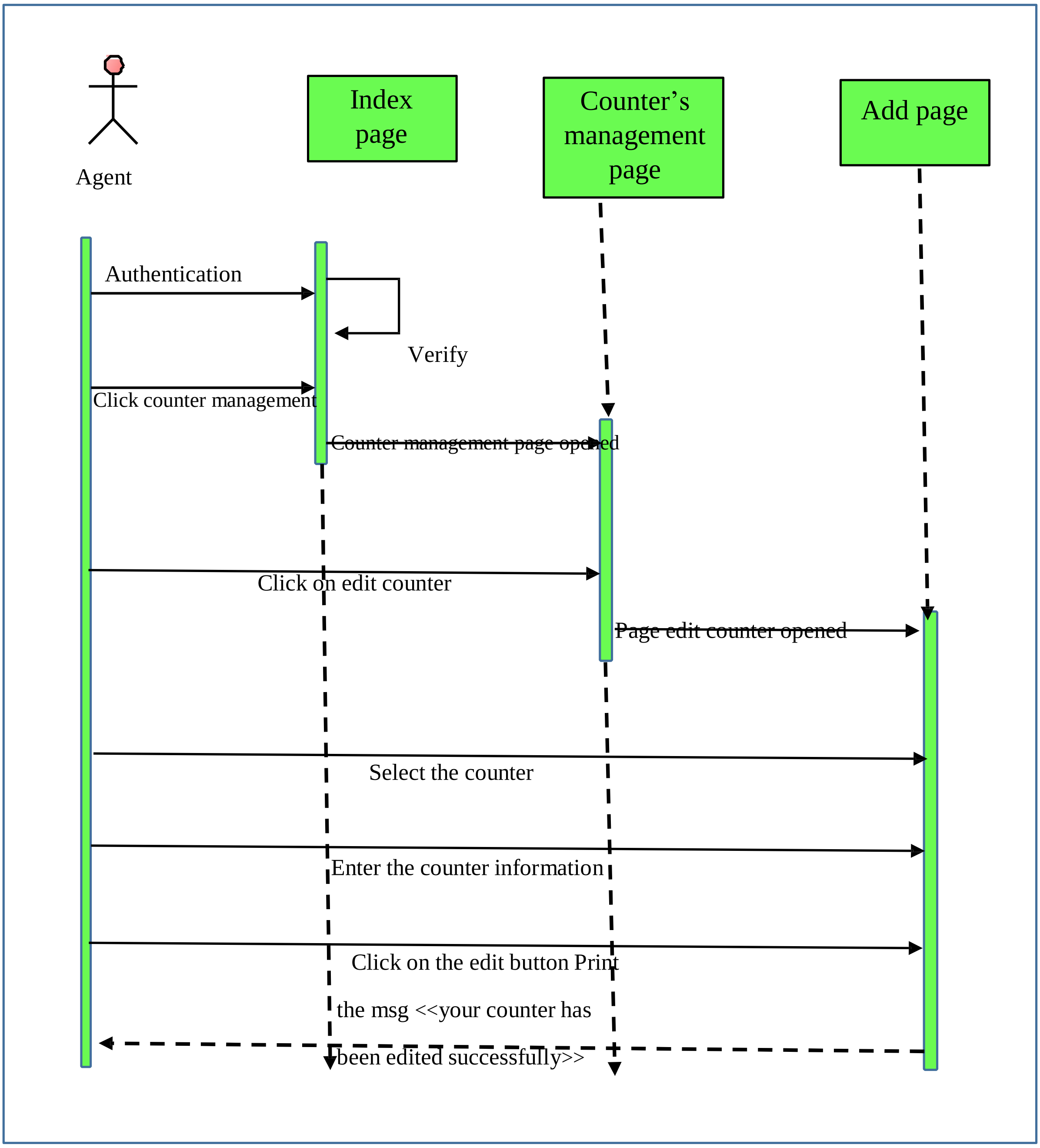

Agent
Index page
Counter's management page
Add page
Authentication
Verify
Click counter management
~~Counter management page opened~~
Click on edit counter
~~Page edit counter opened~~
Select the counter
Enter the counter information
Click on the edit button Print
the msg <<your counter has
been edited successfully>>

Figure 2.9: Sequence Diagram <<Edit Counter>>

3.2.5. Sequence diagram <<manual entry of the counter index>>:

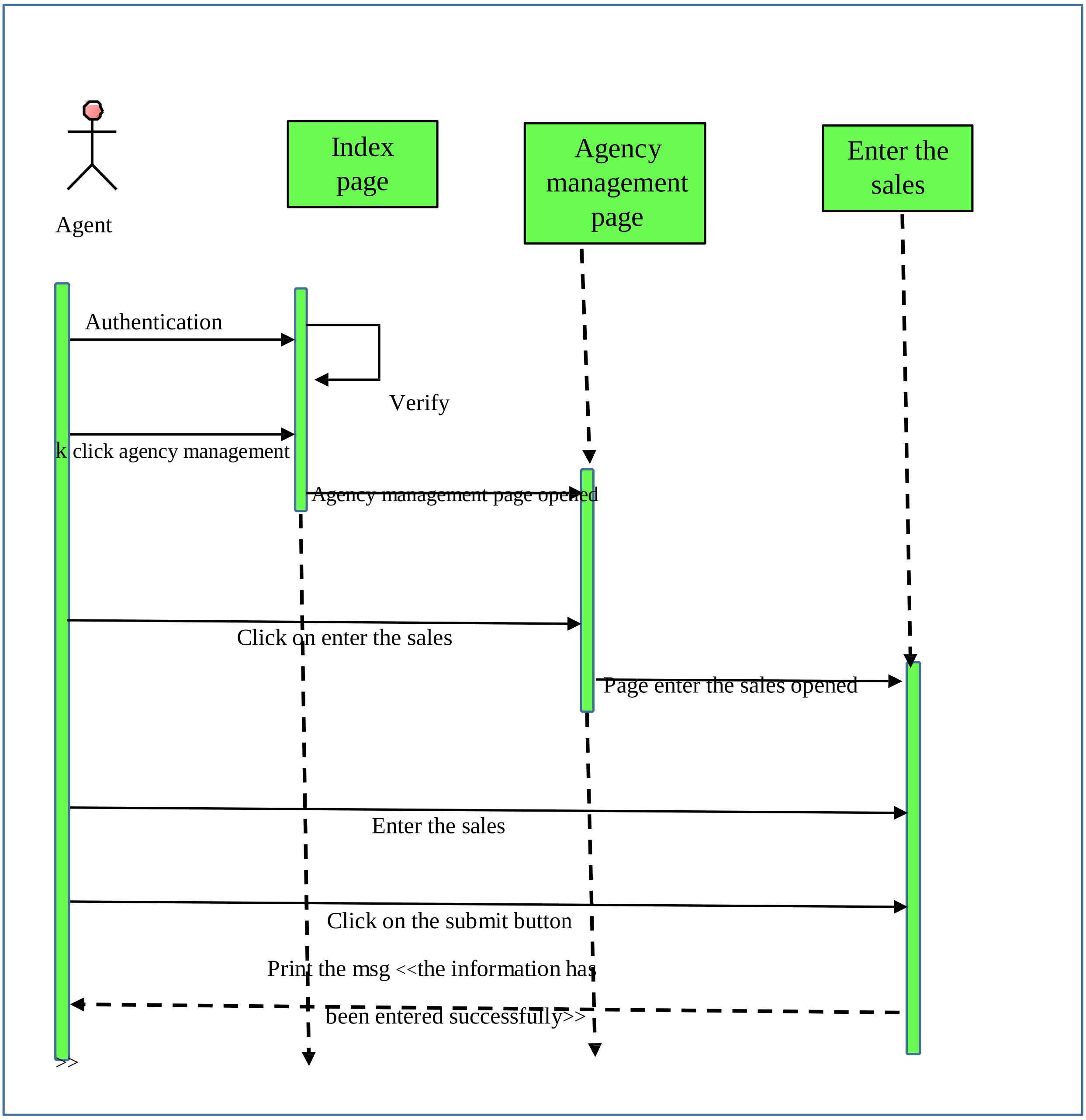

Agent
Index page
Agency management page
Enter the sales
Authentication
Verify
k click agency management
Agency management page opened
Click on enter the sales
Page enter the sales opened
Enter the sales
Click on the submit button
Print the msg <<the information has
been entered successfully>>
>>

