UNIVERSITÉ HASSAN II DE CASABLANCA

faculté des Sciences
BEN M'SICK

Conception & Production of a mobile Stock Management application

Realized by :

KHOUNA RIDA

KILANIHASSAN

Plan

1	General context
2	
3	Problematic
4	
5	Adopted Solution
6	
7	Needs Analysis and Specification
	Conception
	Production
	Conclusion et perspectives

General Context

Introduction

Problematic

Problems

Coupling

Offline Mode

Evolution

Response time

Adopted Solution

Adopted Solution

Solution

- Save all the Stock on your Smartphone to always have them with you.

- The possibility of performing operations on the stock data without the connection to the database server.

Topic

Development of a mobile Stock Management application

The main objectives:

Allow the User to control several Establishments

Consult the Stock in different ways

Stock Search

Print inventory

The possibility of performing all kinds of operations in offline mode

Planning of the provisional project

ID	Nom de tâche	Début	Terminer	Durée	2016		
					mars	avr.	mai
1	Etude de l'existant et documentation	01/03/2016	08/03/2016	6j			
2	Analyse des besoins et rédaction du cahier de chages	09/03/2016	16/03/2016	6j			
3	Elaboration du modèle conceptuel	17/03/2016	23/03/2016	5j			
4	Développement de l'option d'Authentification	24/03/2016	31/03/2016	6j			
5	Développement de l'option de Déconnexion	01/04/2016	07/04/2016	5j			
6	Développement de l'option de consultation de l'inventaire	08/04/2016	27/04/2016	14j			
7	Développement de l'option de Recherche d'Articles	28/04/2016	02/05/2016	3j			
8	Développement de l'option d'impression de l'inventaire	03/05/2016	05/05/2016	3j			
9	Développement de l'option d'initialisation de Base	06/05/2016	10/05/2016	3j			
10	Test de l'Application	11/05/2016	17/05/2016	5j			
11	validaion	18/05/2016	23/05/2016	4j			
12	Redaction du Rapport	04/04/2016	24/05/2016	37j			

