



University of Hassan II Casablanca
Ben M'Sick Faculty of Sciences
Department of Mathematics and Computer Science



Bachelor in Mathematics and Computer Science

Project Graduation



Stock Management Application for Android

MyStock

Produced by: KHOUNA RIDA

KILANI HASSAN

Academic supervisor: Mr.Merzak Abdelaziz

Jury: Mr.Nouh Said

Contents

Contents	2
Tables.....	3
Figures.....	4
Thanks.....	6
General Introduction :.....	7
Chapter 1 : General framework.....	8
1. Introduction :.....	8
2. Project presentation:	8
3. Methodology and formalisms adopted:	8
3.1 Introduction :	8
3.2 Unified processes:	9
4. Conclusion :	9
Chapter 2: Project study.....	10
1. Introduction :.....	10
2. Study of the existing :	10
3. Criteria for choosing the Android system :	12
3.1 Platform flexibility:.....	12
3.2 Ease of development :.....	12
3.3 Growth in the Smartphone market:.....	12
4. SOAP Web Service:	13
4.1 Introduction :	13
4.2 Le développement des protocoles applicatifs :.....	13
4.3 Choice of SOAP :	14
5. Conclusion :	15
Chapter 3: Specification and needs analysis	16
1. Introduction :.....	16
2. Requirements specification:.....	16
2.1 Functional needs:	16
2.2 Non-functional needs :.....	16
3. Needs Analysis:.....	17
3.1 Identification of actors and use cases:	17
3.2 Use case diagram:	17
4. Conclusion :	27
Chapter 4: Conception	28
1. Introduction :.....	28
2. Sequence diagrams:.....	28

3.	Activity diagrams:	35
4.	Class diagram:	40
5.	Relational model:	41
6.	Conclusion :	42
Chapter 5: Production		43
1.	Introduction :	43
2.	System architecture :	43
3.	Work environments :	44
3.1	Material environment:.....	44
3.2	Software environment:.....	44
4.	SOAP web service implementation :	45
5.	Android side implementation :	46
5.1	Choice of Android version :	46
5.2	Used Libraries :.....	47
6.	Application interfaces :	48
6.1	Online mode :.....	48
6.2	Offline mode :	62
7.	Adding Options:	68
7.1	The search option in the Menu Bar:.....	68
7.2	The Refresh options	69
7.3	The quick access menu.....	70
7.4	Color change option	70
Conclusion and perspectives		71
Bibliography et Webography		72

Tables

Table 1: Authentication scenario	19
Table 2: Consult inventory use case scenario	23
Table 3: Item Search Scenario	24
Table 4: Item Printing Scenario	25
Table 5: Base Initialization Scenario	26
Table 6: All the software tools used in the production phase	45
Table 7: Distribution of Android API versions	46

Figures

Figure 1 - Division by OS of worldwide smartphone deliveries (millions of units)	13
Figure 2: Hotel case manager global use case diagram	18
Figure 3: Inventory Consultation use case diagram	20
Figure 4: Article Search use case diagram	24
Figure 5: Repository Printing use case diagram	25
Figure 6: Database Initialization use case diagram	26
Figure 7: Authentication sequence diagram	29
Figure 8: Consult inventory Sequence Diagram	30
Figure 9: Adding Item Sequence Diagram	31
Figure 10: Item Modification Sequence Diagram	32
Figure 11: Article Search Sequence Diagram	33
Figure 12: Printing warehouse sequence diagram	34
Figure 13: Database Initialization Sequence Diagram	35
Figure 14: Authentication activity diagram	36
Figure 15: Inventory Consultation Activity Diagram	37
Figure 16: Item Search Activity Diagram	38
Figure 17: Inventory Printing Activity Diagram	39
Figure 18: Class diagram	40
Figure 19: MLD	41
Figure 20: System architecture	43
Figure 21: Distribution of Android API versions	46
Figure 22: Authentication home interface	48
Figure 23: Error message when authentication fails	49
Figure 24: Manager's home interface if the parameters are valid	49
Figure 25: Manager authentication interface	49
Figure 26: Search type choice interface	50
Figure 27: Items consultation interface with the NO filtering option	51
Figure 28: Interface to select the warehouse Where the item is located	51
Figure 29: Category list interface	51
Figure 30: List of Families interface	52
Figure 31: List of Subfamilies interface	52
Figure 32: : Items list interface	52
Figure 33: Items consultation interface with the Filtering option YES	53
Figure 34: Items list interface	53
Figure 35: Interface item info in edit mode	54
Figure 36: Interface item info	54
Figure 37: Interface item info in edit mode	55
Figure 38: List of items interface	55
Figure 39: List of items interface	56
Figure 40: Interface for Editing Items in Online Mode	56
Figure 41: List of items interface	57
Figure 42: Add item interface	57
Figure 43: Add item interface	58

Figure 44: Interface for Adding Items in Online Mode	58
Figure 45: Item Scanner interface	59
Figure 46: Item Search Interface	59
Figure 47: Item Search Interface	59
Figure 48: Inventory Menu Interface.....	60
Figure 49: Warehouse list interface	60
Figure 50: Warehouse Print Interface	60
Figure 51: Main Menu Interface.....	61
Figure 52: Logout Interface.....	61
Figure 53: List of items interface	62
Figure 54: Item info interface	62
Figure 55: Item info interface	63
Figure 56: List of items interface	63
Figure 57: Alert Sends change	63
Figure 58: Alert Sends change	64
Figure 59: Interface for Editing Items in Offline Mode	64
Figure 60: List of items interface	65
Figure 61: Interface Adding the item	65
Figure 62: Item list interface.....	66
Figure 63: Alert Sends change	66
Figure 64: Alert Sends change	67
Figure 65: Interface sends update to server	67
Figure 66: Optimized search option interface.....	68
Figure 67: List of items interface	69
Figure 68: Data update interface.....	69
Figure 69: Quick Access Option Interface.....	70
Figure 70: List of warehouses interface with color option	70

Thanks

This report is the fruit of three months of work on a project that we had the pleasure of carrying out. Before starting the report of this professional experience, it seems important to us to start this document by expressing our gratitude to Our academic supervisor Mr.Marzak Abdelaziz, who accompanied us throughout this experience with a lot of patience and pedagogy.

It is thanks to his help, his availability and his valuable recommendations that we were able to complete this work. We hope that he will find in this memoir the testimony of our deep gratitude. Finally, we would like to thank the members of the jury for agreeing to evaluate our work.

General Introduction :

The continuous and flexible way of accessing information through various services is becoming a new paradigm for programming and organizing operations.

Web and mobile applications are becoming more popular and are of increasing interest to companies concerned with improving their technological image and meeting performance requirements. The Smartphone market is therefore experiencing a real boom in which the usual players are trying to rush. Google, having realized the potential of this market, decided to enter it by buying a startup working on an open operating system for mobile terminal: **Android**.

During the internship of the end of studies project, we were asked to design, develop and integrate an Android application, allowing to follow all the reports at all times, in addition, it allows storekeepers to consult the hotel inventory.

This report can be subdivided into five chapters:

- The first chapter «**Project framework**» consists of the presentation of the project framework, then the planning of the project.
- The second chapter entitled «**Study of the project**» which we will do a study of the existing, the criteria for choosing Android system as well as SOAP web services.
- The third chapter entitled «**Specification and needs analysis**» is composed of a description of the different modules of the application (functional and non-functional needs) related to the nature of development on mobile as well as a description of the expected behavior of the system.
- The fourth chapter is devoted to design, in which the conceptual and logical specifics of the work presented by the class diagram will be detailed, as well as the sequence diagrams.
- The fifth chapter «**Realization**» will be reserved to present the hardware and software environment with the stages of implementation of the solution as well as the tests and the validation of the application.

Finally, we close this report with a conclusion in which we summarize our solution and setting out some future perspectives.

Chapter 1 : General framework

1. Introduction :

This project is part of the end of studies project at the Ben M'Sik Faculty of Sciences for obtaining the bachelor's degree in Mathematics and Computer Science.

In this chapter we will introduce the general framework of the project. Then, we will pass to present the methodology and formalities adopted for work planning.

2. Project presentation:

More and more companies need to make their applications accessible on the web. There are many motivations: to widen the audience of users, to communicate existing applications.

Ensuring stock management via an on-board application represents a considerable advantage for senior managers and for hotels to improve their inventory management. It is in this context that this end-of-study project is located, which aims to design and develop such an application on Android. More particularly, this application offers direct access to the Inventory Items for instant monitoring of any type of change applied to the quantity. It is also useful as a search method by ensuring two search options, the first by the Article code, and the second by the bar code after having scanned it on the Article.

The application is characterized by the STANDALONE option which allows the storekeeper to consult the inventory in two modes:

1. Online Mode.
2. Offline mode.

3. Methodology and formalisms adopted:

3.1 Introduction :

A development process defines a sequence of steps, in part ordered, which contributes to obtaining a software system or the evolution of an existing system, to produce quality software, which meets the needs of users in predictable times and costs.

Although there are a multitude of software development methodologies, choosing a method for a given project is a crucial decision. Because often the end product does not correspond to the actual need of the moment.

3.2 Unified processes:

A unified process is a process built on UML (Unified Modeling Language). Unified processes are the result of unification, not processes, but more precisely the best practices of object development.

A unified process is distinguished by the following characteristics:

- ✓ **Iterative:** The software requires a progressive understanding of the problem through successive refinements and developing an effective solution incrementally by multiple iterations.
- ✓ **Risk-driven:** The major causes of failure of a software project must be eliminated as a priority
- ✓ **Centered on architecture:** The choice of software architecture is made during the first phases of software development. The design of the product components is based on this choice.
- ✓ **Driven by use cases:** The process is driven by user needs presented by use cases.

4. Conclusion :

Throughout this chapter, we have generally approached the framework of our graduation project. First, we introduced our application and the needs leading to its realization. We then presented a brief study on the different methodologies to justify our methodological and conceptual choice, which will allow us to achieve the objectives of our project.

We will pass in the next chapter to the determination of the problems presented by this subject as well as the need for the transition from a software version to a mobile version and the criterion for choosing the Android system as a development platform.

Chapter 2: Project study

1. Introduction :

To succeed in the design and development phases, a preliminary study must be well developed. We begin, first, by presenting what is existing. Then we discuss the new features brought by our software solution. Finally, we justify through a study, our choice of technical solutions adopted.

2. Study of the existing :

Before starting our work, it is necessary to make a presentation of INFOR VR / STOCKS (VISUAL RESTAURANT/STOCKS) the ERP which has a relationship with our Android application.

INFOR Visual Restaurant for point of sale allow to:

- Avoid mistakes between waiters and kitchen staff.
- Identify the most popular menu items.
- Manage inventory control requirements regardless of the restaurant's seating capacity.
- Boost service, productivity and profits.
- Make your servers operational quickly and sustainably.

Intended for use with a touch screen and with a user-friendly graphical interface, INFOR Visual Restaurant is easy to master and use by servers, managers and owners, whether they work in the dining room or in the kitchen. or on back office management. Fully modular and easy to upgrade, Visual Restaurant helps to operate each restaurant center or profit center of establishments in local mode or via wireless technology.

It allows in particular to manage:

- Room service
- Bars
- Restaurants
- Terraces
- Banquets

Native Mobile App

- Shops

INFOR Visual Restaurant also supports several types of operations:

- ✓ Table service
- ✓ High-end fast food
- ✓ Takeaway
- ✓ Gastronomic restaurant
- ✓ Bar service

INFOR Visual Restaurant can be integrated with other INFOR solutions, allowing access to all the essential information. The Customer Loyalty, Reservation and Waiting List modules which allow tracking of reservations, improving the use of tables and providing better service to customers.

This new computer system, designed according to the most modern standards and technologies .NET with in particular a grouping of all the data within a centralized database linked to web services, is based on a modern and secure infrastructure.

We cite below the weaknesses of the current system:

- ✓ A waste of time and costs (to the detriment of the customer or the service provider).
- ✓ Managers had to obtain information in a non-automated way.
- ✓ Business decisions were made primarily on the intuition of the executive.
- ✓ The control of the organization of work and the functioning of the team is completely manual (loss of time and possibility of loss of information).

For this fact, we tried to find a mobile and reliable solution.

To achieve the objectives and meet the requirements, the main idea is to design and develop a mobile application for users of the Android terminal.

The project consists of developing :

- ✓ A functional Android application for any hotel, allowing to offer all the functionalities for the storekeepers, responsible for the management of the hotel stock.
- ✓ A SOAP web service module on the .NET Framework which ensures synchronization and communication between the Android application and the application server.

We will then present the criteria that led us to choose Android systems as well as SOAP as the protocol for the web service module.

3. Criteria for choosing the Android system :

3.1 Platform flexibility:

Unlike the Apple Store, Google Play will not have restrictions on third party development. In the United States, AT&T had acquired the rights to sell Apple iPhones for the next five years from the date of its release. And in the case of the BlackBerry, it is not a device entirely independent of the operator, since the major part of the sale passes through its various supports, worldwide.

This approach had left people frustrated, tied to a monopoly mobile operator, regardless of their desire to choose another operator. But, since Android is an open source operating system, it could take advantage of the benefits of device independence and service provider independence.

3.2 Ease of development :

Although consumers could benefit from a low cost of Android Smartphones, the developers had unrestricted personalization rights. From a developer point of view, Android has several advantages,

Of which we can cite:

- ✓ The entire application platform can be used and replaced by selective elements.
- ✓ The DALVIK virtual machine reinforces the power of management systems.
- ✓ Support for 2D and 3D graphics (OpenGL ES 1.0), so lots of projects for animation developers.
- ✓ Reinforced data storage (using the SQLITE platform)
- ✓ Support for GSM, EDGE, GPRS, UMTS 3G, HSDPA, HSUPA, LTE, Wi-Fi network applications
(Depends on hardware)
- ✓ Android development environment includes an emulator, a debugger and a plug-in for the Android Studios IDE.

3.3 Growth in the Smartphone market:

Nothing seems to be able to hinder the overwhelming dominance of Android: with a market share of 78.6% in 2013 (+9.6 points in one year), it crushes the competition. IOS still limits the Breakage and monopolizes the high-end. Behind, BlackBerry is gradually disappearing (1.9%), overtaken by a Windows Phone (3.3%) used mainly by Nokia (89.3%).

Native Mobile App

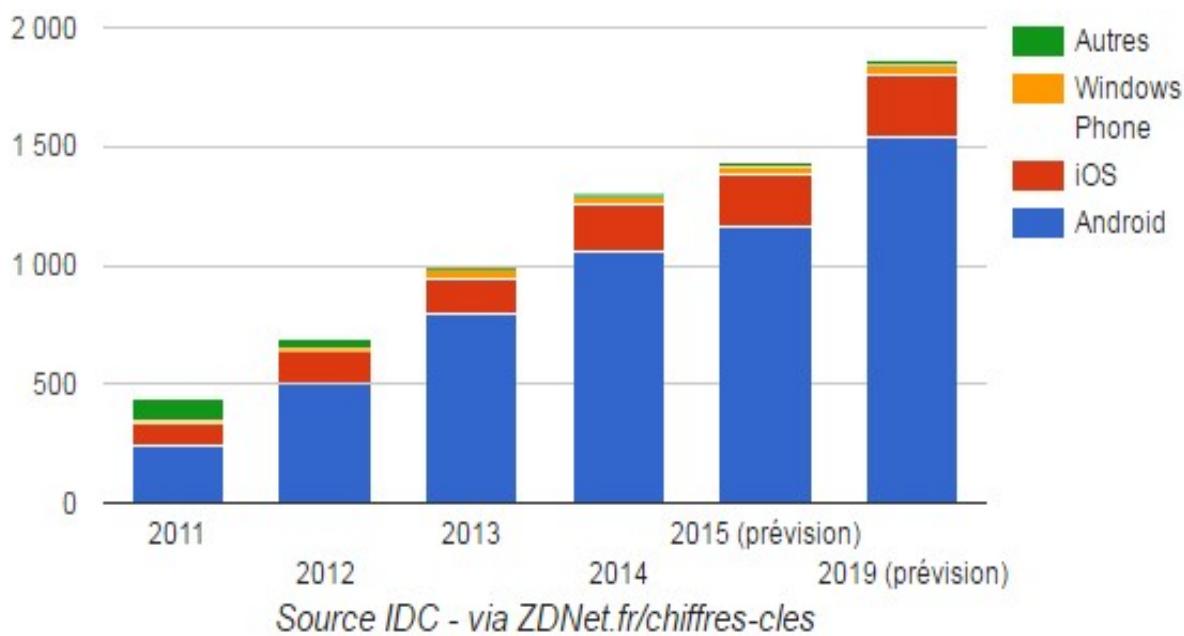


Figure 1 - Division by OS of worldwide smartphone deliveries (millions of units)

More than a billion Androphones in 2015 - Android has benefited from the growth of the global smartphone market (+ 9.8%). Between 2011 and 2015, the number of androphones increased from 243.5 to 1,016 million units. Android's market share increased by more than 30 points over the same period, jumping from 49.2% to more than 81% in 2015.