Figure 2.10: Sequence diagram <<manual entry of the counter index>>

3.2.6. Sequence diagram <<start new statement>>:

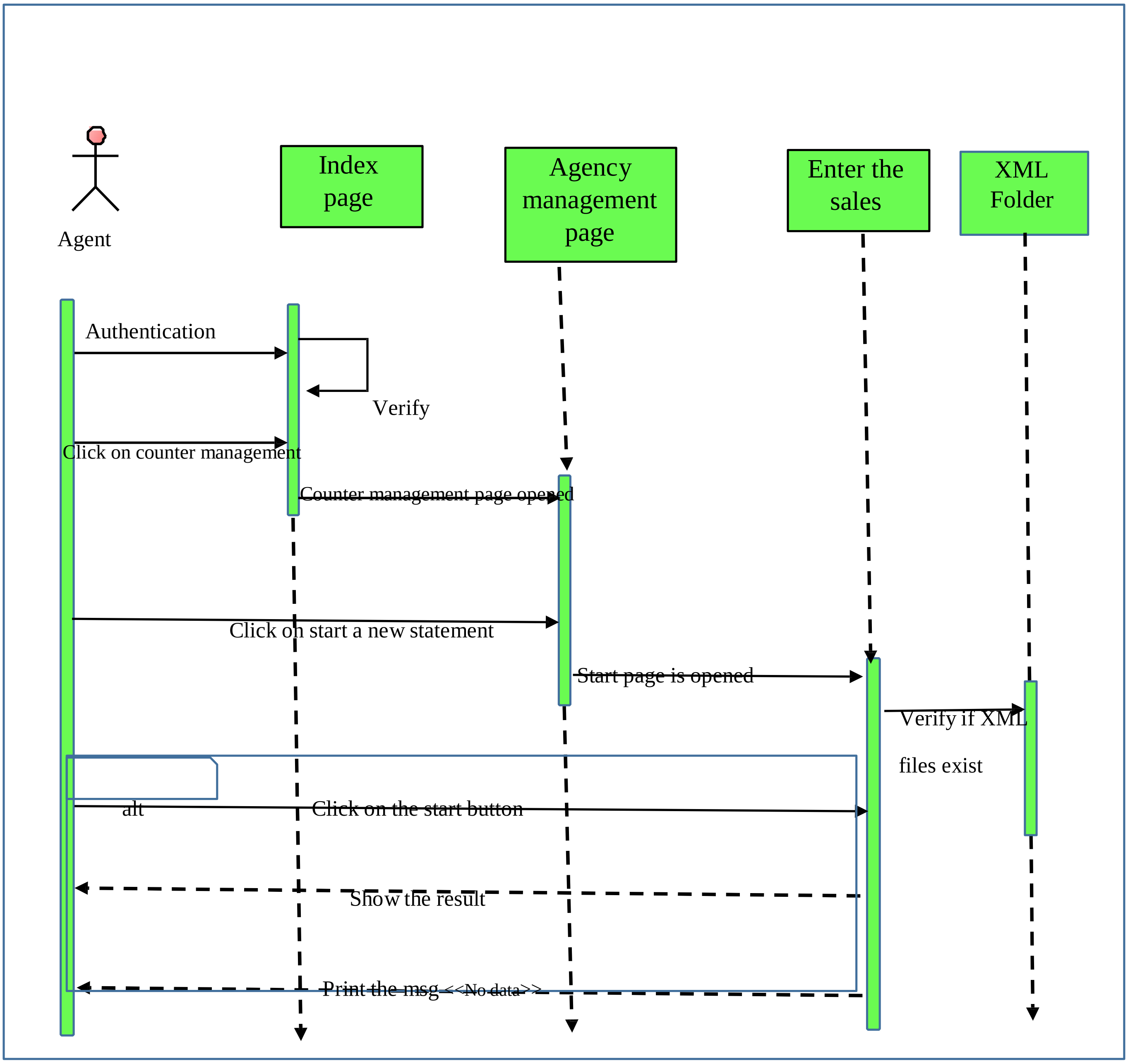

Agent
Index page
Agency management page
Enter the sales
XML Folder
Authentication
Verify
Click on counter management
Counter management page opened
Click on start a new statement
Start page is opened
Verify if XML files exist
alt
Click on the start button
Show the result
Print the msg <<No data>>