Analysis

System actors

Storekeepers

Functional needs: Use Case

Gestion des stocks

Storekeeper

Edit Item

Consult inventory — <<extend>> → Edit Item

Consult inventory — <<extend>> → Add item

Print inventory — <<include>>

Print inventory — <<include>> → Authenticate

Search Item — <<include>>

Initialize the local DB — <<include>>
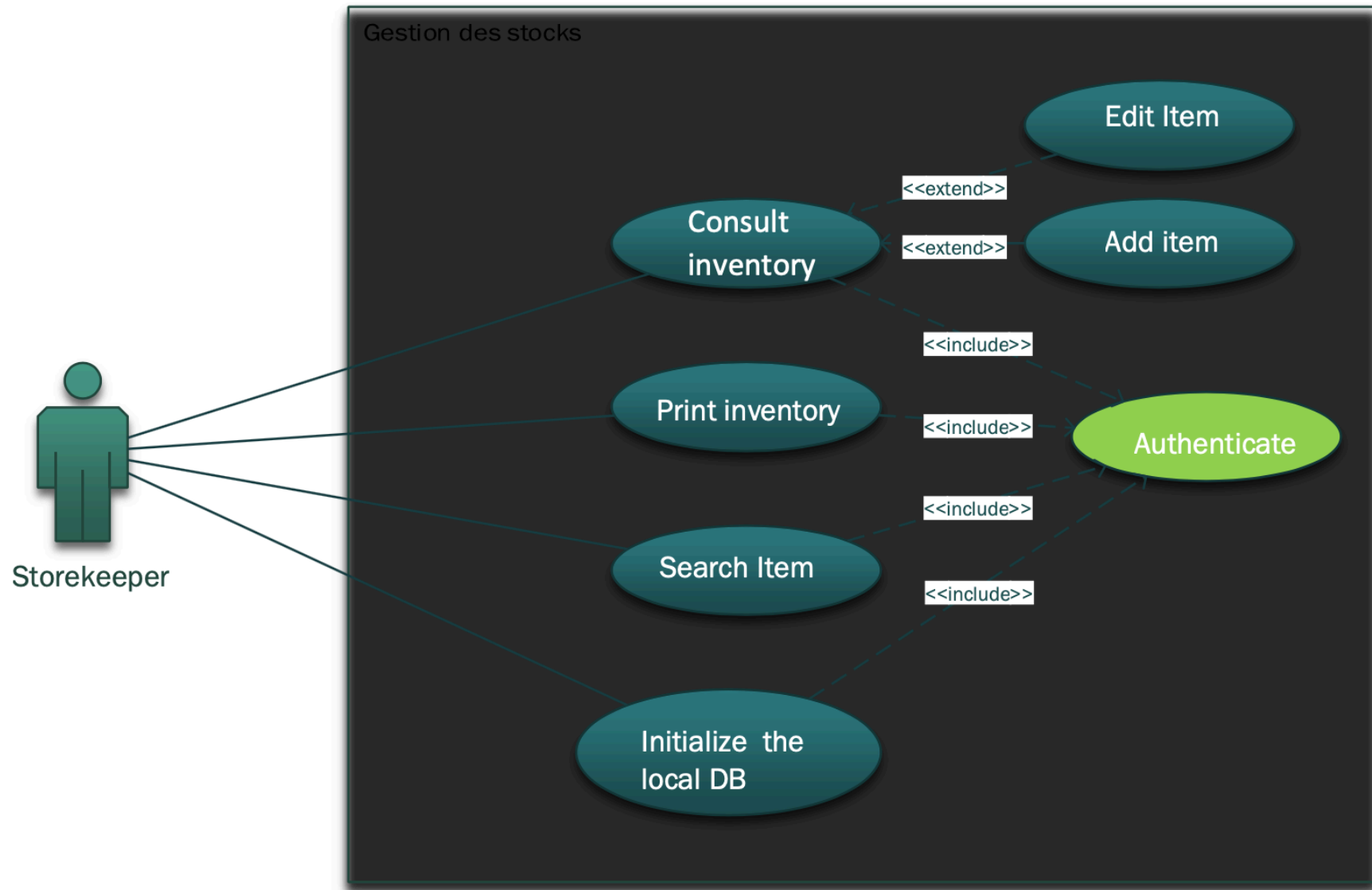
Specifications of non-functional requirements
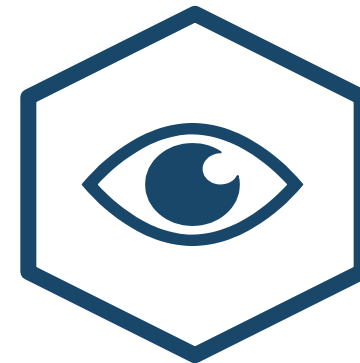
Compatibility