Android benefits from the entry level. However, and despite a comfortable advance, the Android ecosystem is experiencing some changes. Due to Samsung's sharp slowdown in 2014, the growth in smartphone deliveries is now based more on smaller manufacturers. Despite everything, Korean remains the world number one on the market.

4. SOAP Web Service:

4.1 Introduction :

Widen the audience of users, sell online services or communicate existing applications brings us back to asking questions: what architecture to choose where

Design? What format should I use to exchange data on the web? And should we use an existing application protocol or develop a protocol adapted to our needs?

4.2 Le développement des protocoles applicatifs :

Making your applications accessible on the web involves defining an application protocol and a data format. An application protocol is a high-level, user-oriented protocol. Non-functional aspects like the transport of information are delegated to the lower layers like TCP / IP or even HTTP and SMTP.

We have seen that there are many motivations for companies to deploy web services today. Whatever the motivation, the problem of interoperability is present. Even if today the applications that must interact are known, it is very likely that a new application will one day be integrated into the system.

Another aspect of the interoperability issue is the format of the data exchanged. There are so many different functional areas that it seems difficult to specify a data format that would effectively express all needs. XML today defines a set of information structuring rules that allow us to see all the benefits that could be obtained from truly interoperable systems. A recent example is the RSS data format which allows different customers to consume one type of information regardless of how it was produced.

The first idea to solve the problem of protocol interoperability could be to develop a generic protocol allowing all applications to communicate on the web. It was the approach adopted by XML-RPC then SOAP. These specifications use the remote method call paradigm. Messages are sent over the web in a standardized XML format.

This approach has two advantages:

- ✓ It facilitates the developer's work since the interaction model by calling business methods is the same as that of the application (most often developed in the object or procedural paradigm)
- ✓ And improves an aspect of interoperability by specifying the format in which information should be encoded.

However, by describing more and more exhaustively these different interaction models, SOAP has become a Framework for the development of application protocols and no longer a generic protocol. Two SOAP web services are not directly interoperable. In order for them to be, they have to agree on how they use SOAP.

4.3 Choice of SOAP :

The choice of development of a web service module is mainly imposed since a set of web services were at the beginning already developed and used in VR / STOCKS.

We have seen that SOAP today is more of an application protocol framework than a generic protocol. Its specification describes all the types of exchanges possible between applications. The advantage of this approach is that it allows the development of tools for automatic generation of web services. However, two applications that want to chat with each other have to agree on how they use SOAP, thereby hampering interoperability.

5. Conclusion :

In this chapter we have presented a global view of the existing software solution and what brings us back to moving to a mobile version, as well as the criteria for choosing the Android system. In the next chapter we will start the specification and needs analysis part.

Chapter 3: Specification and needs analysis

1. Introduction :

Following the presentation phase of the project framework and the study of the Project, we will invoke in this part the specification of our application. This step is a necessary step to organize the development of an application. It constitutes the starting point for our work. It makes it possible to define the different functionalities that it is capable of achieving to best meet the requirements of the business and the needs of customers. Thus, we will present the different services and features that the application must provide to the user. We start with a general description of the application, before presenting a detailed description of the sections making up the application. Afterwards, we will establish a functional analysis which can then be modeled by the use case diagram.

2. Requirements specification:

This part takes care of the descriptive part of the needs expressed by the customers. These needs will allow us to identify the features that will need to be implemented in our application.

2.1 Functional needs:

A functional specification for the system allows the description of its different functionalities for its realization.

Through the UML language use case diagrams, we will present the main functionalities that the system must provide, as well as the list of actors who interact with them. A use case makes it possible to highlight the functional relationships between the actors and the system studied.

The use case diagram makes it possible to visually represent a sequence of actions carried out by a system, producing a result on an actor, called main actor, and this independently of its internal functioning.

An actor, in the UML sense, represents the role of an external entity (human user or not) interacting with the system.

2.2 Non-functional needs :

In addition to the functional requirements, a set of non-functional requirements must be verified. This will determine the quality of response from our application and the level of user satisfaction.

- **Ergonomics and flexibility:** Our application must offer a user-friendly and ergonomic interface that can be used by the user by considering all possible interactions with the Android Smartphone.

Native Mobile App

- **Reliability:** We also need to check the reliability of our application which can be run throughout the working hours without errors.
- **Security:** Our application must guarantee the security of the exchanged data.
- **Extensibility:** Our application is extensible. It allows the addition of new modules and functionalities in the future.
- **Compatibility :** One of the most important points when developing an application on a mobile environment is to ensure its compatibility with any version of the system without forgetting its compatibility with third-party applications and elements (such as web services).

3. Needs Analysis:

3.1 Identification of actors and use cases:

The storekeeper: This actor is one of the managers of the hotel. He has access to the following modules:

- **Inventory consultation:** Find all the practical information (Item Code, Item Label, Warehouse, Category, Family and Subfamily) in the form of a table.
- **Search Item:** It contains two options, either with the Item Code or the Item Bar Code.
- **Scan an Item:** Scan the Item Barcode to change its Quantity.
- **Initialize the Database:** Drop the local base.
- **Logout:** Allows you to disconnect from the establishment.

3.2 Use case diagram:

The general use case diagram gives us an overview of the different services offered by our application. It presents a general vision of the functionalities provided by our system towards the user. The following figures illustrate the general use case diagrams for actors:

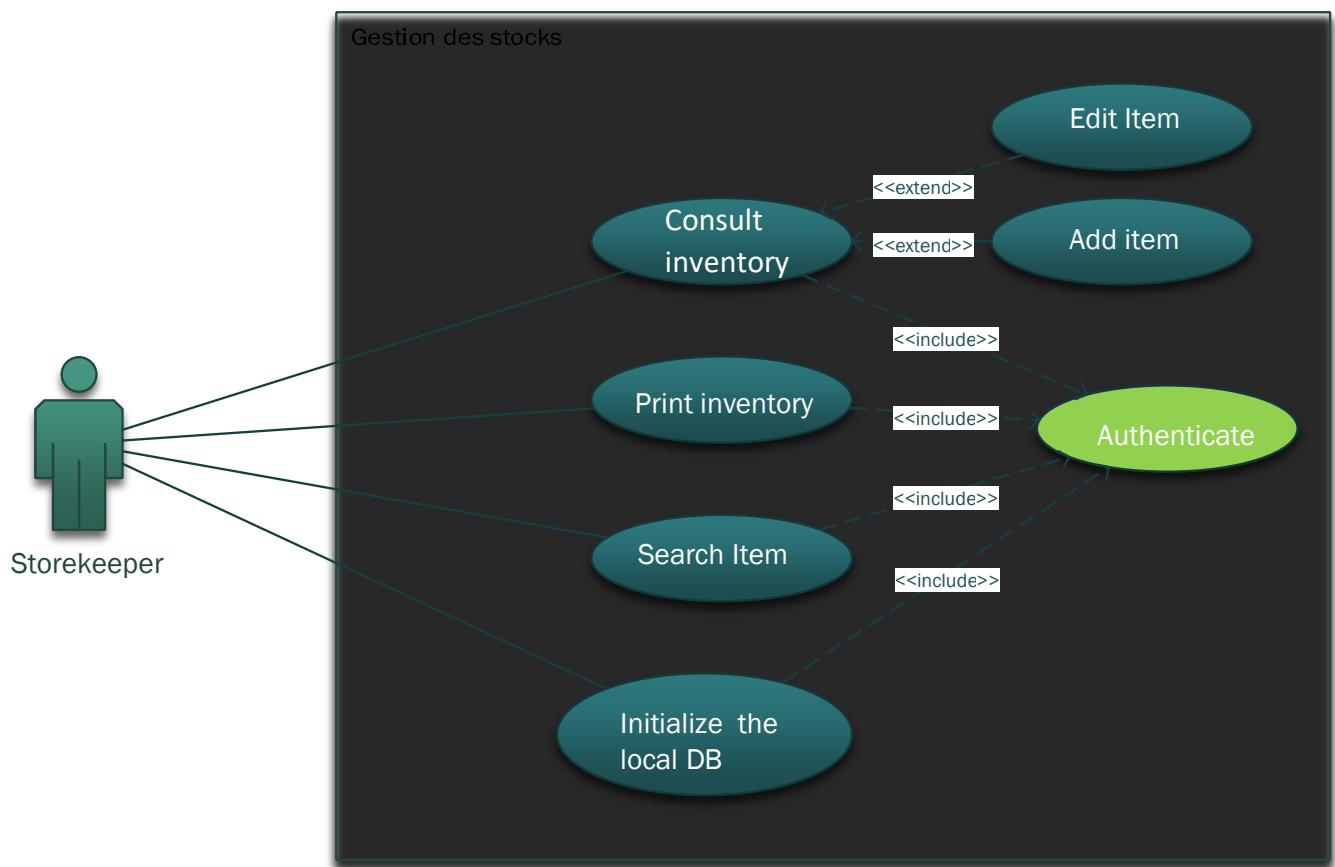


Figure 2: Hotel case manager global use case diagram

- Details of the "Authentication" use case :

Case number	01
Case name	Authentification
Actor	The storekeeper
Description	Process user access to the application
Precondition(s)	<ul style="list-style-type: none"> - Installation of the Application on a mobile or tablet. - Internet connection is available.
Post condition(s)	The "Inventory Menu" interface is displayed
Scenario	<ol style="list-style-type: none"> 1. The user launches the application 2. The system displays the authentication interface 3. User enters establishment number and password 4. The system checks the validation of the establishment number and password 5. The system retrieves the IP address from the server 6. The system checks the availability of Internet access 7. The system checks the connection with the server 8. System sends establishment number and password to server 9. The server checks the authentication parameters 10. System saves user data for next entry 11. Access the menu page
Exceptions	4a- The system displays "empty field" 4b- The system displays "incorrect login" 5a- The system displays a message box to enter the IP address 6a- The system displays "Internet connection problem" 6a- The system displays "connection problem with the server" 11a- The system displays "The password or establishment number is incorrect"

Table 1: Authentication scenario

Native Mobile App

- Details of the "Consult Inventory" use case:

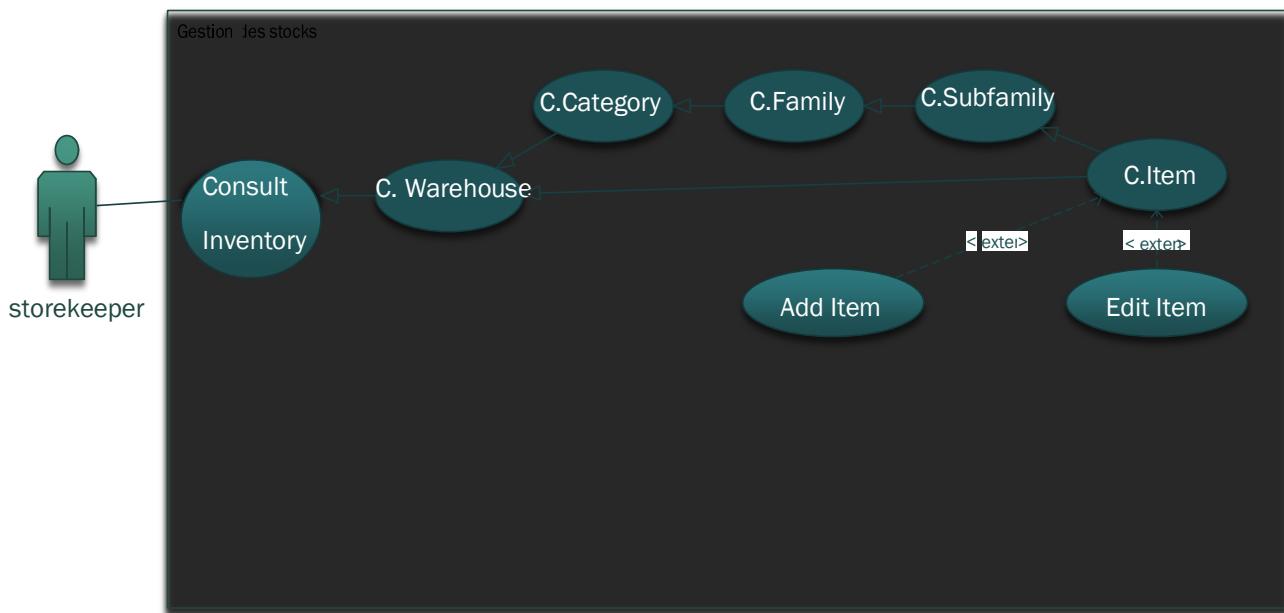


Figure 3: Inventory Consultation use case diagram

Native Mobile App

Case number	02
Case name	Consult the Inventory
Actor	The storekeeper
Description	Allows the storekeeper to consult the group of warehouses, its categories, families under families and items
Precondition(s)	<ul style="list-style-type: none"> - Installation of the Application on a mobile or tablet. - The storekeeper connects to the application.
Post condition(s)	<ul style="list-style-type: none"> - The "Warehouses" interface is displayed - The "Categories" interface is displayed - The "Families" interface is displayed - The "The Sub Families" interface is displayed - The "Items" interface is displayed - The "Item" interface is displayed
Scenario	<ol style="list-style-type: none"> 1. The user clicks on the Enter Inventory button 2. The system displays a list of warehouses 3. The user chooses a warehouse 4. The system displays a list of categories 5. The user chooses a category 6. The system displays a list of families 7. User chooses a family 8. The system displays a list of subfamilies 9. The user chooses a subfamily 10. The system displays a list of Items 11. The User chooses an Item 12. The system displays a page that contains the information of the Item
Exceptions	Display an "empty list" message

Native Mobile App

Case number	02.01
Case name	Edit Item
Actor	The storekeeper
Description	Allows the warehouse worker to modify the quantity of an item
Precondition(s)	<ul style="list-style-type: none"> - Installation of the Application on a mobile or tablet. - The storekeeper connects to the application.
Post condition(s)	"Sending completed successfully" message
Scenario	<ol style="list-style-type: none"> 1. The system displays a page containing the information of the item 2. The user clicks on the Edit button 3. User enters a new quantity 4. The user clicks on the Validate button 5. The system saves the new quantity 6. Return to the list of items 7. The user clicks on the button Send the modification 8. System sends change to server
Exceptions	<p>4a- User clicks Cancel button Display of the message "Sending not performed"</p>