Figure 2.11: Sequence diagram <<start a new statement>>

3.2.7. Sequence diagram <<add agency>>:

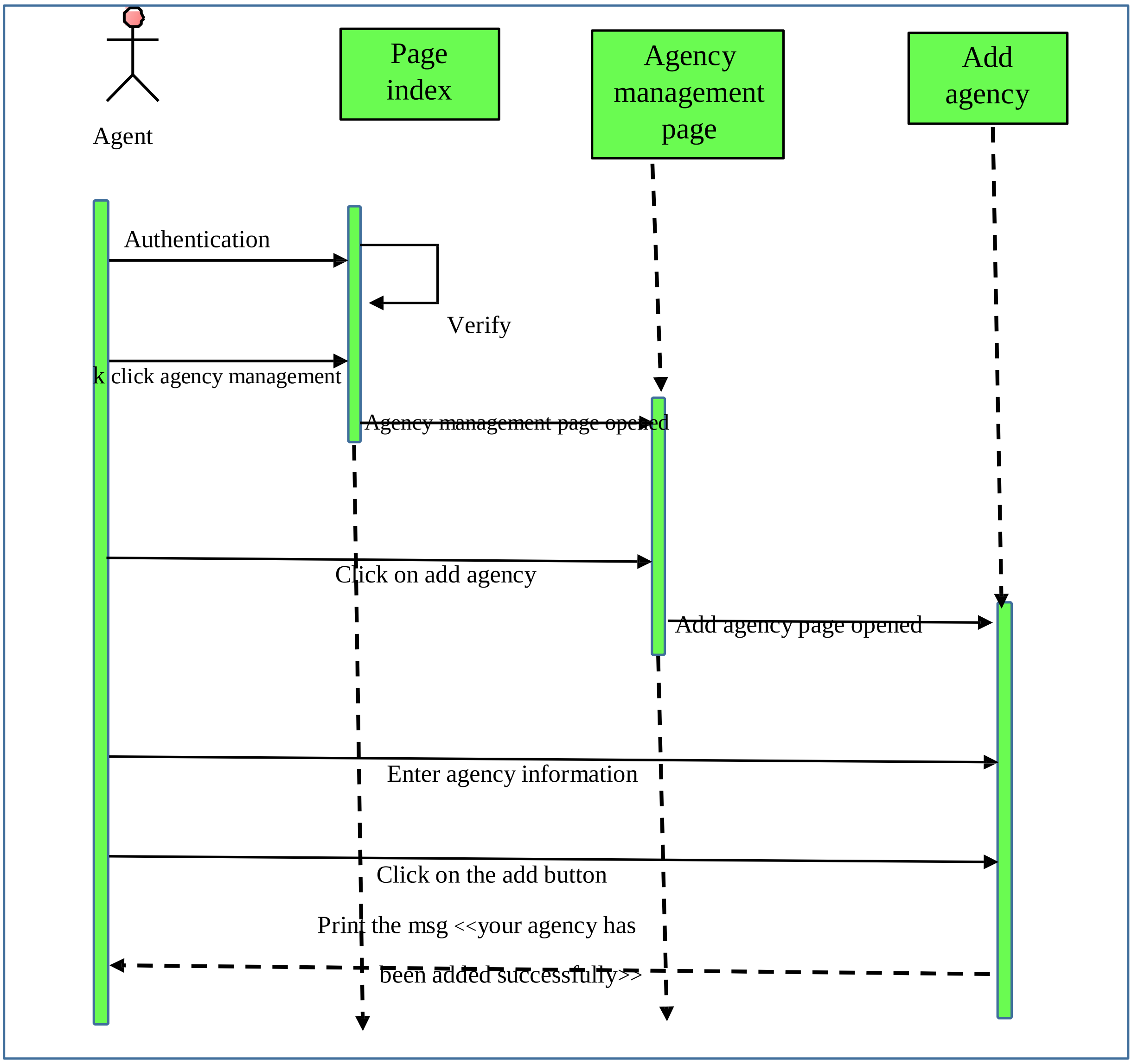


Figure 2.12: Sequence diagram <<add agency>>

3.2.8 Sequence diagram <<Delete agency>>:

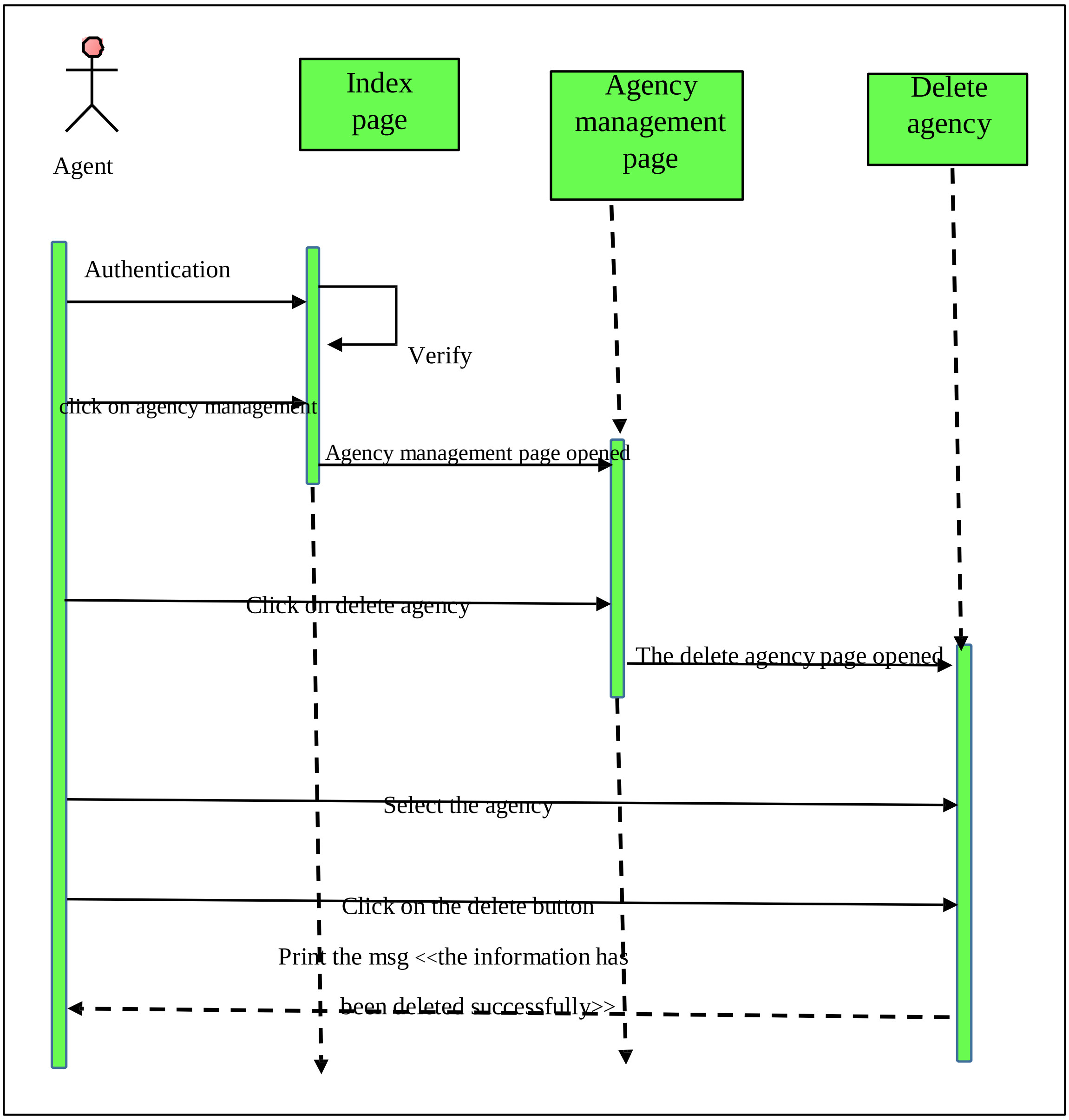

Agent
Index page
Agency management page
Delete agency
Authentication
Verify
click on agency management
Agency management page opened
Click on delete agency
The delete agency page opened
Select the agency
Click on the delete button
Print the msg <<the information has
been deleted successfully>>

Figure 2.13: Sequence Diagram <<Delete Agency>>

3.2.9. Sequence Diagram <<Enter the Agency Sales>>:

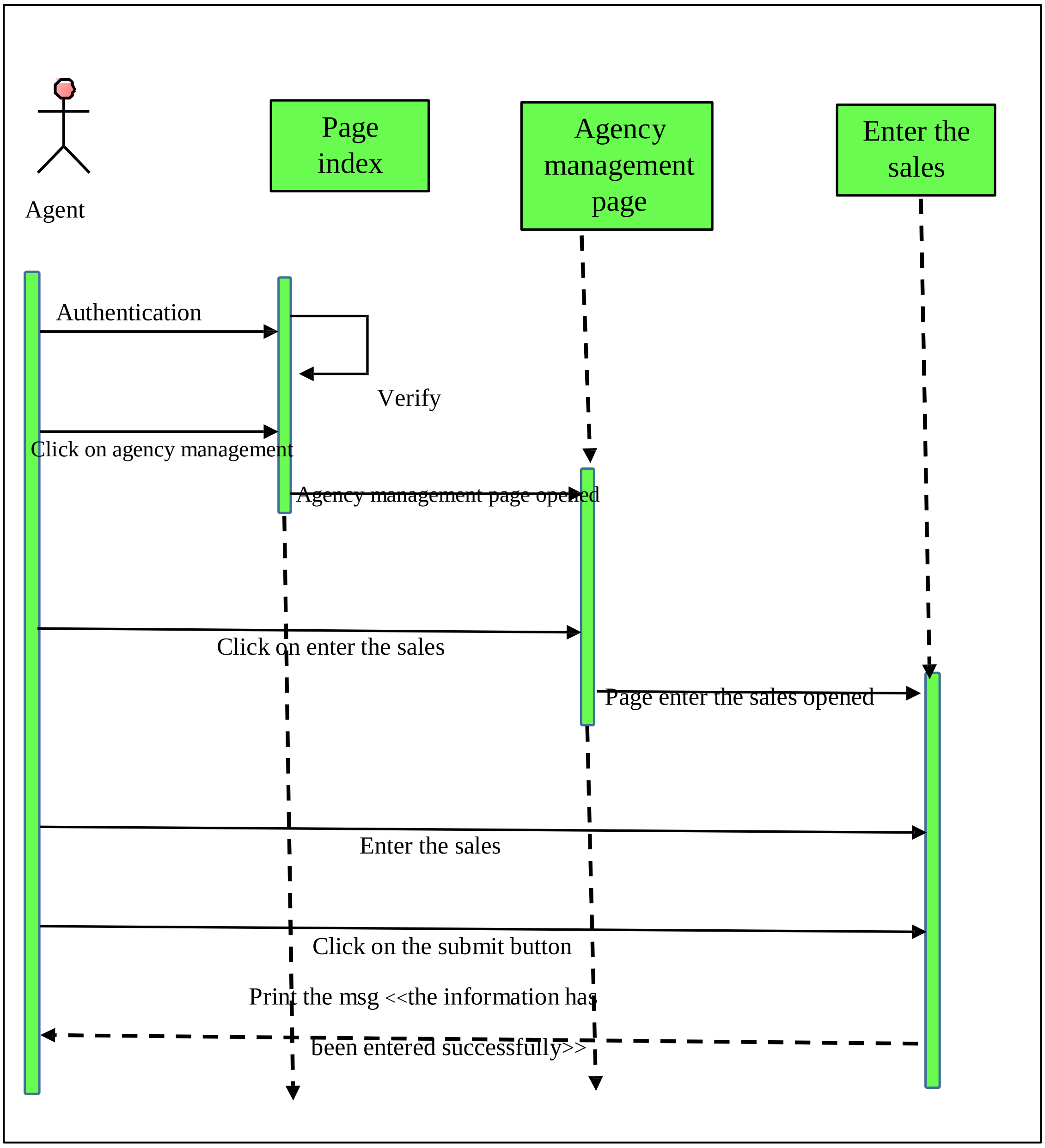

Agent
Page index
Agency management page
Enter the sales
Authentication
Verify
Click on agency management
Agency management page opened
Click on enter the sales
Page enter the sales opened
Enter the sales
Click on the submit button
Print the msg <<the information has
been entered successfully>>