Security

Extensibility

Ergonomics

Conception
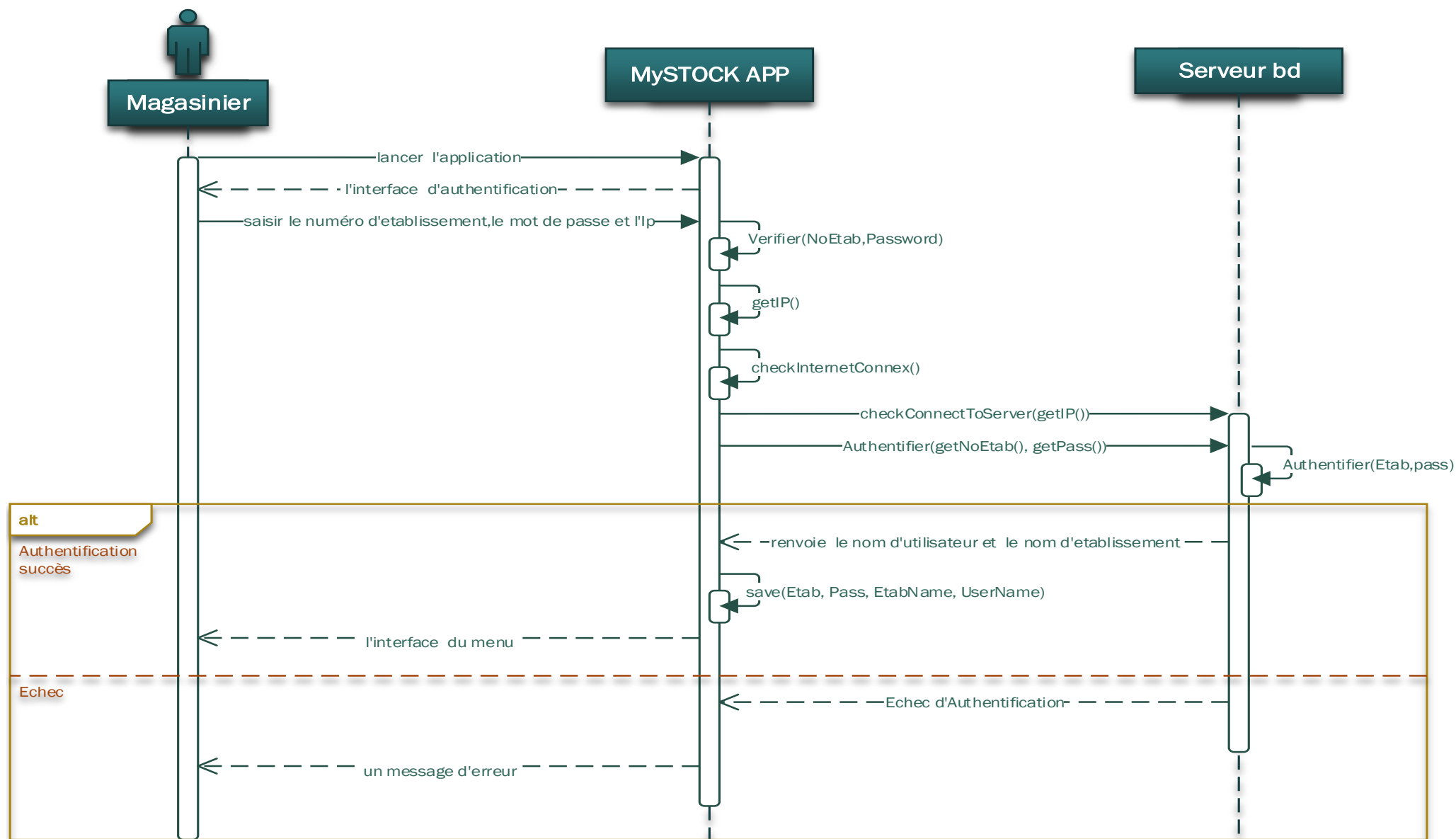
Search Item
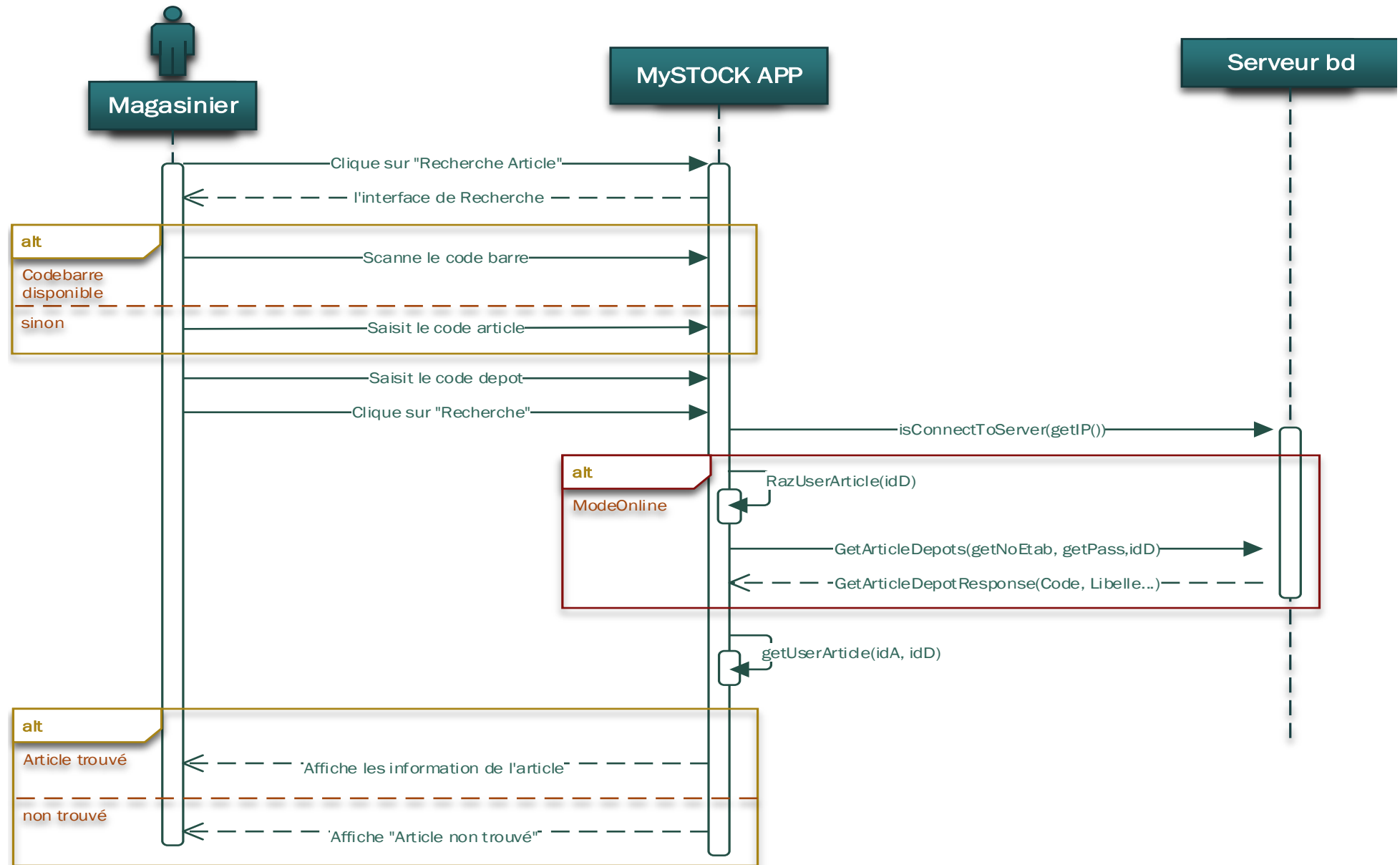
Activity diagram: Authentication