Native Mobile App

Case number	02.02
Case name	Add Item
Actor	The storekeeper
Description	Allows the storekeeper to add a new item
Precondition(s)	<ul style="list-style-type: none"> - Installation of the Application on a mobile or tablet. - The storekeeper connects to the application. - The consultation filter is deactivated - Access to the "Items" page
Post condition(s)	"Addition completed successfully" message "Sending completed successfully" message
Scenario	<ol style="list-style-type: none"> 1. The user clicks on the Add button 2. The system displays a form 3. User enters item information and scans barcode if available 4. The user clicks on the Add button 5. The system saves the new article 6. Return to the list of articles 7. The user clicks on the button Send the modification 8. System sends change to server
Exceptions	4a- User clicks Cancel button Display a message "Adding not done" Display a message "Sending not completed"

Table 2: Consult inventory use case scenario

Native Mobile App

- Details of the "Item Search" use case:

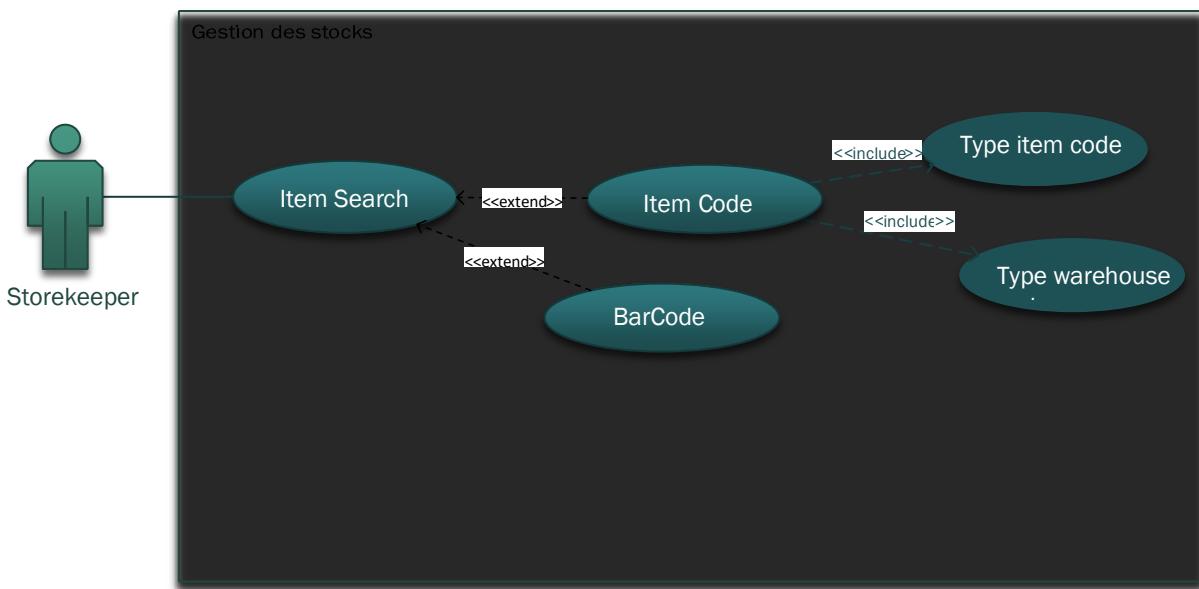


Figure 4: Article Search use case diagram

Case number	03
Case name	Search an item
Actor	The storekeeper
Description	Allows the storekeeper to easily search for an item using its code
Precondition(s)	<ul style="list-style-type: none"> - Installation of the Application on a mobile or tablet. - The storekeeper connects to the application.
Post condition(s)	Displaying Item Information
Scenario	<ol style="list-style-type: none"> 1. User clicks on Search Item button 2. The system displays a form 3. User enters warehouse code 4. The user enters the item code 5. The user clicks on the Search button 6. The system displays the item information
Exceptions	5a- User on Cancel button 6a- The system displays "Item not found"

Table 3: Item Search Scenario

Native Mobile App

- Details of the "Print Inventory" use case:

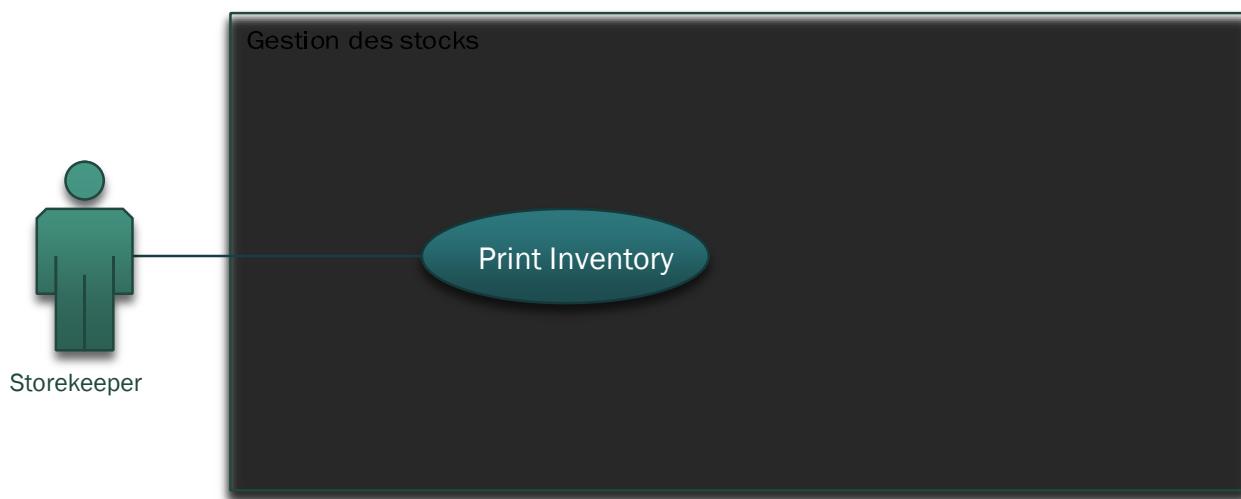


Figure 5: Repository Printing use case diagram

Case number	04
Case name	Print inventory
Actor	The storekeeper
Description	Allows the shopper to print an item sheet
Precondition(s)	<ul style="list-style-type: none">- Installation of the Application on a mobile or tablet.- The storekeeper connects to the application.
Post condition(s)	The "Table" interface is displayed
Scenario	<ol style="list-style-type: none">1. The user clicked on the Print inventory button2. The system displays a list of warehouses3. The user chooses a warehouse4. The system displays a table of items and its information
Exceptions	4a- The system displays an empty table

Table 4: Item Printing Scenario

- Details of the "Initialization of the Database" use case :

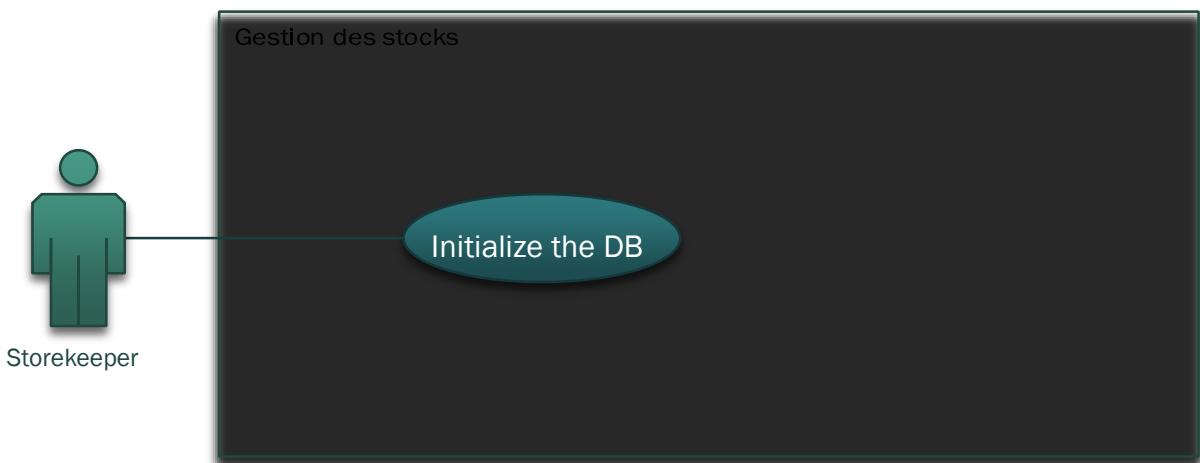


Figure 6: Database Initialization use case diagram

Case number	05
Case name	Initialize the local database
Actor	The storekeeper
Description	Allows the storekeeper to empty the local database.
Precondition(s)	<ul style="list-style-type: none"> - Installation of the Application on a mobile or tablet. - The storekeeper connects to the application.
Post condition(s)	<ul style="list-style-type: none"> - "Empty base" message - Back to the "Inventory Menu" interface
Scenario	<ol style="list-style-type: none"> 1. The user clicks on the button Initialize the local database 2. The system displays a Local database initialization message box 3. User clicks on confirm button 4. The system empties the local base 5. Return to home page
Exceptions	4a- User clicks on cancel button

Table 5: Base Initialization Scenario

4. Conclusion :

During the exploration phase, we highlighted the requirements of the actors in the system. These requirements have been translated through the modeling of UML use cases. These models allowed us to clearly define the functional needs and define the user scenarios. The identified use cases will guide us later in the conception and implementation of our project.

Chapter 4: Conception

1. Introduction :

Following the specification and needs analysis phase, we will invoke in this part the conception of our application. This phase helps us to understand our system, clarify it and eliminate ambiguities by specifying its structure and behavior.

To do this, we will start with the conception of our system which was carried out using UML (Unified Modeling Language) which is an object-oriented modeling method used to design software. We will propose a detailed conception composed of the different diagrams of classes, sequences and activity.

2. Sequence diagrams:

Sequence diagrams can be used to describe use cases. They make it possible to model the dynamic aspect of the system and present a temporal dimension which emphasizes the chronology of the sending of messages between objects.

For our modeling, we will present the different sequence diagrams by use case.

We have chosen to model the sequence diagrams of the “Authentication”, “Inventory Consultation”, “Item Search”, “Item Print” and “Initialize the Database” use cases.

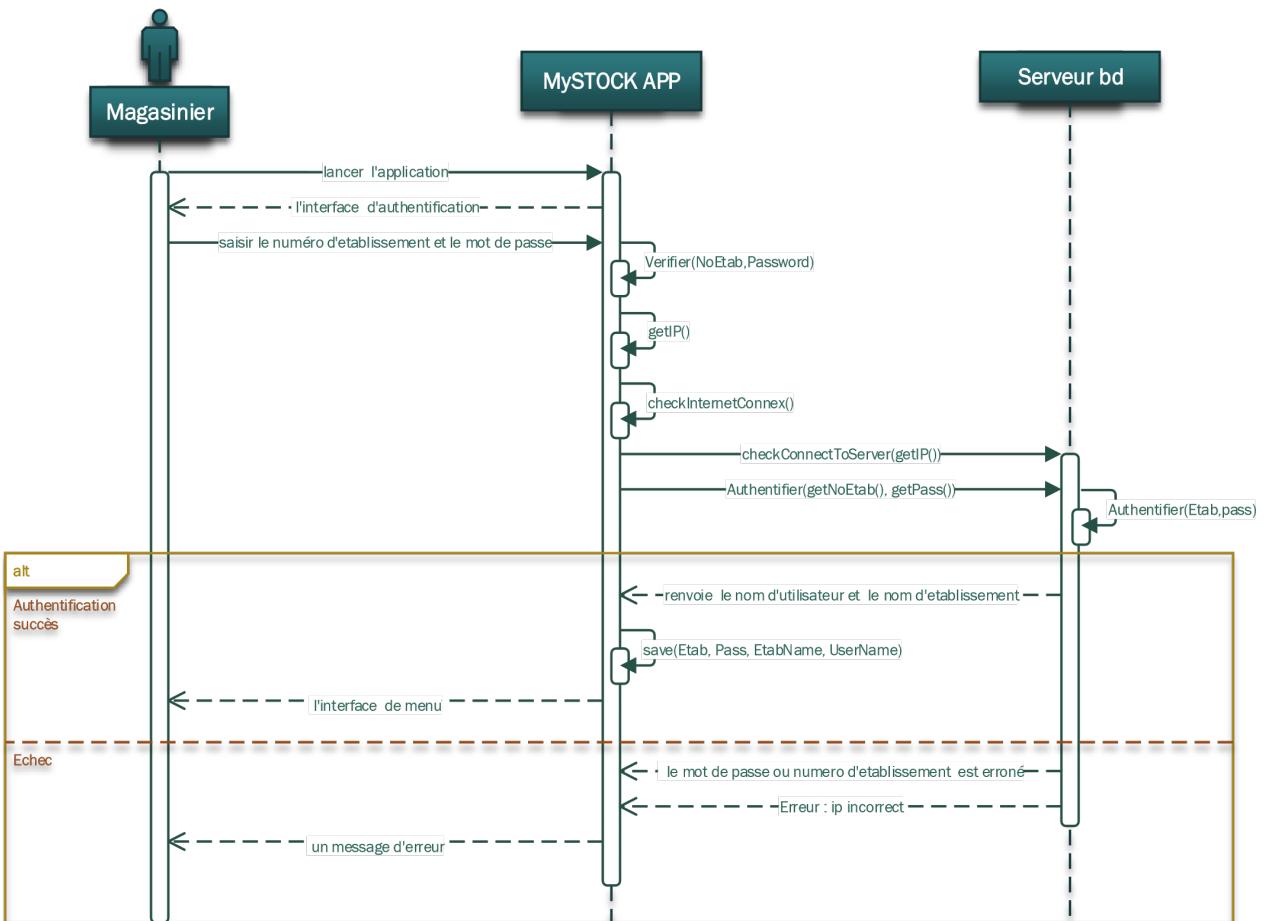


Figure 7: Authentication sequence diagram

When the user requests access to the application, he must first identify himself by his login and password via the application server which is responsible for verifying and consulting the database. If accepted, then it will have access to the system and applications from the corresponding menu.

Otherwise, the application server displays an error message.