Figure 2.14: Sequence Diagram <<Enter the Agency sales>>

3.2.10. Sequence diagram <<Add departure or post source>>:

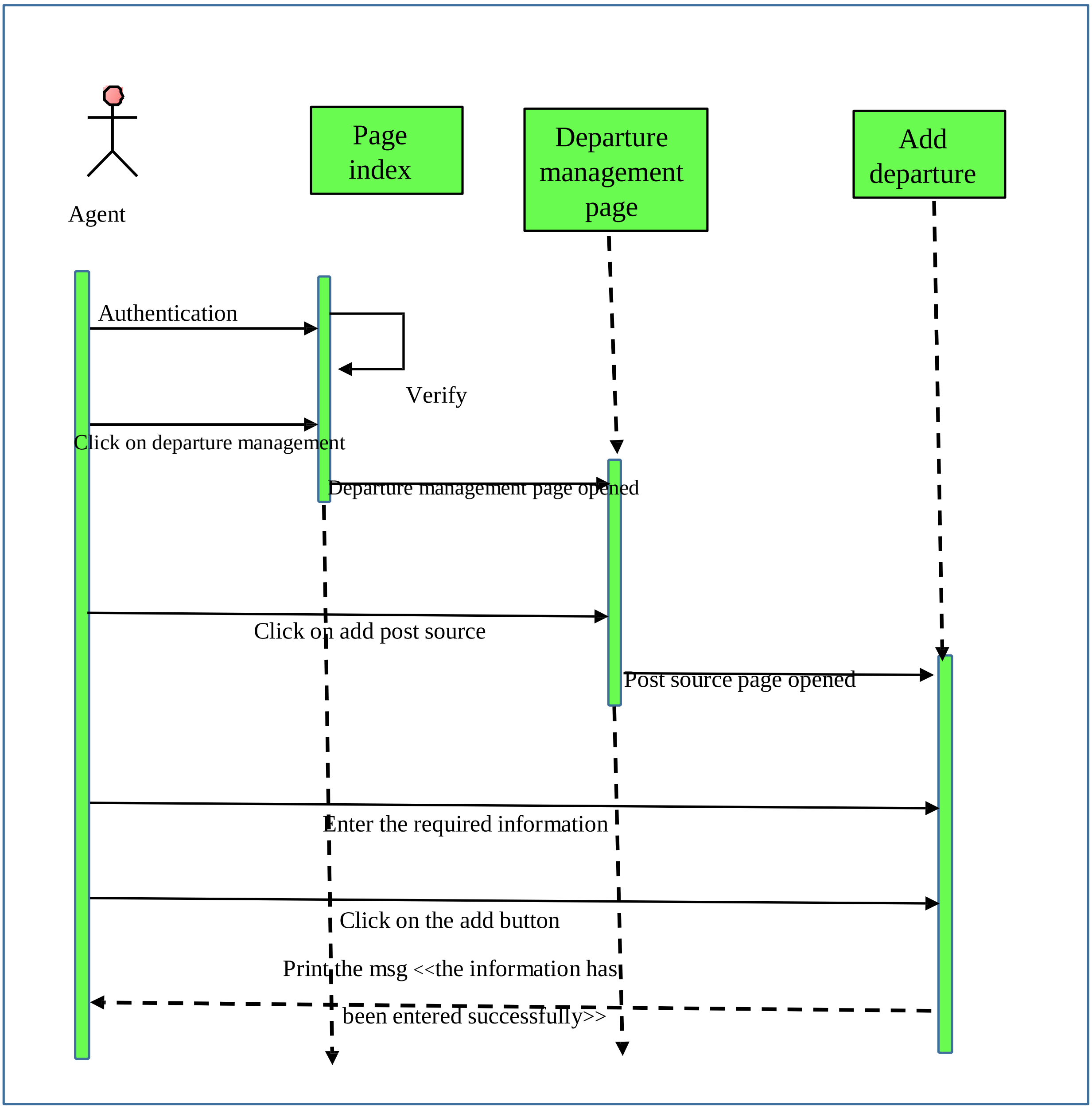

Agent
Page index
Departure management page
Add departure
Authentication
Verify
Click on departure management
Departure management page opened
Click on add post source
Post source page opened
Enter the required information
Click on the add button
Print the msg <<the information has been entered successfully>>

Figure 2.15: Sequence diagram << Add post source>>

3.2.11. Sequence diagram <<Delete post source>>:

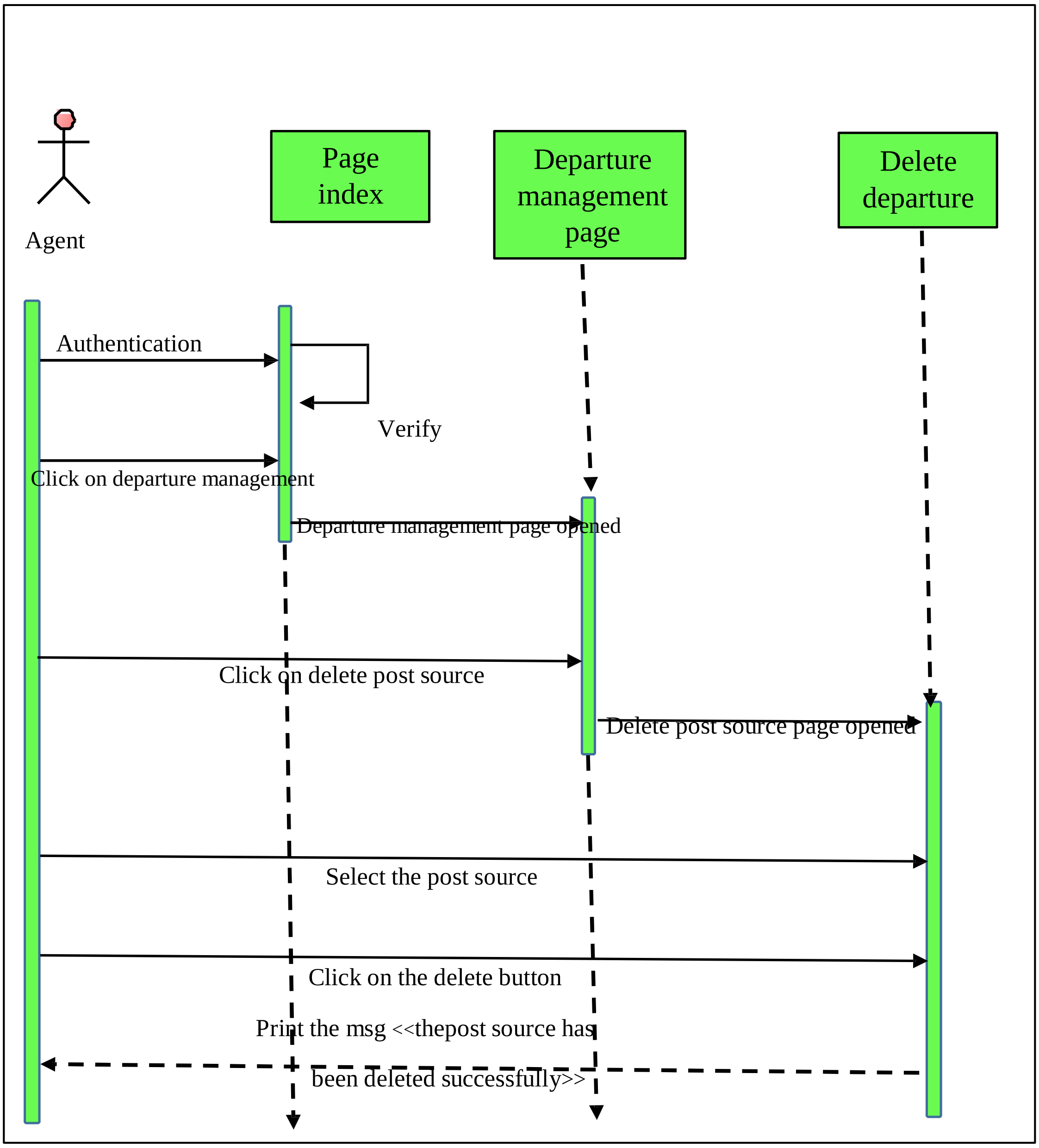
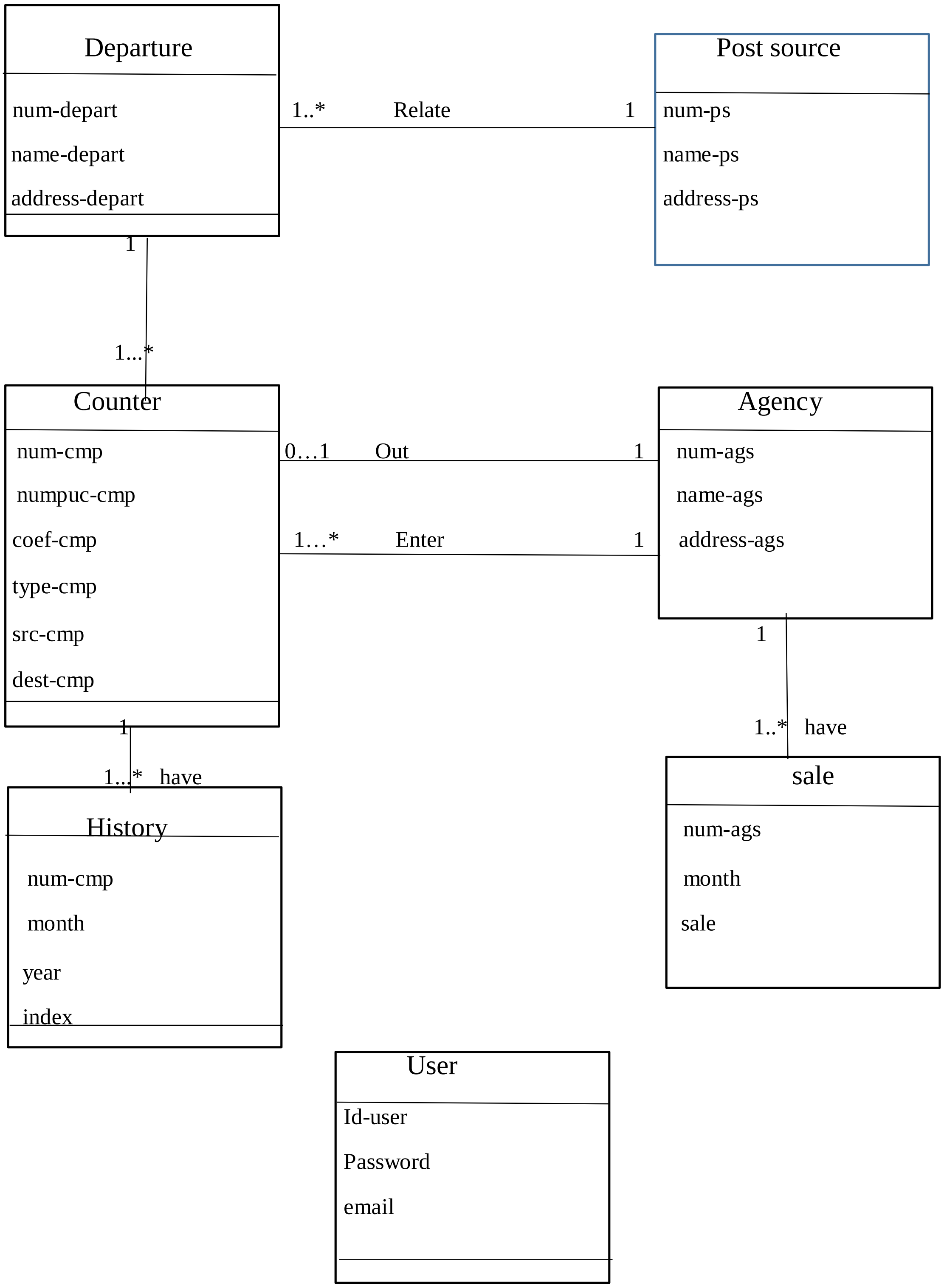