Native Mobile App

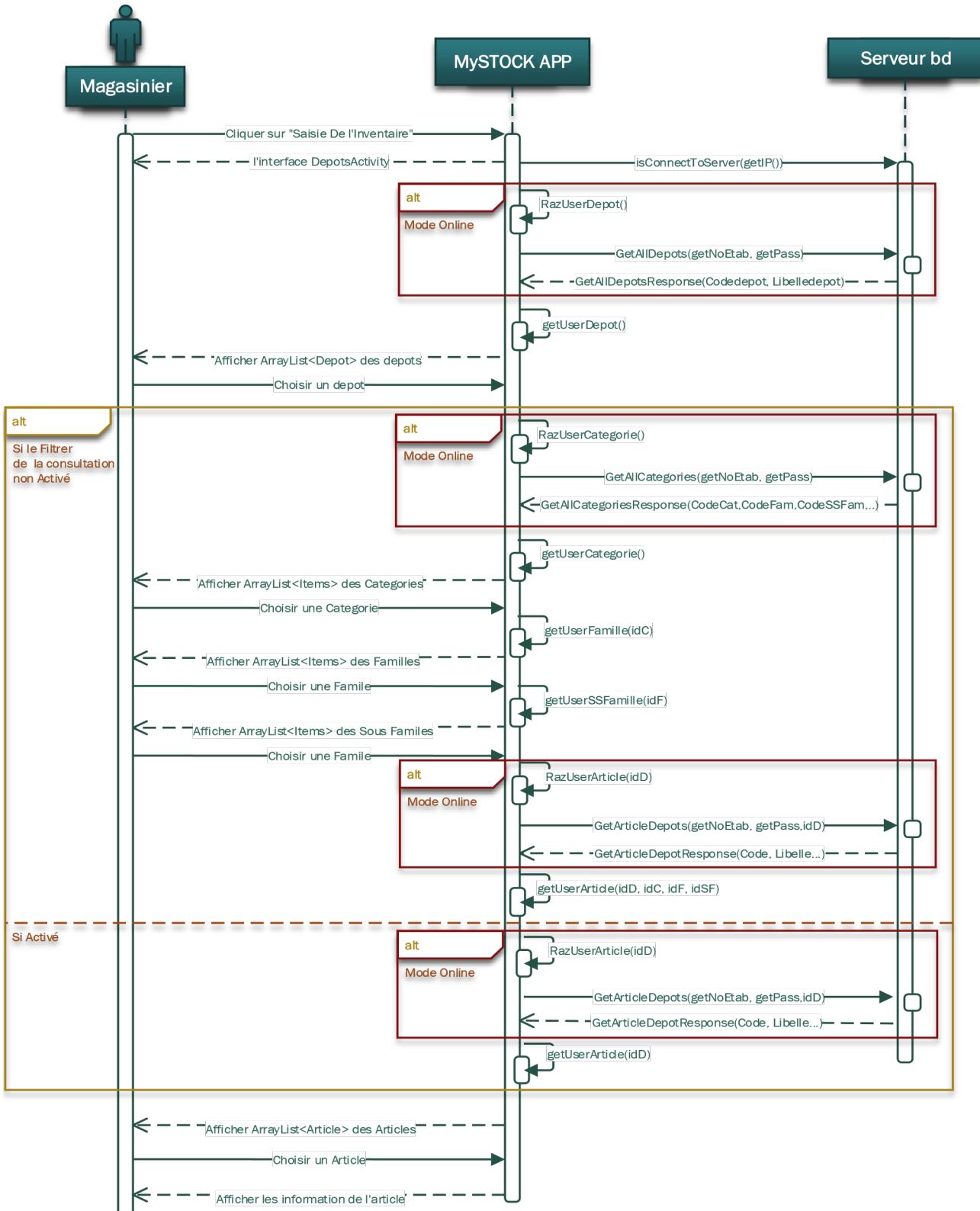


Figure 8: Consult inventory Sequence Diagram

Native Mobile App

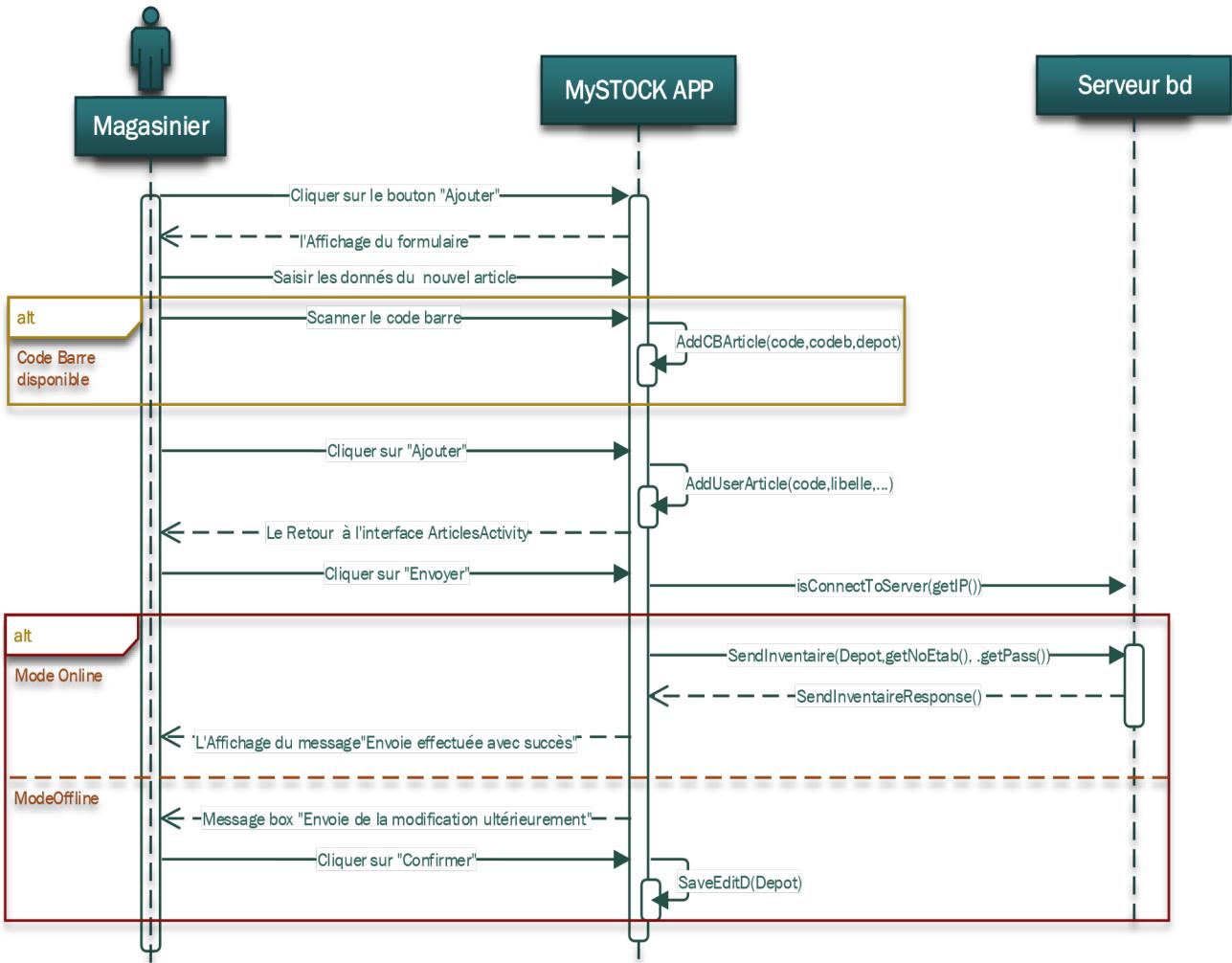


Figure 9: Adding Item Sequence Diagram

Native Mobile App

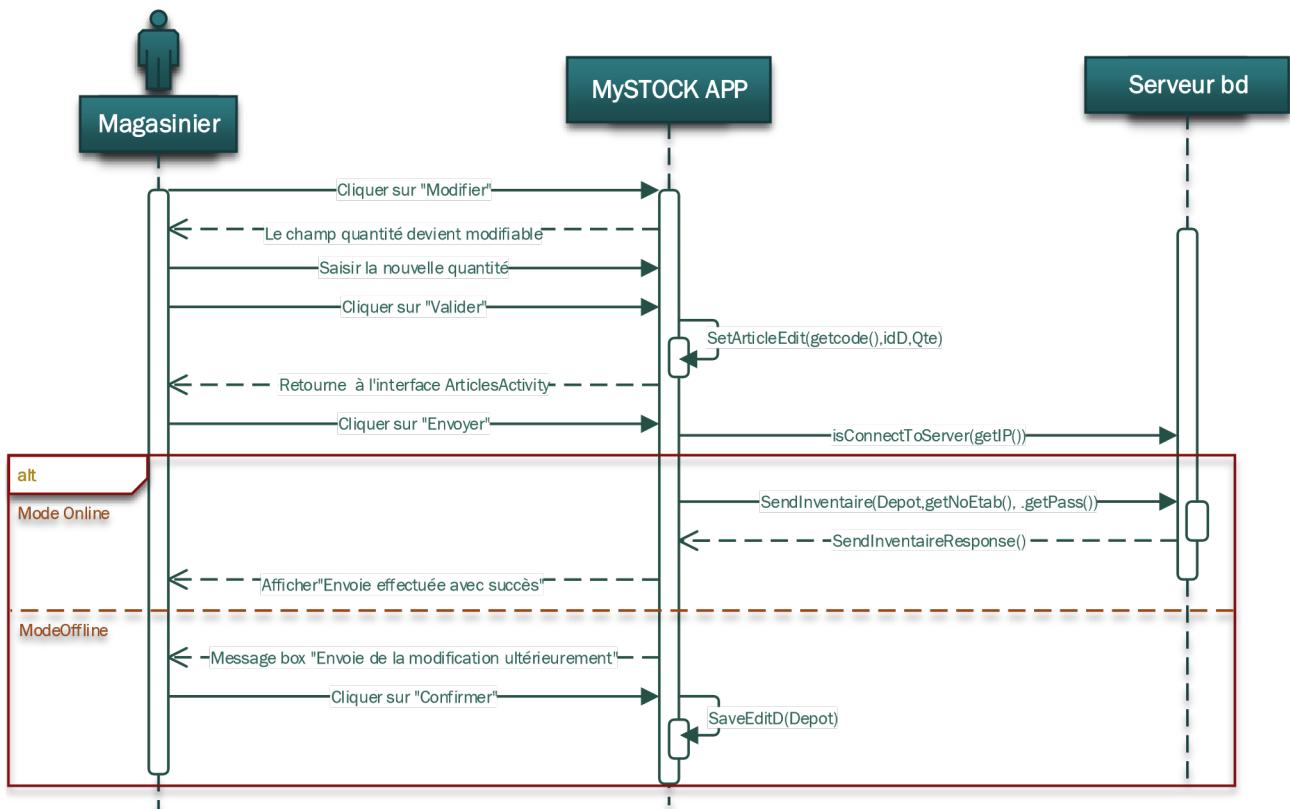


Figure 10: Item Modification Sequence Diagram

To display the data concerning the Hotel Items, the Storekeeper must first open the Inventory Consultation interface, then choose the Filtering option (select "YES" to access the Items directly from the deposit, select "NO" to access the articles from the categories).

If the Filtering option is set to YES then to display the Items, he must select the repository where the Item is located. Thus, the application queries the "ServerIteration" web service which is responsible for retrieving the data from the database which will then be displayed to the user in the form of a list of Items, then he must select the Item he wish to consult.

Otherwise To display the Items, he must select the warehouse where the Item is located. Thus, the application queries the "ServerIteration" web service, which is responsible for retrieving the data from the database, which will then be displayed to the user in the form of a list, the latter retrieving the Categories, Families and Sub-Families. After the Storekeeper choose the Category then the Family and the Sub-Family of the Items, and again the application queries the web service

"ServerIteration" to retrieve the items which will then be displayed to the user in the form of a list, then he must select the item which he wishes to consult.

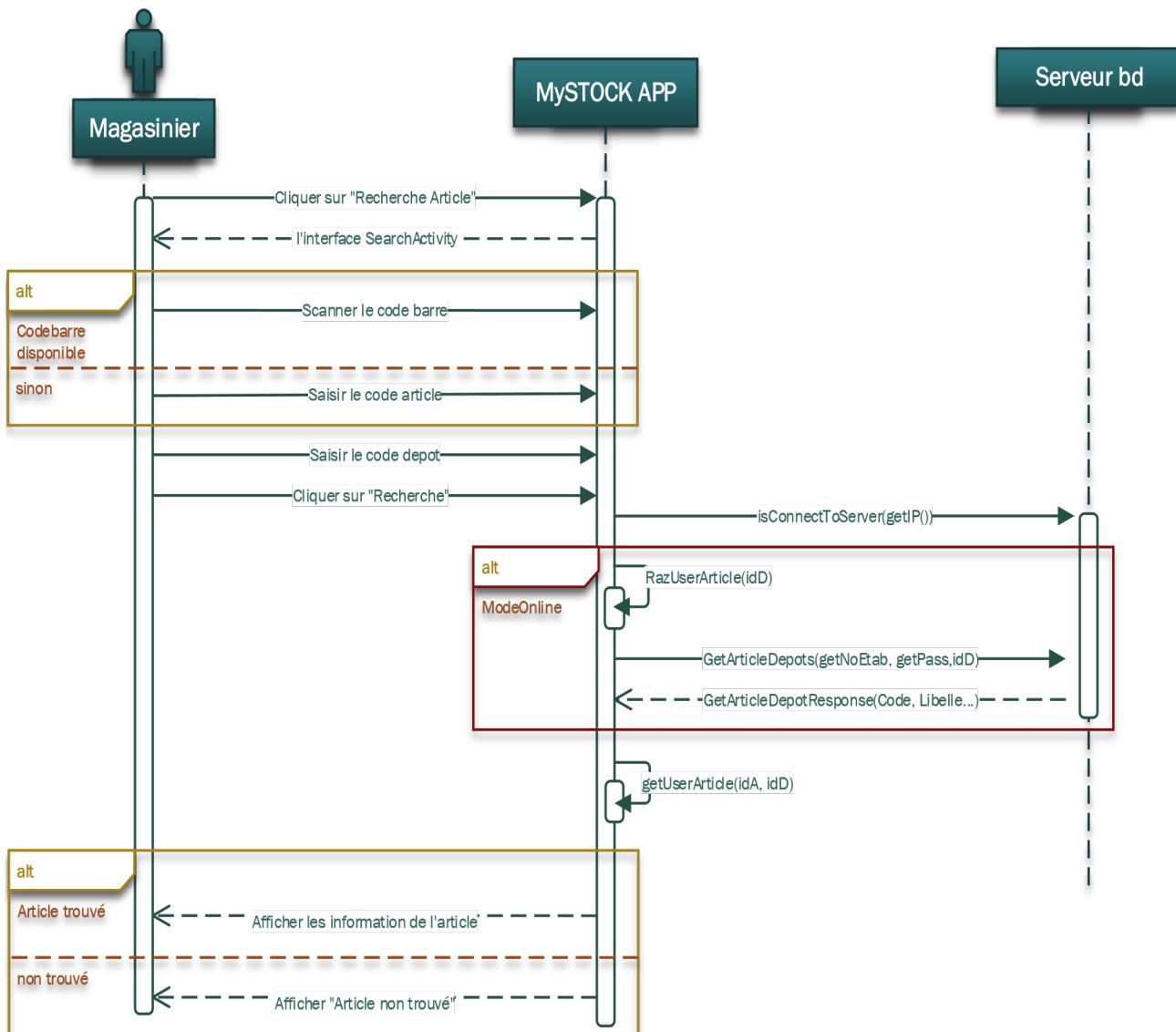


Figure 11: Article Search Sequence Diagram

To display the data concerning the Hotel Items, the Storekeeper must first open the Item Search interface, then choose the Search option.

If the Storekeeper chooses the Search option with the Item Code, then he must enter the Item Code and the Warehouse Code.

Otherwise if he chooses the Search option with the Item Barcode, then he must scan the Item in order to retrieve the Barcode and enter the warehouse code.