Agent
Page index
Departure management page
Delete departure
Authentication
Verify
Click on departure management
Departure management page opened
Click on delete post source
Delete post source page opened
Select the post source
Click on the delete button
Print the msg <<thepost source has been deleted successfully>>

Figure 2.16: Sequence Diagram <<Delete post source>>

3.3. Class diagram:

It represents the classes involved in the system. The class diagram is a static representation of the elements that make the system and their relationships.

The realization of this diagram requires a data dictionary, and the rules required to manage the connection between classes.



Our class diagram is based on:

- A post source, which has several departures.
- A departure belongs to a single source position.
- A counter has a single source and a single destination.
- The source attribute of the counter can be either a starting point or an agency.
- The attribute of the destination table is always a counter agency.

3.3.1 Dictionary Data:

Class name	Identifier	Attributs
Post source	num-ps	num-ps name-ps address-ps
Depart	num-depart	num-depart name-depart address-depart
Agency	num-ags	num-ags name-ags address-ags
Compteur	num-cmp	num-cmp numpuc-cmp coef-cmp type-cmp src-cmp dest-cmp
Sale	num-ags month	num-ags month

		Sale
History	num-cmp month year	num-cmp month year index
User	**I**d-user	**I**d-user password email

Table 2.1: Dictionary of Data

3.3.2. Dictionary links:

Relation	Collection	Identifier	Multiplicity
	Depart Post source	Num-depart Num-ps	1..* 1
	Depart counter	Num-depart Num-cmp	1 1..*
account	Agency counter	Num-ags Num-cmp	1 1..*
output	counter agency	Num-cmp Num-ags	0..* 1
input	history counter	Num-cmp month year	1..* 1

Table 2.2: Dictionary Links

4. Conclusion: This chapter has been dedicated to the design diagrams, and the transition to relational model that allowed us to model our information system that will be represented in the next chapter.

Chapter 3:

Implementation and execution

1. Introduction:

The implementation is a step of translating the proposed structures of the design, in a programming language incorporating a database with maximum safety. In this chapter, we define the system architecture and development of technologies that we used.

2. The development environments:

XAMPP:

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in thePHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP).[4]

Dreamweaver:

Dreamweaver is a WYSIWYG editor (what you see is what you get) for designing, coding and development of websites, pages and Web applications. Whatever the work environment used, Dreamweaver provides tools that help you build Web applications. Macromedia publishes this software. [5]

PHP:

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language, which means that it is the server (the machine that hosts the website in question) that will interpret the PHP code and generate a code that can be interpreted by the software. In most cases, the generated code is the HTML to be read by the browser but can be used for other languages or formats like WML, GIF, PDF, SVG, etc. It is designed to enable the creation of dynamic applications, mostly Web applications. PHP can be installed on the main web servers on the market, the most common being are IIS and Apache.

This coupling allows you to retrieve information from a database, a file system (content files and the tree) or simply sent by the browser of data to be read or stored for use later.

In a Web use, execution of PHP code is as follow: when a visitor requests to see a Web page, the browser sends a request to an HTTP server. If the page contains PHP, the PHP interpreter treats the server and returns the generated code (HTML). [6]

3. Database Management:

To implement our database "Sonelgaz", we used the data base creation environment PHPMyAdmin and database management system MySQL. The table below presents our database.

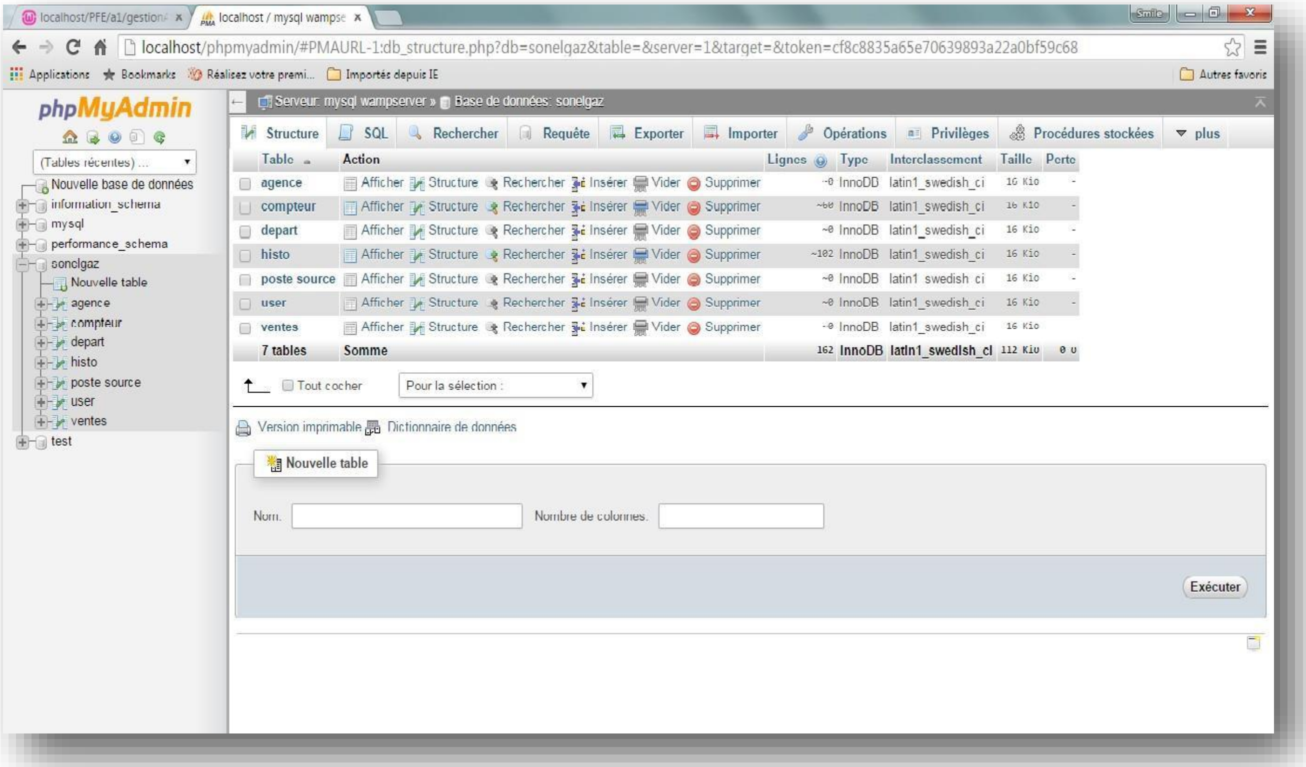


Table	Action	Lignes	Type	Interclassement	Taille	Perte
agence	Afficher Structure Rechercher Insérer Vider Supprimer	~0	InnoDB	latin1_swedish_ci	16 Kio	-
compteur	Afficher Structure Rechercher Insérer Vider Supprimer	~60	InnoDB	latin1_swedish_ci	16 Kio	-
depart	Afficher Structure Rechercher Insérer Vider Supprimer	~0	InnoDB	latin1_swedish_ci	16 Kio	-
histo	Afficher Structure Rechercher Insérer Vider Supprimer	~102	InnoDB	latin1_swedish_ci	16 Kio	-
poste source	Afficher Structure Rechercher Insérer Vider Supprimer	~0	InnoDB	latin1_swedish_ci	16 Kio	-
user	Afficher Structure Rechercher Insérer Vider Supprimer	~0	InnoDB	latin1_swedish_ci	16 Kio	-
ventes	Afficher Structure Rechercher Insérer Vider Supprimer	~0	InnoDB	latin1_swedish_ci	16 Kio	
7 tables	Somme	162	InnoDB	latin1_swedish_ci	112 Kio	0 o

Figure 3.1: SONELGAZ database

4. Implementation and test:

4.1 Authentication page:

Login window for the application. This window is displayed at the startup of the application which contains the login authentication.

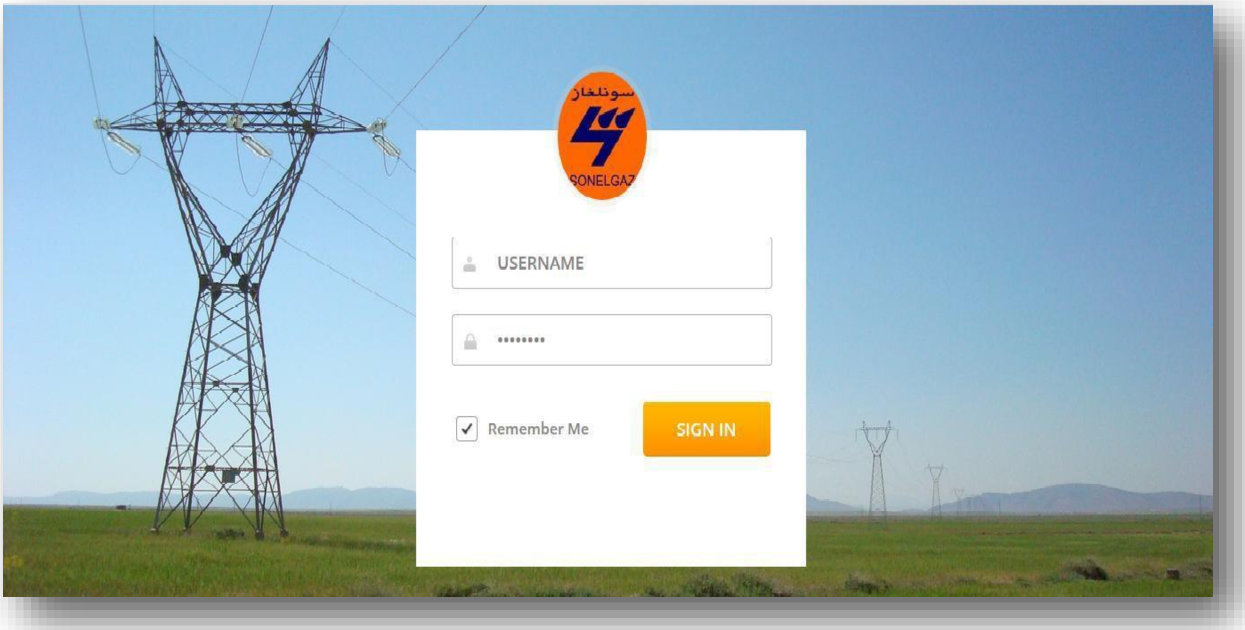


Figure 3.2. Authentication page

4.2 Index Page:

This page allows the agent to access:

- Page "consultation of results".
- Page "counter management".
- Page " agencies management ".
- Page "post source and departure".

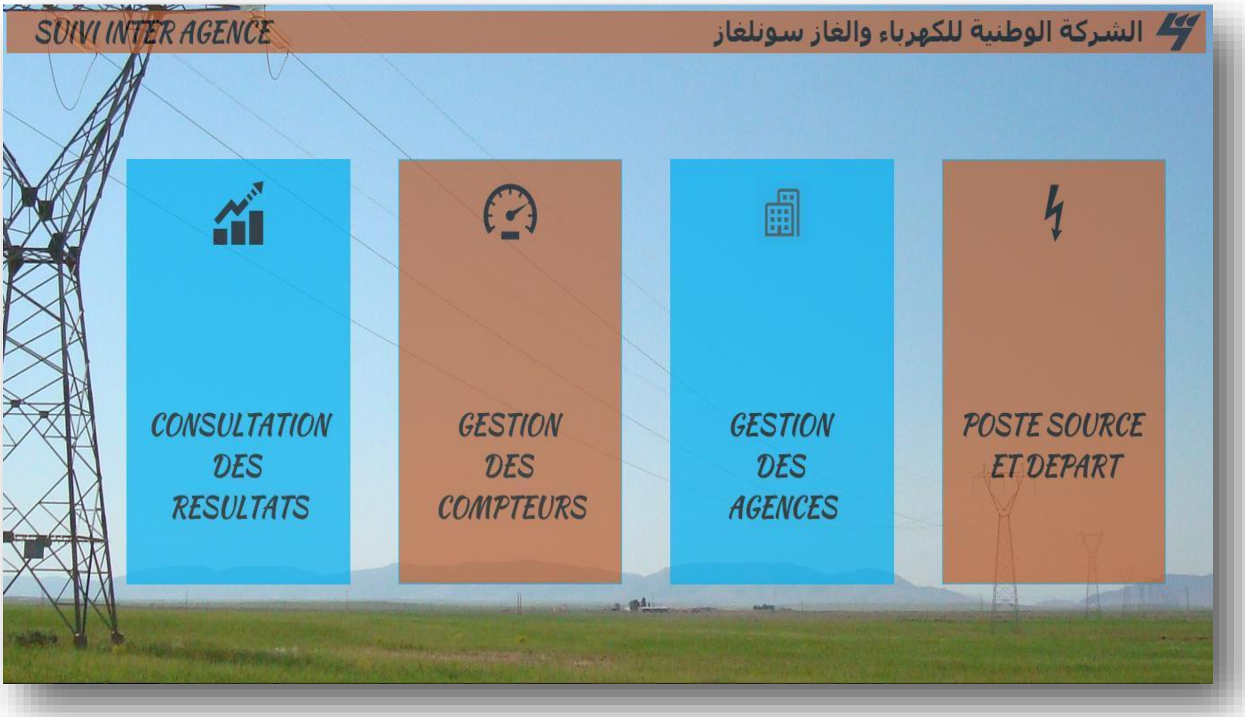


Figure 3.3 : Index Page.