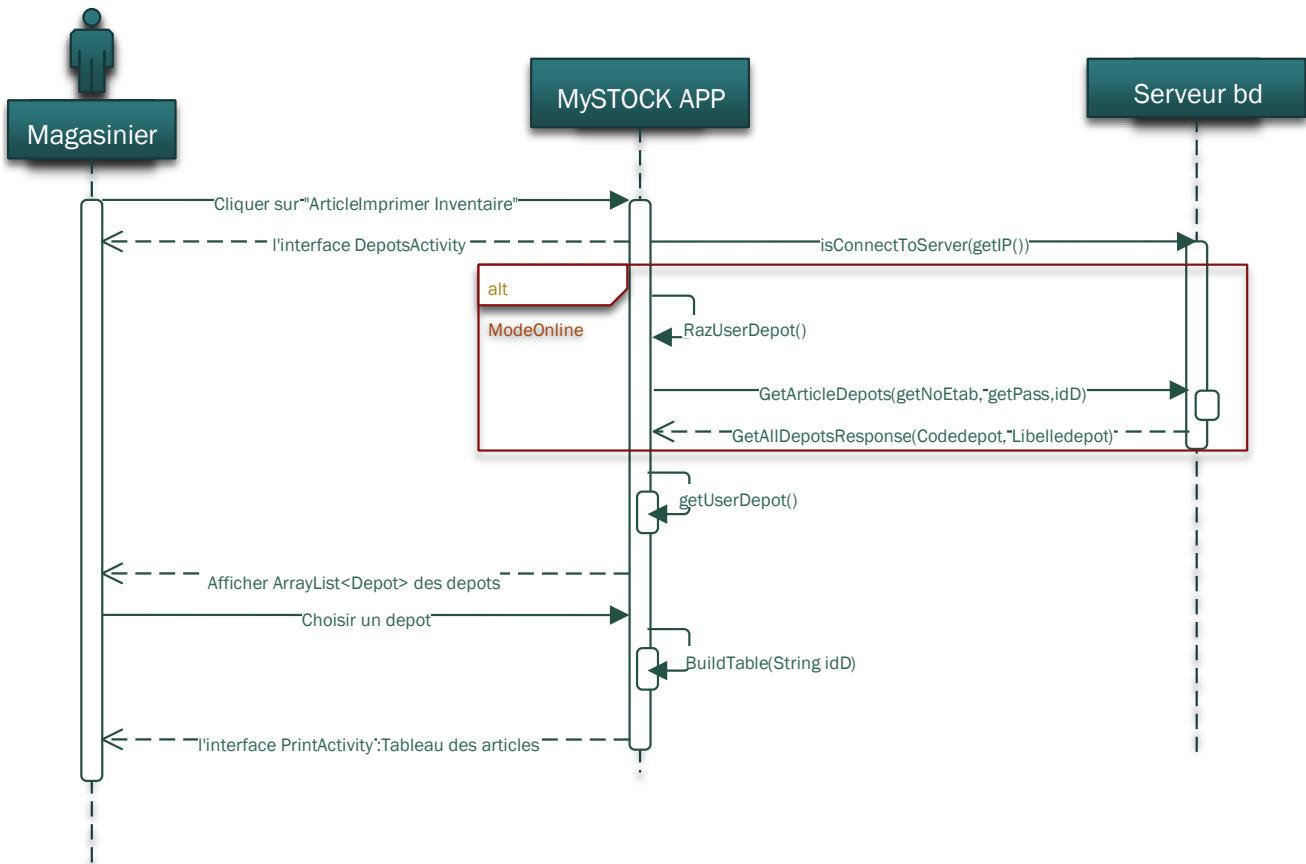


Figure 12: Printing warehouse sequence diagram

To Print the Items, the user must click on the Print Inventory Button, then he must select a warehouse.

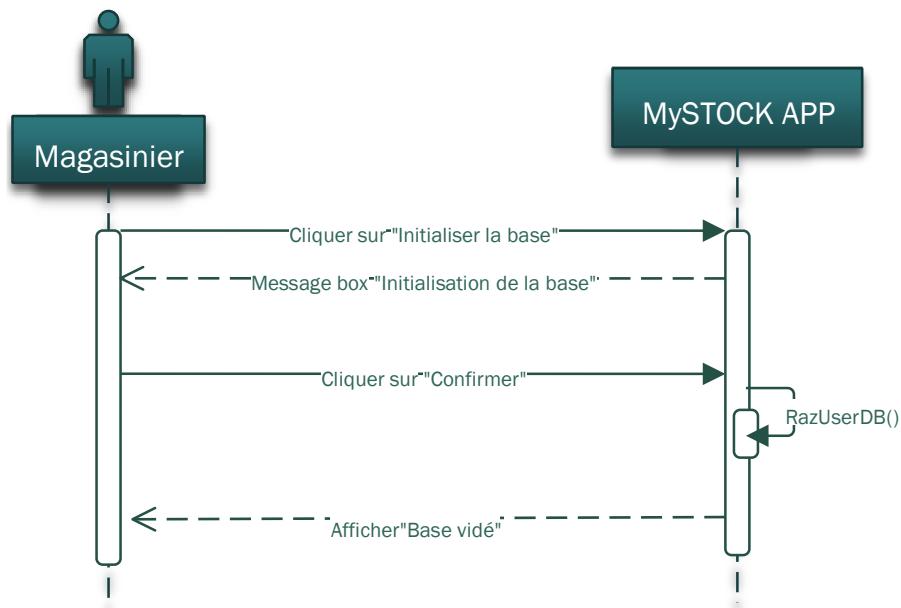


Figure 13: Database Initialization Sequence Diagram

To empty the local Database, the Storekeeper must click on the Database Initialization Button.

3. Activity diagrams:

The activity diagrams allow you to focus on the treatments. They are therefore particularly suitable for modeling the flow of control flows and data flows. They thus make it possible to graphically represent the behavior of a method or the course of a use case.

In the following, we present the activity diagrams for some use cases in our system.

To access to our application, the user must authenticate by entering his login and password. The authentication process can be summarized in the following activity diagram:

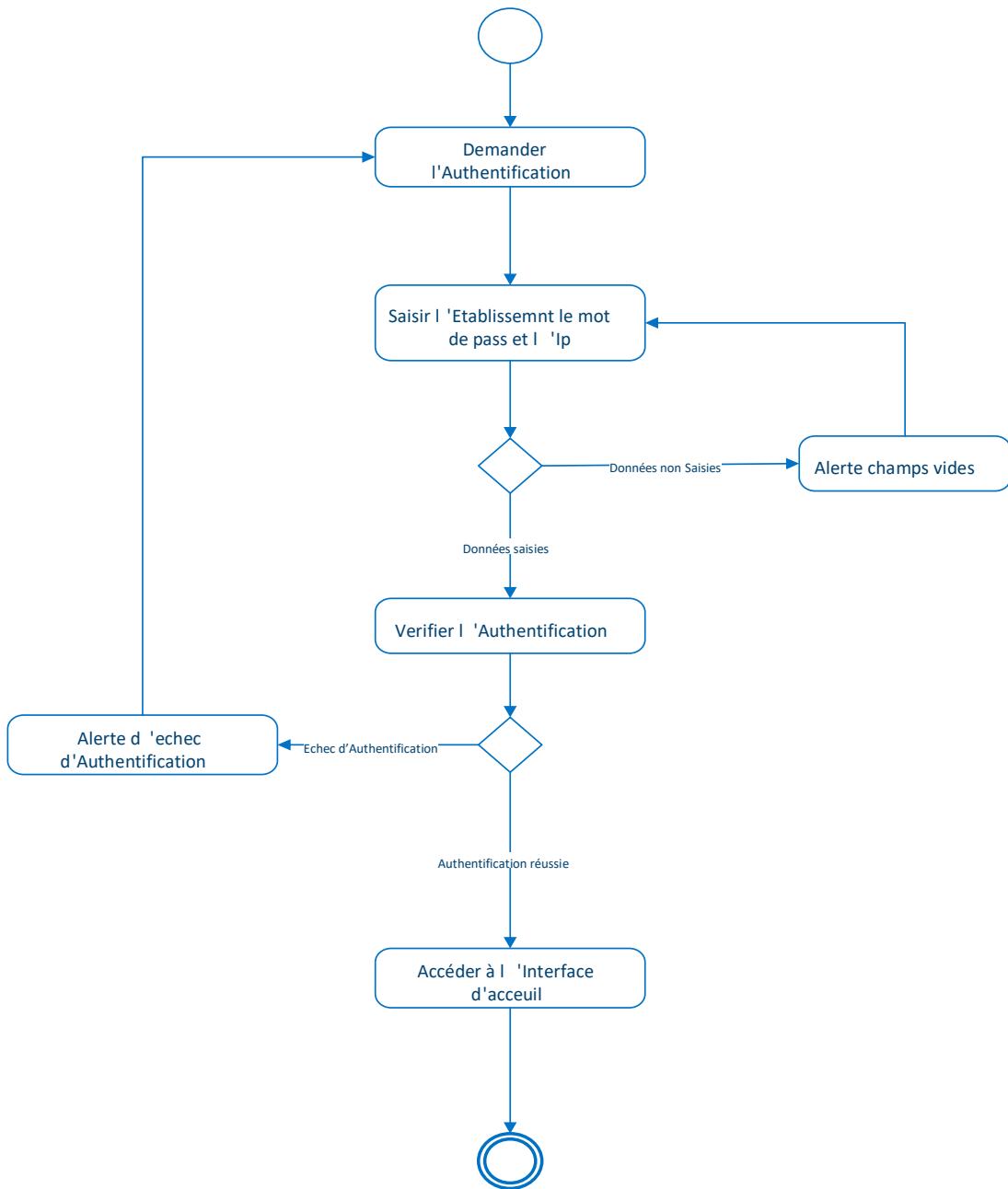


Figure 14: Authentication activity diagram

After being authenticated, the user can access the home interface which corresponds to him and which allows him to choose one of the functionalities offered by the application.

- We start with the activity diagram of the "Consult the Inventory" use case. This use case begins with the choice of the Filtering option.

Native Mobile App

If the Filtering value is YES then we must select the warehouse which will lead us to a list of Items in order to choose the Desired Item which results in the display of the data in the form of a table.

Otherwise if the Filtering value is NO then you have to select the deposit, the Category, the Family, and the Subfamily or the latter will lead us to a list of Items in order to choose the Desired Item which leads to the display of data in tabular form.

The figure expands the activity diagram of the "Consult Inventory" use case.

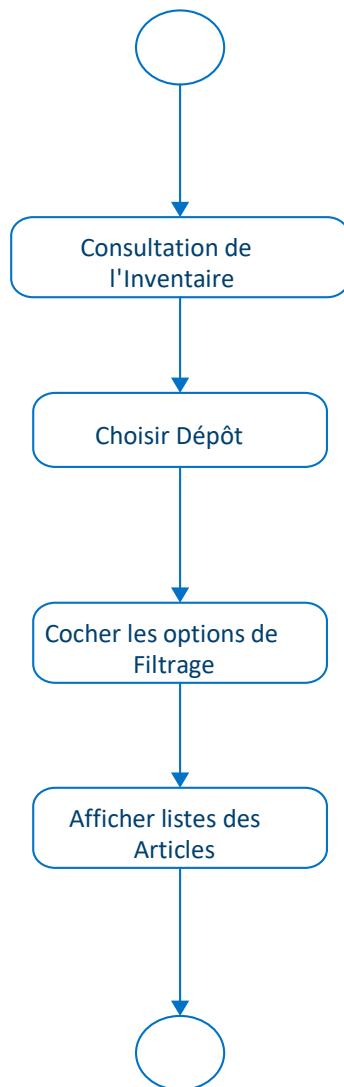


Figure 15: Inventory Consultation Activity Diagram

Native Mobile App

The following figure shows the activity diagram for the "Item Search" use case. The Storekeeper must enter the Item Code, the Warehouse Code. The process ends with the display of the item in the form of a data table.

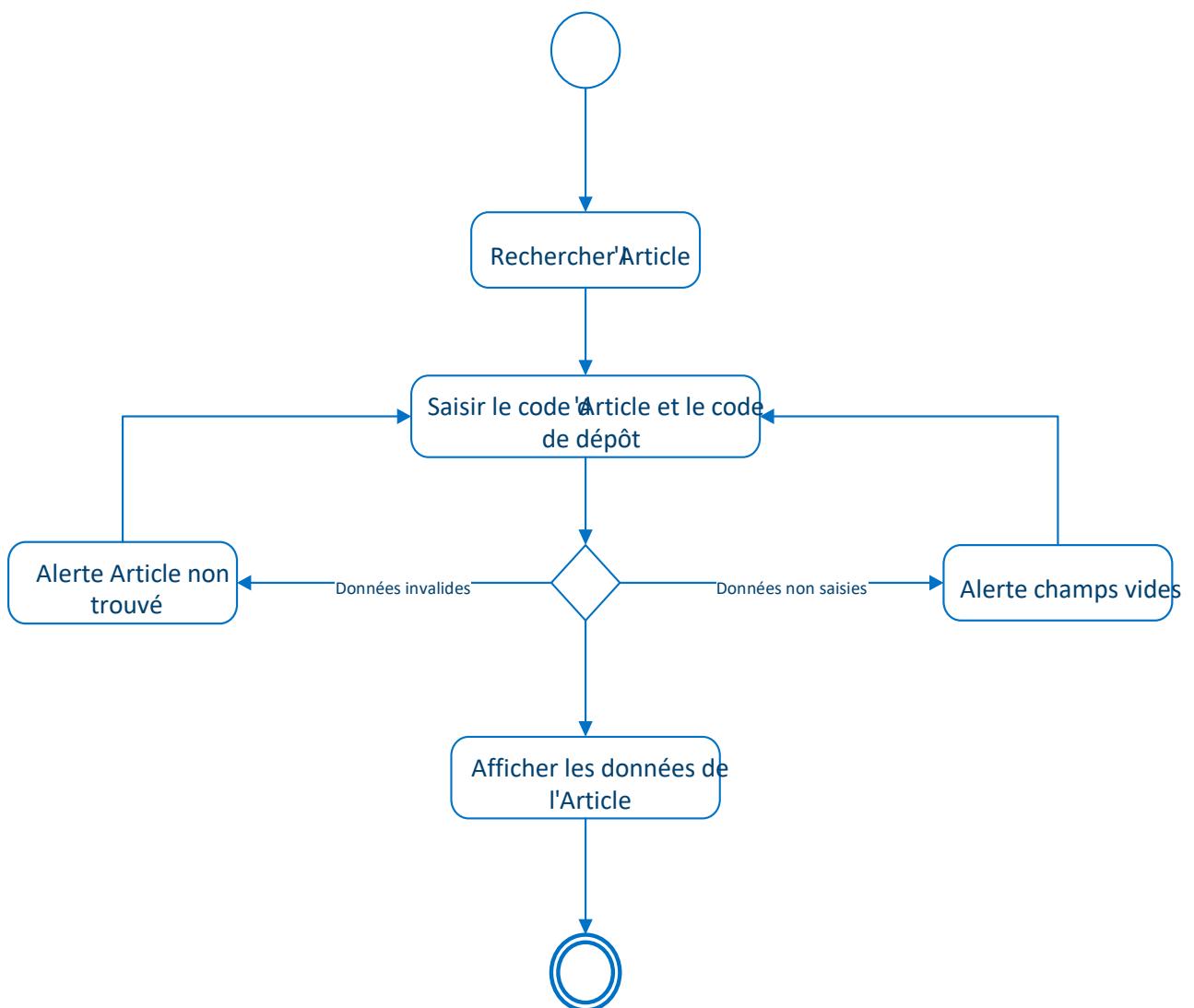


Figure 16: Item Search Activity Diagram

Native Mobile App

The following figure shows the activity diagram for the "Print Inventory" use case. The Storekeeper must choose a Warehouse, which will lead to a page which displays the Items of the Warehouse.

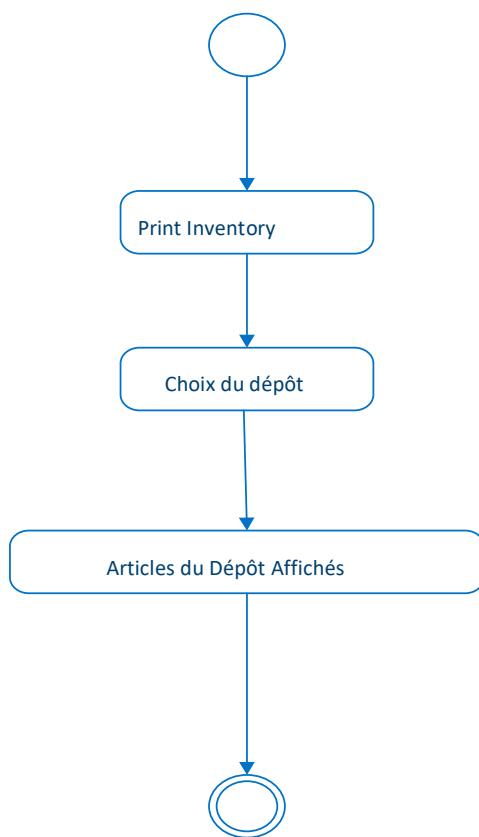


Figure 17: Inventory Printing Activity Diagram

4. Class diagram:

A class diagram gives a static view of the system. It is considered to be the most important of object-oriented modeling. It mainly contains classes. A class contains attributes and operations. It allows you to define what the components of the final system will be and to structure development work very effectively.

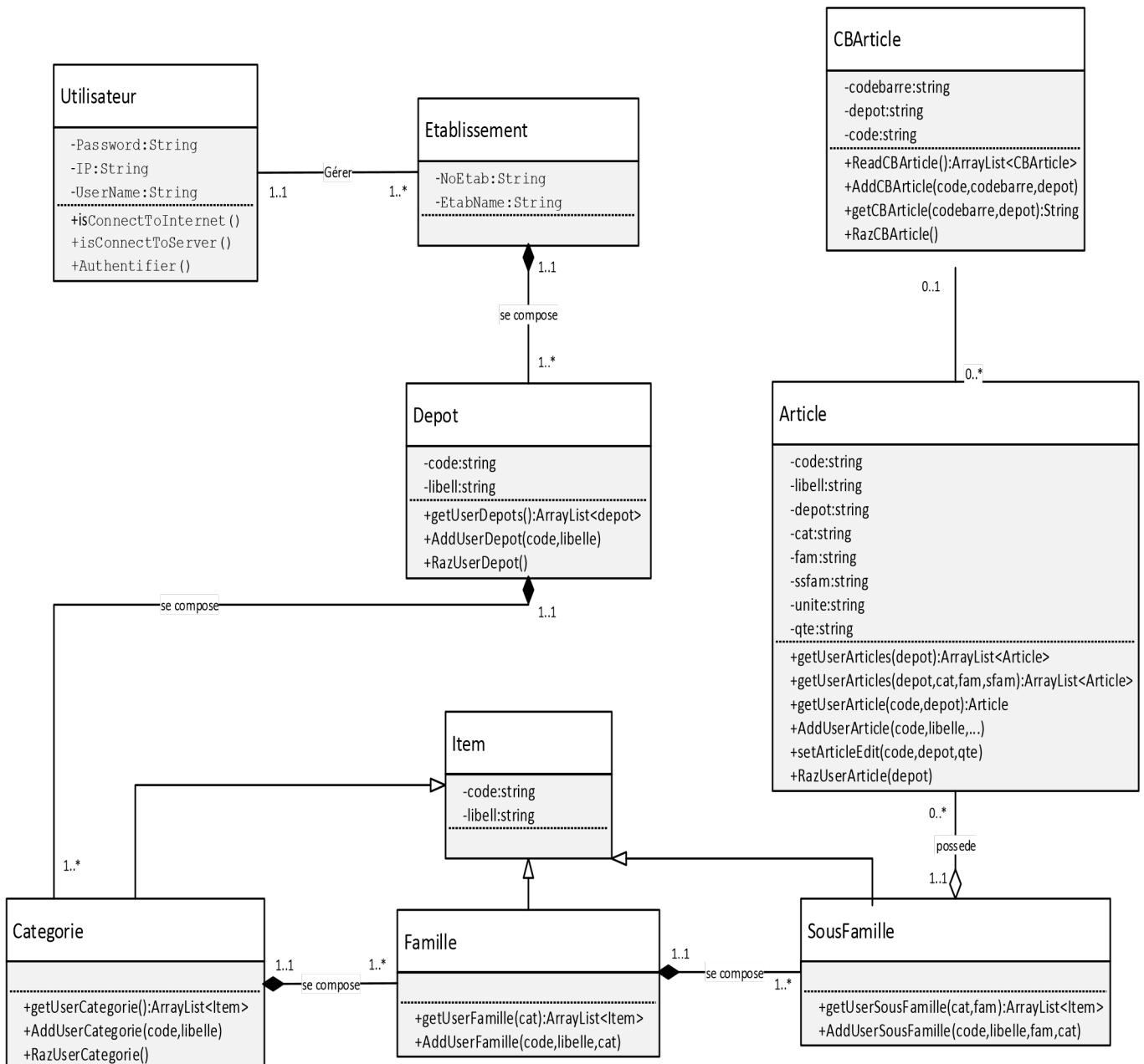


Figure 18: Class diagram

5. Relational model:

The relational model The MLD or relational model consists in describing the data structure used. It takes into account the IT constraints of data structuring. It is therefore a question of specifying the type of data used during processing (text, integer, real, etc.). Thus, the logic model is dependent on the DBMS used, which in our case is MS Access. This model also integrates the data constraints which are directly defined in the parameters of the database tables: type of attribute data (text, numeric or Boolean), data limit values and units of measure for quantitative data, format.

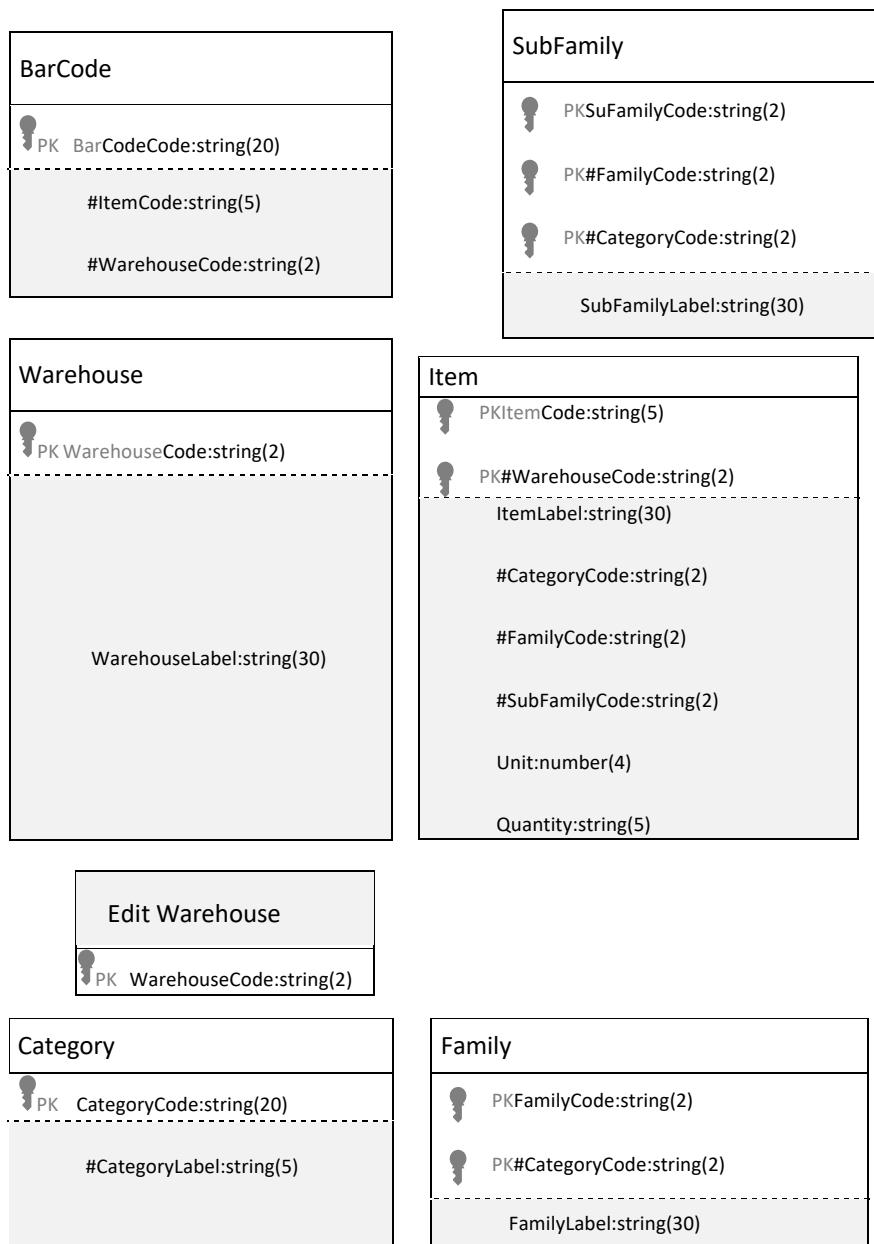


Figure 19: MLD

6. Conclusion :

In this chapter, we have highlighted an extended conception of the main use cases of the **MyStock** application, during which we developed the class diagrams, sequence diagrams and activity diagrams of our system. as well as the design of the database used.

In the next chapter, we will detail the development environment as well as the tools that led us to the realization of our project.

Chapter 5: Production

1. Introduction :

This chapter makes it possible to achieve the objectives and the solutions requested. We start with a presentation of the work environment on the hardware and software levels. Next, we discuss the practices adopted during programming. Finally, we detail what we have developed.

2. System architecture :

The architecture of our system is illustrated by the figure ... it is shared between:

- **The ANDROID client :** Application container and resource requester.
- **The web service module :** Since the data will be communicated between two heterogeneous environments, the main role of this module is to manage the communication between the ANDROID client and the INFOR application server.

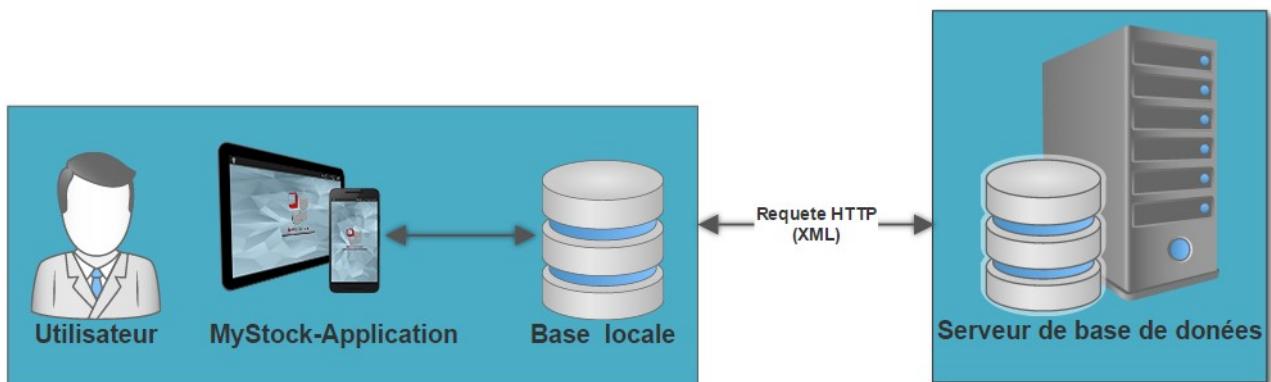


Figure 20: System architecture

Native Mobile App

3. Work environments :

The working environment is essential for the realization of the requested solutions.

3.1 Material environment:

During the project, we used:

- **Ord COMPAQ laptop:** Intel Core i5 / 2.53 GHz processor
- **Operating system:** Microsoft Windows 8.1

3.2 Software environment:

To make our application a reality, we used a set of technical tools. The table above includes these tools:

Tools	Logo	Description
Android Studios		Android studios is a free, extensible, universal, versatile and cross-platform integrated development environment.
Adobe Photoshop CC		Photoshop is a computer-assisted retouching, processing and drawing software published by Adobe. It is mainly used for processing digital photographs.
Microsoft SQL Server		Microsoft SQL Server is a powerful and reliable and comprehensive and reliable database management system for office applications and websites.
SQLite		SQLite is a library written in C which offers a relational database engine accessible by the SQL language.

Native Mobile App

Microsoft Visio 2016 Professionnel		Simplify complex information with professional diagrams that you can create in just a few clicks. Visio simplifies the creation of diagrams, whether it is to quickly capture a flow diagram that you have just drawn up on a whiteboard, to diagram a computer network, to build an organization chart, to document a business process or to draw a floor plan.
---	---	--

Table 6: All the software tools used in the production phase

4. SOAP web service implementation :

Web services are hosted on the local INFOR application server.

This list represents all of the web services that have already been developed and used in the PMS:

- ✓ Authentifier
- ✓ PingRQ

The following figure shows the list of web services that have been added to ensure communication between the Android client and Application Server.

- ✓ GetAllWarehouses
- ✓ GetItemWarehouse
- ✓ GetAllCategories
- ✓ SendInventory
- ✓ SendBCInventory

5. Android side implementation :

5.1 Choice of Android version :

We chose to develop the application with the Android 4.4.2 version known as KitKat since Android terminals that support this version represent more than 32.5% worldwide.

The following figure shows a statistic from the official website of the Android platform on what we mentioned before.

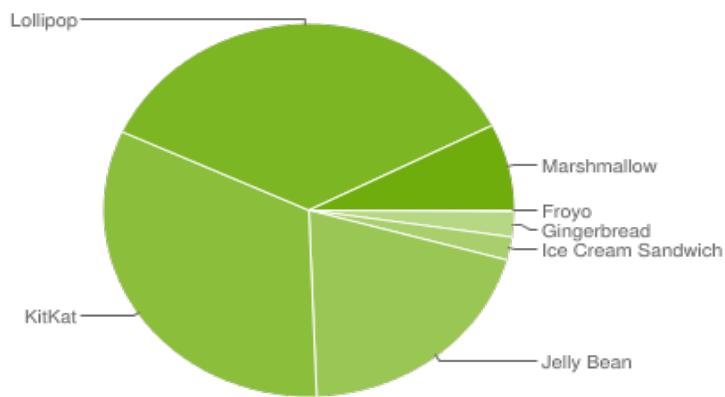


Figure 21: Distribution of Android API versions

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.0%
4.1.x	Jelly Bean	16	7.2%
4.2.x		17	10.0%
4.3		18	2.9%
4.4	KitKat	19	32.5%
5.0	Lollipop	21	16.2%
5.1		22	19.4%
6.0	Marshmallow	23	7.5%

Table 7: Distribution of Android API versions

5.2 Used Libraries :

- **kSOAP2** : is a lightweight library that is used to communicate the Android application with SOAP web services. It is designed to interact with most popular SOAP engines.
- **ZXING** : is an open-source multi-format 1D / 2D barcode image processing project implemented in Java. This project focuses on using the built-in camera on mobile phones and decoding bar codes on the device, without communicating with a server.

6. Application interfaces :

The application has two modes of access:

- ✓ Online mode.
- ✓ Offline mode.

Starting with Online mode first.

6.1 Online mode :

This Mode is used for the purpose of loading or sending data to the server.

The first opening of the application is done in Online Mode

- Authentication interfaces :

The figure shows the interface that allows us to choose the configuration of the application.

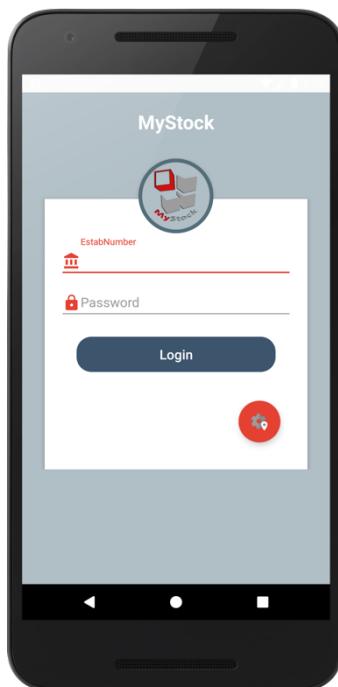


Figure 22: Authentication home interface

Native Mobile App

The following figure (Figure 24) is the authentication interface with which the storekeeper can connect to the system. After pressing the "Connection" button, if the parameters are valid, the system displays the home interface (Figure 23) which shows the different modules which we will detail below. Otherwise, the system displays an authentication failure error message or an empty field error message (Figure 25).