4.3. Result Consultation page:

This page allows the agent to access to:

- Page "loss rates".
- Page "sales".

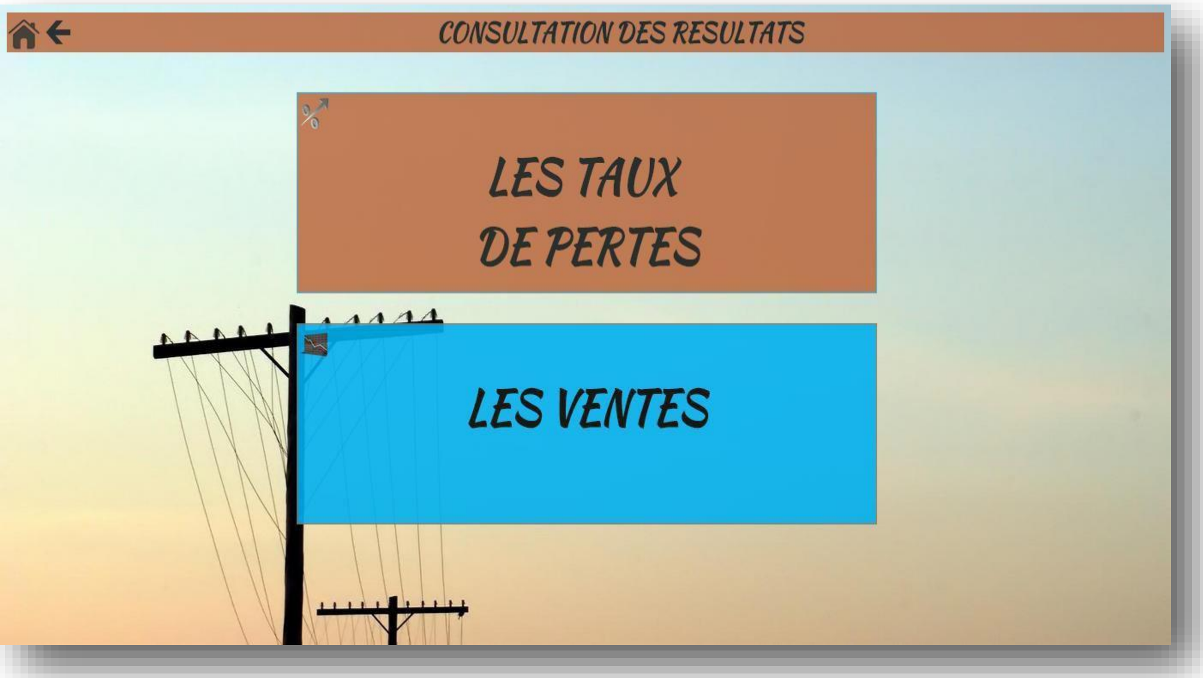


Figure 3.4: Page of results consultation.

4.4. Loss rates page:

This page allows the agent to check the losses of each agency in a month and a year:

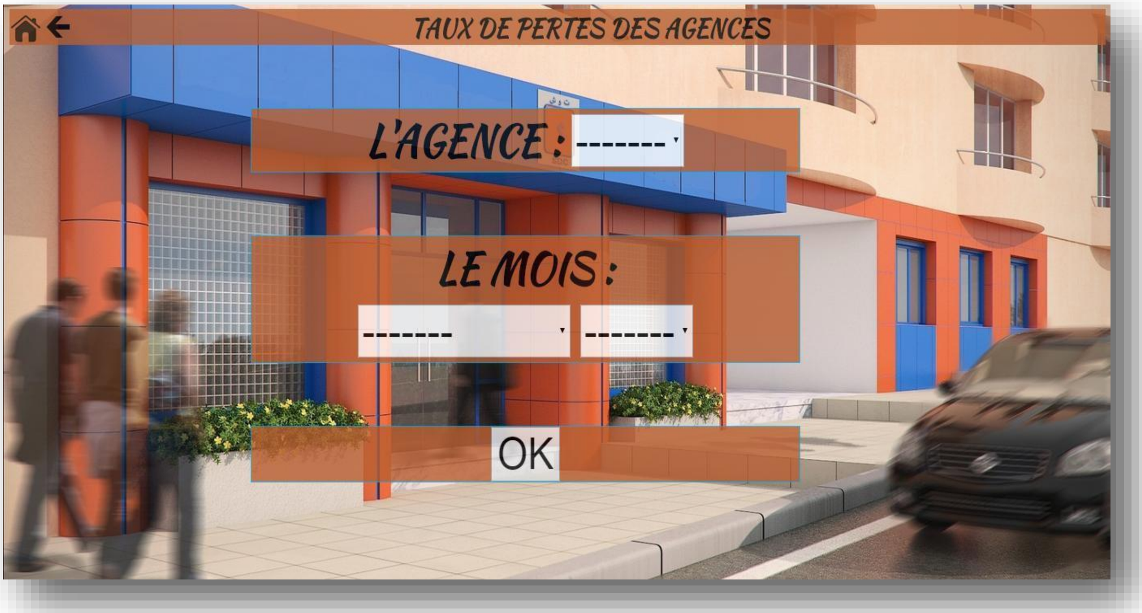


Figure 3.5: Loss rates Page

4.5. Counter management Page:

This page gives permission for the agent:

• Add counter: it allows the agent to add a new counter of a new client.

• Change counter’s data: it allows the agent to change the information of the targeted counter

• Clear counter: it’s used every ending year to calculate the consomation of the new year only.

• Entering index: to load the index of the counter.

• Star the treatment of XML file.

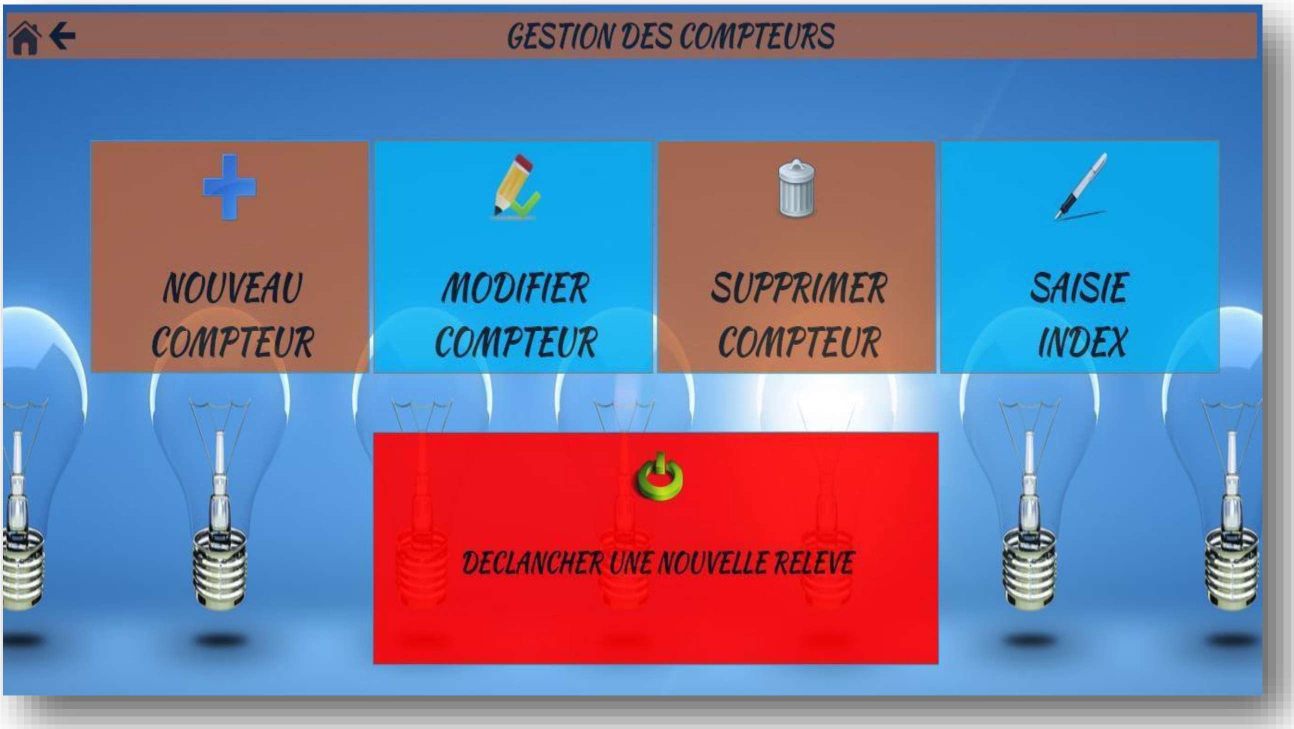


Figure 3.6. Counter management page.