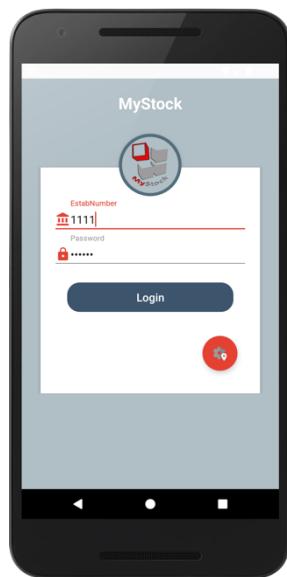


Figure 25: Manager authentication interface

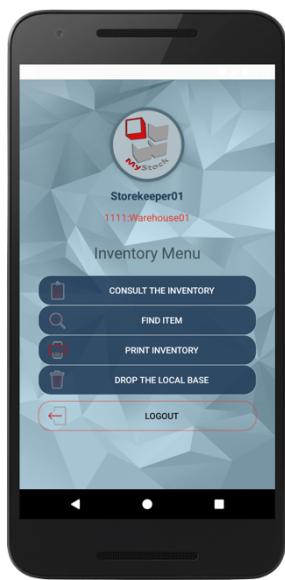


Figure 24: Manager's home interface if the parameters are valid



Figure 23: Error message when authentication fails

Native Mobile App

- Interfaces of modules linked to the storekeeper:

- Inventory consultation :

The following figure shows the interface which gives access to the various Item.

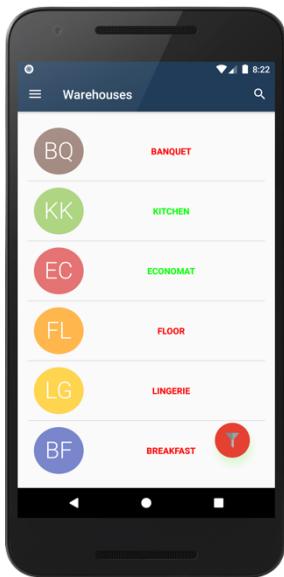


Figure 26: Search type choice interface

The Items consultation process is done with two methods which are based on the Filtering option:

- ✓ If the user selects **NO** (Figure 27) (by default no) in the Filtering option then the process will be as follows:
 1. First, select the warehouse where the Item is located (Figure 28).
 2. The selected warehouse will display a list of Categories (Figure 29).
 3. Now the user has to select the Category, which will display the following families (Figure 30).
 4. After that it will be the turn to select the Family, in order to give us the following result (Figure 31).
 5. At the end the selection of the Sub Family will display the Items (Figure 32).

Native Mobile App

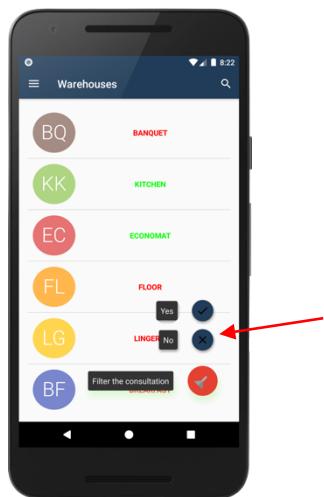


Figure 27: Items consultation interface with the NO filtering option

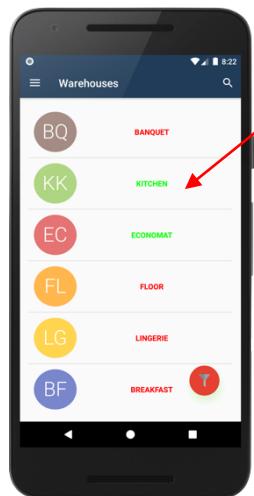


Figure 28: Interface to select the warehouse Where the item is located

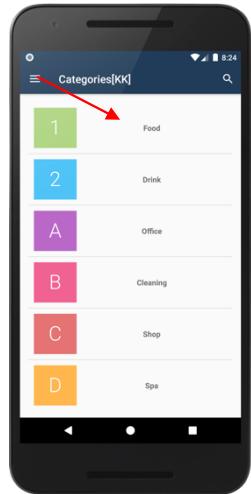


Figure 29: Category list interface

Native Mobile App

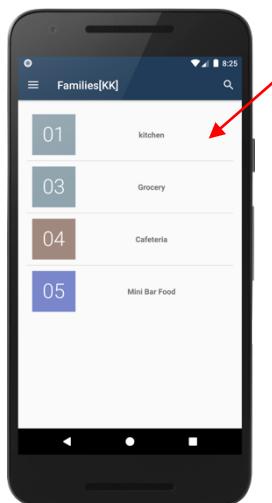


Figure 30: List of Families interface

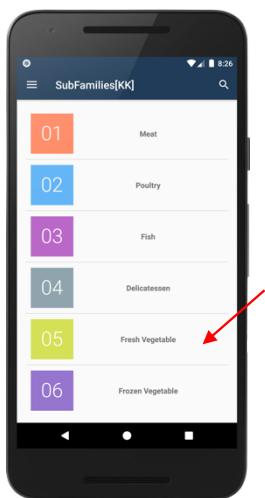


Figure 31: List of Subfamilies interface

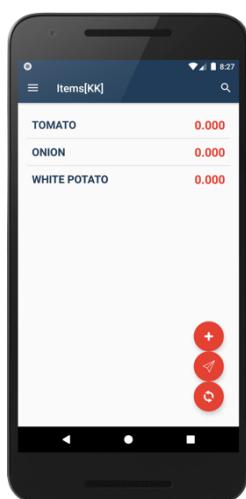


Figure 32: Items list interface

Native Mobile App

- ✓ If the user selects **YES** (Figure 33) in the Filtering option then the process will be as follows:
 1. The selected repository will display a list of Articles (Figure 34).

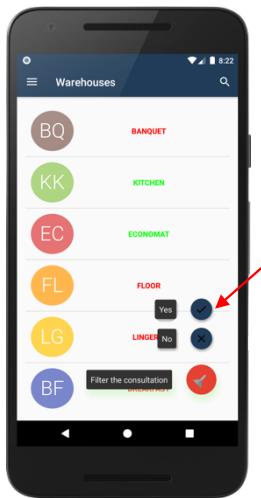


Figure 33: Items consultation interface with the Filtering option YES

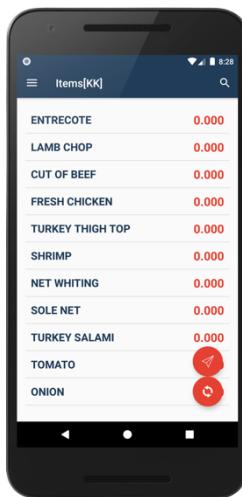


Figure 34: Items list interface

➤ *Editing Items :*

- ✓ The Item Modification process comes after consulting the Items:
 1. The consultation can be done with the two Filtering options, here we used the YES option of Filtering.
 2. When the user selects an item from the previous list, the following interface will display the item info (Figure 35). To change the quantity, the user must first click on the Edit button.
 3. Then he must enter the new quantity (Figure 36).
 4. To validate the modification, click on the Submit button (Figure 37).
 5. After confirming the modification, the application will return to the previous interface, where we will distinguish that the quantity has changed (Figure 38).

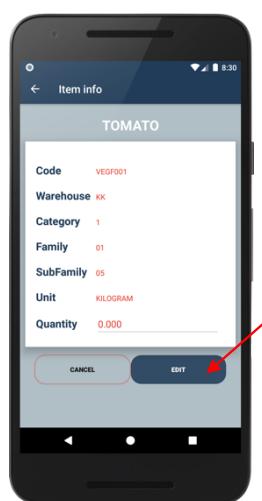


Figure 36: Interface item info

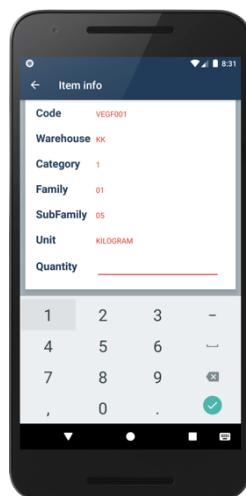


Figure 35: Interface item info in edit mode

Native Mobile App

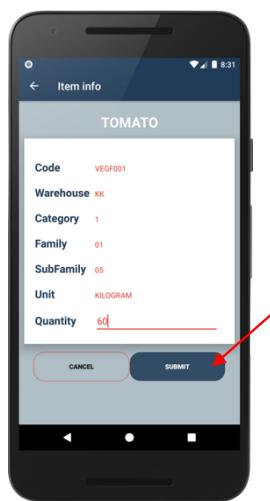


Figure 37: Interface item info in edit mode

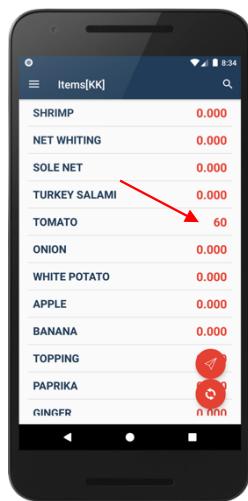


Figure 38: List of items interface

Native Mobile App

- ✓ After the modification, the user can send this modification to the server:
 1. To send the modification to the server, the user must click on the send button (Figure 39).
 2. Which will change shape after sending (Figure 40).

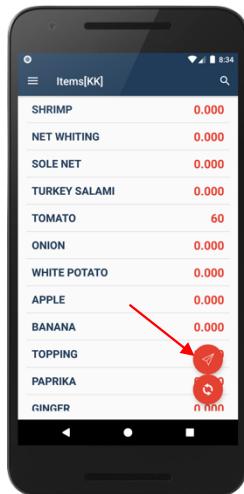


Figure 39: List of items interface

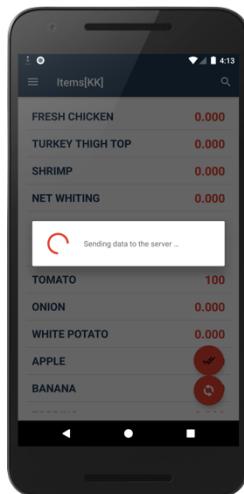


Figure 40: Interface for Editing Items in Online Mode

Native Mobile App

- Adding an Item :
- ✓ The process of Adding Items comes after consulting them:
 1. The consultation must be with the Filtering option **NO**.
 2. To Add an Item, the user must click the Add Button (Figure 41).
 3. When the user presses the Add button, the following interface will display the Item information (Figure 42), Here the Addition can be done with two options: Add an Item by giving his code Or Add an Item by scanning its Barcode.
 4. The user must enter the Code, Label, Unit, and Quantity. At the end, press the Add button (Figure 44).
 5. After validating the addition, the application will return to the previous interface, where we will distinguish that the article has been added (Figure 45).

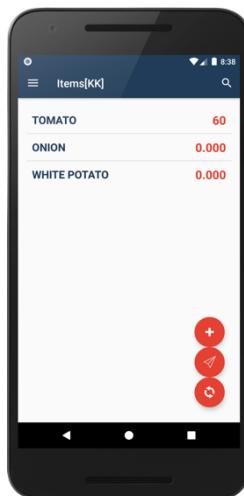


Figure 41: List of items interface

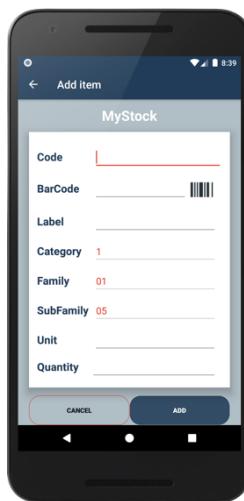


Figure 42: Add item interface

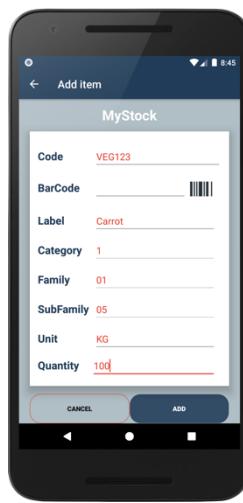


Figure 43: Add item interface

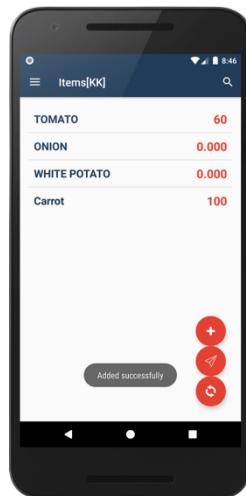


Figure 44: Interface for Adding Items in Online Mode

Note: The sending of the Add request is done in the same way as the Modification.

- Item Search:
 - ✓ The Item search process is also done with two options like in Adding:
 1. To Search for the Item, the user must first scan the Item Barcode (Figure 45) or enter the item code, and enter the warehouse code, then click on the search button (Figure 46).
 2. Hence the display of an item information in the following way (Figure 47).

Native Mobile App



Figure 45: Item Scanner interface

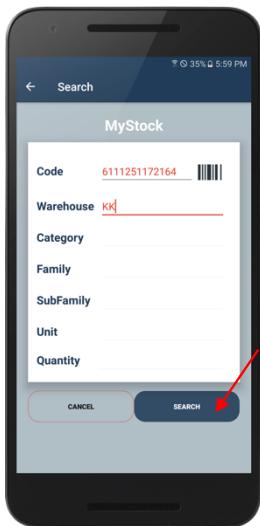


Figure 46: Item Search Interface

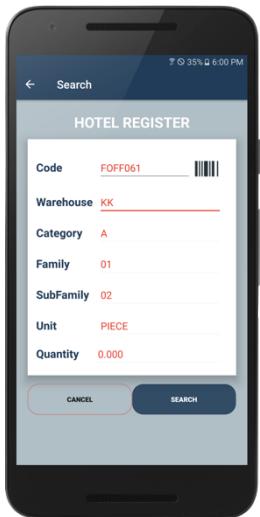


Figure 47: Item Search Interface

Native Mobile App

- Print a Warehouse:
- ✓ The process of printing the warehouse is as follows:
 1. To Print a warehouse, the user must click on the PRINT INVENTORY button on the main menu (Figure 48).
 2. Then choose a Warehouse (Figure 49).
 3. After the following interface will appear (Figure 50).



Figure 48: Inventory Menu Interface

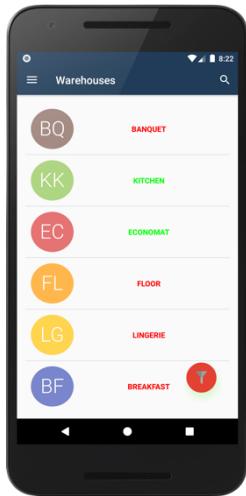


Figure 49: Warehouse list interface

A screenshot of a mobile phone showing a detailed list of items from a specific warehouse. The screen has a header 'Print' and a sub-header 'Warehouse :KK'. Below the sub-header is a table with columns: Code, Label, Category, Family, SubFamily, and Unit. The table contains 14 rows of data, including items like MEAT001 (ENTRECOTE), MEAT002 (LAMB CHOP), MEAT003 (CUT OF BEEF), VOL001 (FRESH CHICKEN), VOL002 (TURKEY THIGH TOP), FISH001 (SHRIMP), FISH008 (NET WHITING), FISH009 (SOLE NET), CHAR001 (TURKEY SALAMI), and VEGCFD01 (TOMATO).

Code	Label	Category	Family	SubFamily	Unit
MEAT001	ENTRECOTE	1	01	01	KILOGRAM
MEAT002	LAMB CHOP	1	01	01	KILOGRAM
MEAT003	CUT OF BEEF	1	01	01	KILOGRAM
VOL001	FRESH CHICKEN	1	01	02	KILOGRAM
VOL002	TURKEY THIGH TOP	1	01	02	KILOGRAM
FISH001	SHRIMP	1	01	03	KILOGRAM
FISH008	NET WHITING	1	01	03	KILOGRAM
FISH009	SOLE NET	1	01	03	KILOGRAM
CHAR001	TURKEY SALAMI	1	01	04	KILOGRAM
VEGCFD01	TOMATO	1	01	05	KILOGRAM

Figure 50: Warehouse Print Interface

Native Mobile App

➤ Logout :

To log out, click on the Logout button (Figure 51).