<u>4.6. Agencies Management Page</u>:

This page allows the agent to do the following:

- Add agency: it gives the agent the possibility of adding a new agency that can be a a consumer or a distributer.
- Delete agency.
- Entering sales: loading the sales of a specific agency.

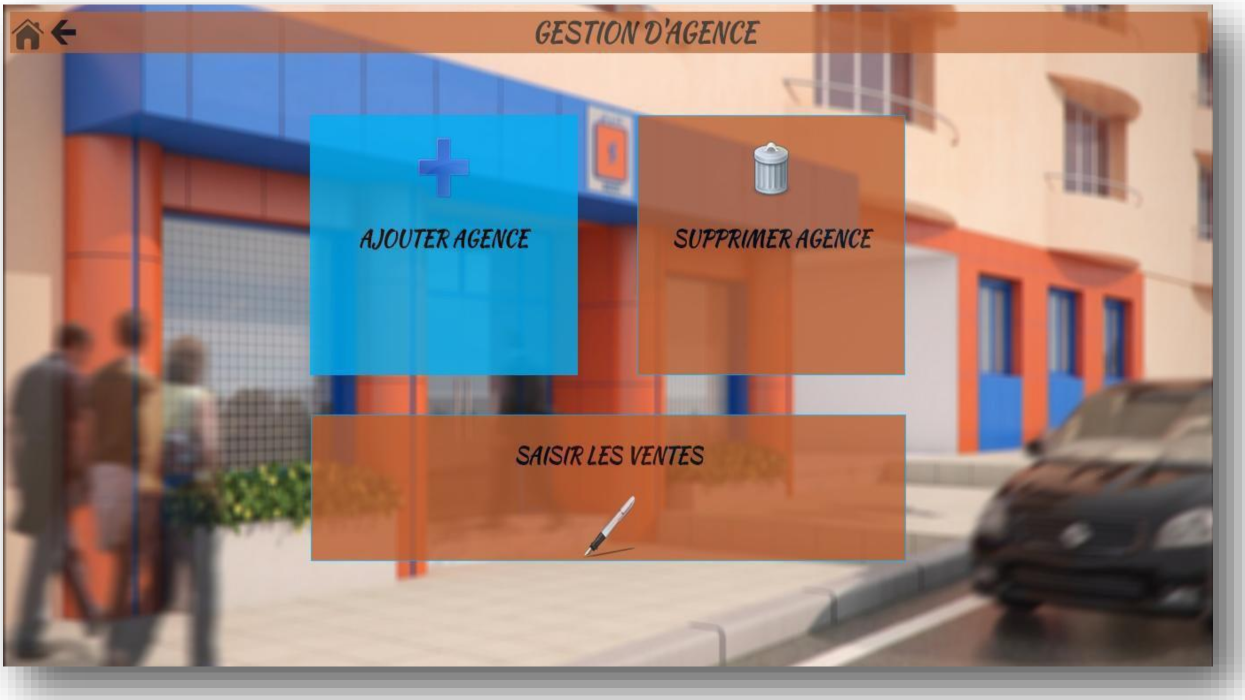


Figure 3.7: Page of agencies management

4.7. Post Source and Departure Page:

This page gives permission to:

- Add departure: add a new source of electricity which can be a producer or an agency.
- Delete departure.
- Add electricity post source.
- Delete electricity post source.

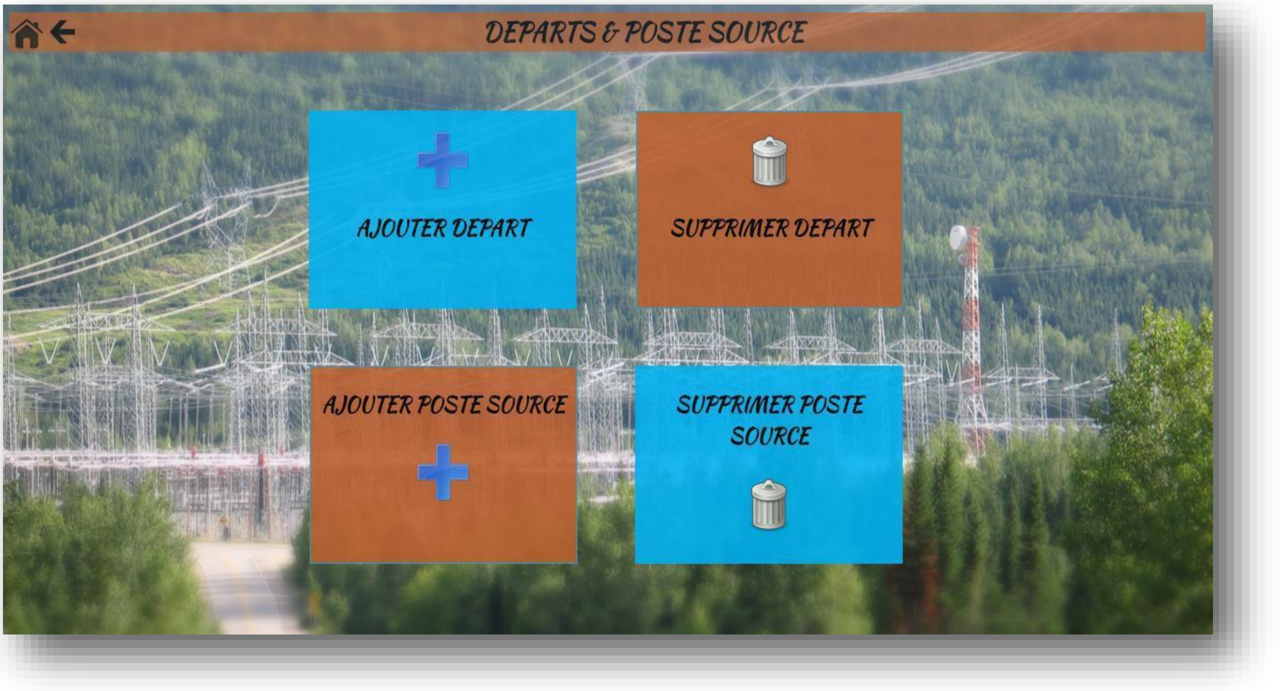


Figure 3.8: Post source and departure page

5. Conclusion:

In this final chapter, the study was based on the realization and implementation of the application with a presentation of development tools and the most significant interfaces of it.

Conclusion

In this thesis, we have presented the various stages of the design and implementation of our application for the management of electricity loss system.

We studied the conceptual design of our solution using the UML modeling language; we have implemented and realized our solution using the SGBD, MySQL, the PHP programming language and some scripts to beautify our interface.

This work helps to design and develop an application that automatically manages the information to save time which is used nowadays by SONELGAZ Agency of Blida.

Finally, I would like that this study is a starting point for a new management of electrical losses.

References:

[1] ACE6000 Commercial Electricity Meter Portfolio

[2] FASTRACK modem M1203 manual.

[3] The Object Primer 3rd Edition: The Modeling Development with UML 2.

[4] "Interview with Kai Seidler from the XAMPP project". Retrieved 2015-06-07.

[5] Press Releases. Adobe, Inc. Retrieved 15 November 2011.

[6] PHP on Hormones – history of PHP presentation by Rasmus Lerdorf given at the MySQL Conference in Santa Clara, California. Retrieved 2009-12-11.