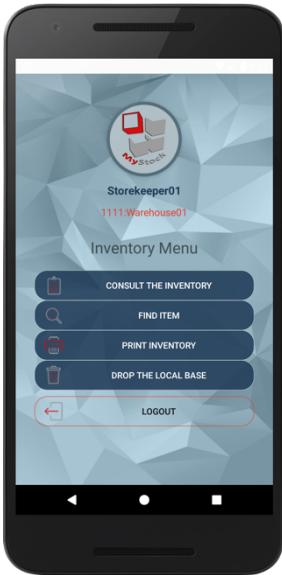


Figure 51: Main Menu Interface

Which will lead to the Authentication interface (Figure 52).

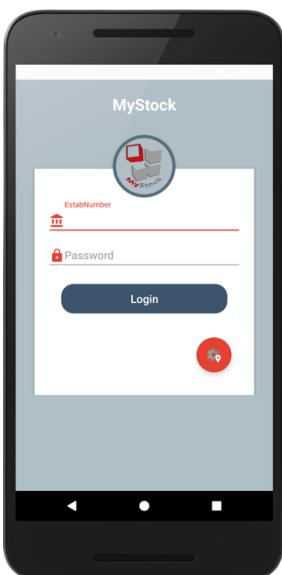


Figure 52: Logout Interface

6.2 Offline mode :

This mode has big advantages compared to Online Mode, it allows a great readability of the application, because the operations on the data is done on the Phone database (SQLITE).

- The Operations carried out in this Mode are the same as those in Online Mode except for two operations which are:

➤ Modifying the Quantity of the Item :

1. We use the same approach as in Online Mode to access the list of items.
2. We then consult the desired item (Fig. 53).
3. To modify the Quantity, the user must click on the Edit button (Figure 54).
4. Then he must enter the new quantity, then validate it by clicking on the submit button (Figure 55).
5. After confirming the modification to the server, the user must click on the Send button (Figure 56).
6. An alert will be displayed, the user must confirm the shipment when the phone is connected to the Internet (Figure 57).

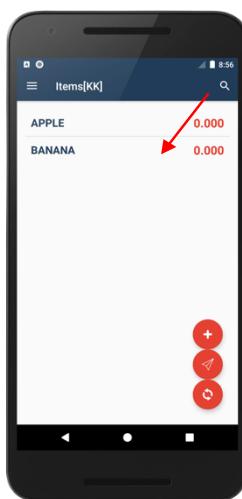


Figure 53: List of items interface

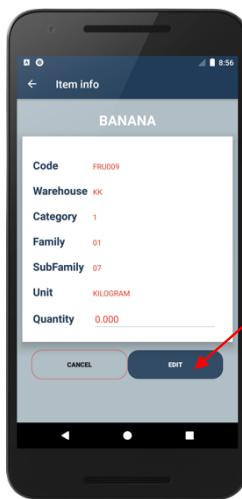


Figure 54: Item info interface

Native Mobile App

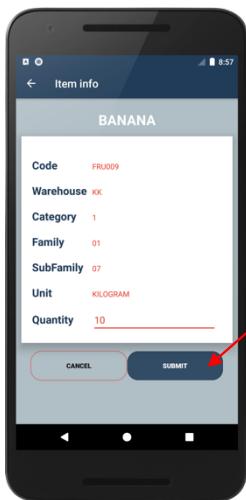


Figure 55: Item info interface



Figure 56: List of items interface

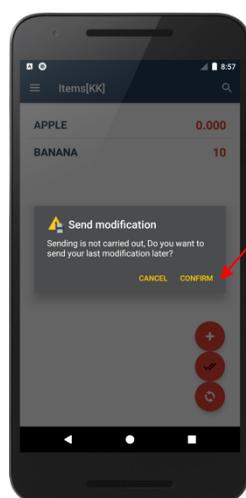


Figure 57: Alert Sends change

Native Mobile App

- ✓ So, When the phone detects the Internet connection.
 1. The Application displays the confirmation message (Figure 58).
 2. The user must click on confirm to send the modification to the server (Figure 59).

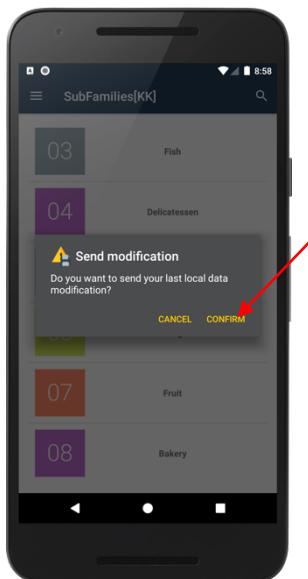


Figure 58: Alert Sends change

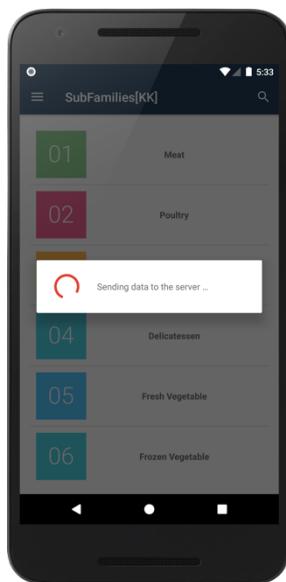


Figure 59: Interface for Editing Items in Offline Mode

➤ Adding the Item:

The process of Adding the Item is as follows:

1. You must first consult the Items to choose the corresponding Item (Figure 60).
2. Then the user must click on Add after entering the item info (Figure 61).
3. To send the Addition to the server, the user must first click on the Send button (Figure 62), which will display a dialog, or this will ask to confirm the sending when the phone is connected to the internet (Figure 63).
4. So When the application detects the presence of the Internet connection, it displays a confirmation alert (Figure 64), The user must click on confirm to send the modification to the server (Figure 65).

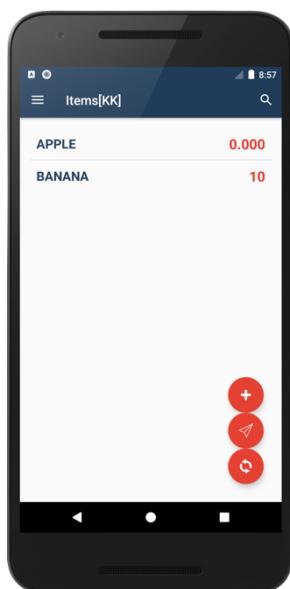


Figure 60: List of items interface

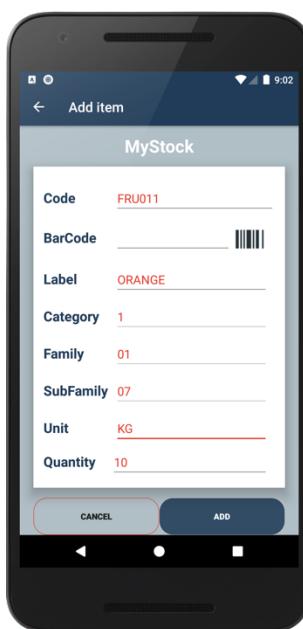


Figure 61: Interface Adding the item

Native Mobile App

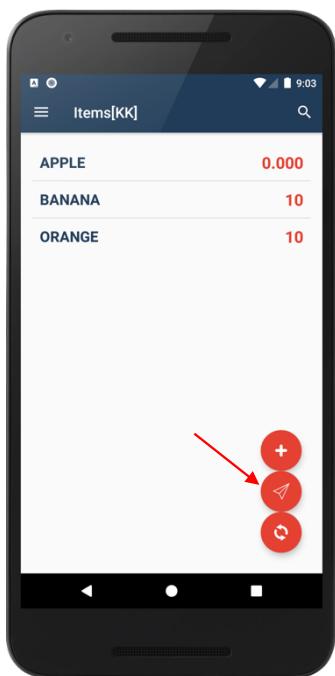


Figure 62: Item list interface

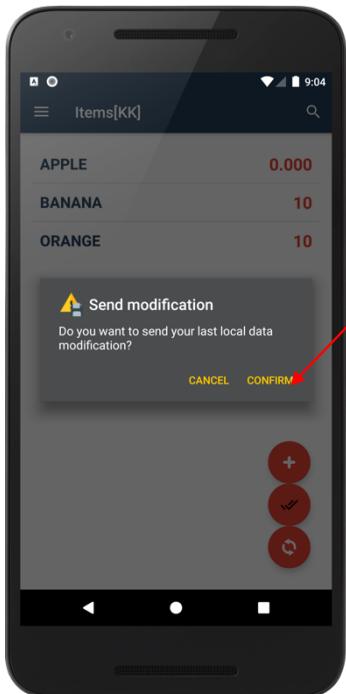


Figure 63: Alert Sends change

Native Mobile App

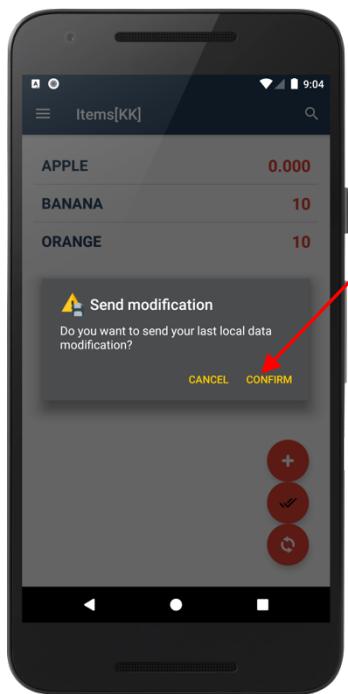


Figure 64: Alert Sends change

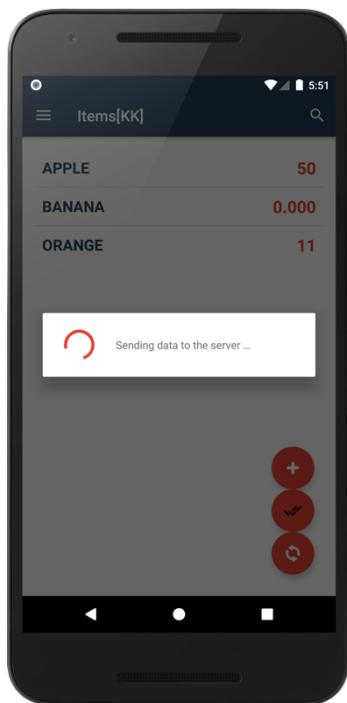


Figure 65: Interface sends update to server

7. Adding Options:

7.1 The search option in the Menu Bar:

- ✓ It facilitates the search by writing the name in the Bar.

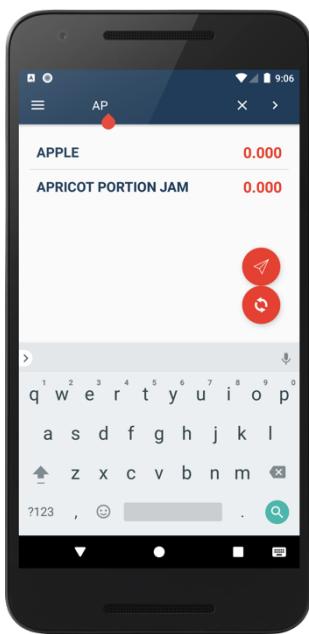


Figure 66: Optimized search option interface

7.2 The Refresh options

- ✓ This option refreshes the local database:
 1. To refresh the local database, the user must click on the refresh button (Figure 67), which will update the data (Figure 68).

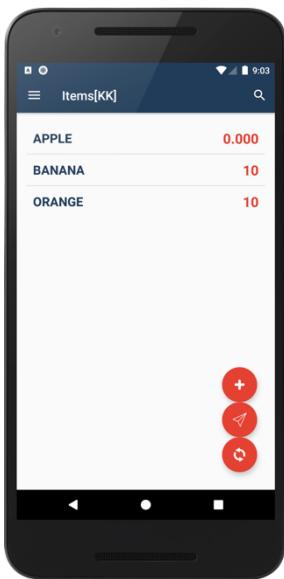


Figure 67: List of items interface

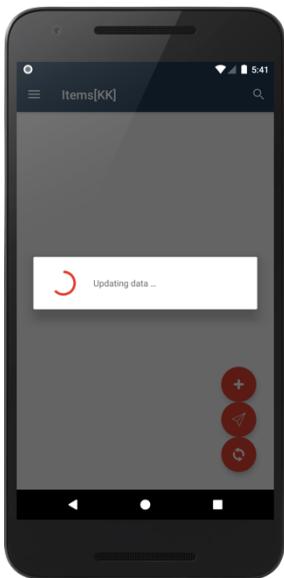


Figure 68: Data update interface

Native Mobile App

7.3 The quick access menu

- ✓ This menu allows you to access tasks more quickly.

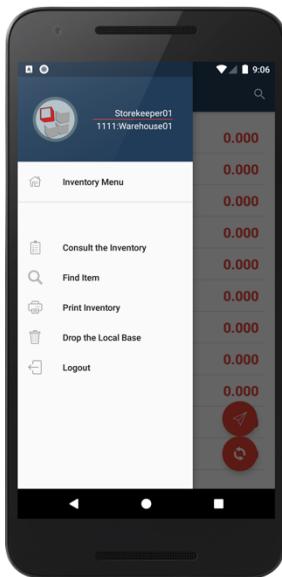


Figure 69: Quick Access Option Interface

7.4 Color change option

- ✓ This option is included only in the warehouse display interface, it indicates the full available warehouses data on offline mode.
 1. The green color indicates that the warehouse data is available on offline mode.
 2. The red indicates that the warehouse data is not available on offline mode.

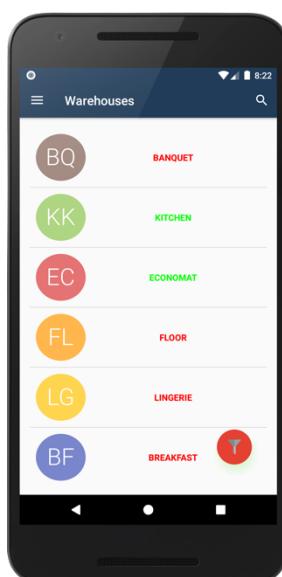


Figure 70: List of warehouses interface with color option

Conclusion and perspectives

Throughout this report, we have tried to present the different steps that we have taken to achieve our Android application.

We started this report with a study of the project context and an analysis of functional and non-functional needs while developing use case diagrams for the system, as well as its different parts. Once the conception is developed, we moved on to the realization part while presenting the different stages of the development of the mobile application.

Certainly, the solution that we have managed to achieve meets the majority of the objectives that we set from the start. However, it is important to point out that certain tasks during this work posed real challenges for us.

To conclude, this project was an undeniable contribution to us on all levels: personal, functional, methodological and particularly on the technical level. In terms of prospects, our application can subsequently be adapted by improvements. It is possible to enrich it with other functionalities and we can also generalize the development of our application on other mobile operating systems.

Bibliography et Webography

[1] Infor: <https://www.infor.com/>

[2] Android : <https://developer.android.com>

[3] SOAP: <http://www.java2s.com/Code/Jar/k/Downloadksoap2android252jar.htm>

[4] BarCode scanner : <https://github.com/dm77/barcodescanner>

[5] Icon Letters: <https://github.com/amulyakhare/TextDrawable>

[6] Message box: <http://developer.android.com/reference/android/app/ProgressDialog.html>

[7] The icons: <https://www.iconfinder.com/>