Class diagram

Utilisateur
- -Password: String
- -IP: String
- -UserName: String
- +isConnectToInternet()
- +isConnectToServer()
- +Authentifier()

— Gérer — 1..1   1..*

Etablissement
- -NoEtab: String
- -EtabName: String

1..1

se compose

1..*

Depot
- -code: string
- -libelt: string
- +getUserDepots(): ArrayList<depot>
- +AddUserDepot(code, libelle)
- +RazUserDepot()

se compose   1..1

CBArticle
- -codebarre: string
- -depot: string
- -code: string
- +ReadCBArticle(): ArrayList<CBArticle>
- +AddCBArticle(code, codebarre, depot)
- +getCBArticle(codebarre, depot): String
- +RazCBArticle()

0..1

0..*

Article
- -code: string
- -libelt: string
- -depot: string
- -cat: string
- -fam: string
- -ssfam: string
- -unite: string
- -qte: string
- +getUserArticles(depot): ArrayList<Article>
- +getUserArticles(depot, cat, fam, sfam): ArrayList<Article>
- +getUserArticle(code, depot): Article
- +AddUserArticle(code, libelle, …)
- +setArticleEdit(code, depot, qte)
- +RazUserArticle(depot)

0..*

possede

1..1

Item
- -code: string
- -libell: string

1..*

Categorie
- +getUserCategorie(): ArrayList<Item>
- +AddUserCategorie(code, libelle)
- +RazUserCategorie()

1..1   1..*   se compose

Famille
- +getUserFamille(cat): ArrayList<Item>
- +AddUserFamille(code, libelle, cat)

1..1   se compose   1..*

SousFamille
- +getUserSousFamille(cat, fam): ArrayList<Item>
- +AddUserSousFamille(code, libelle, fam, cat)

Production

System Architecture
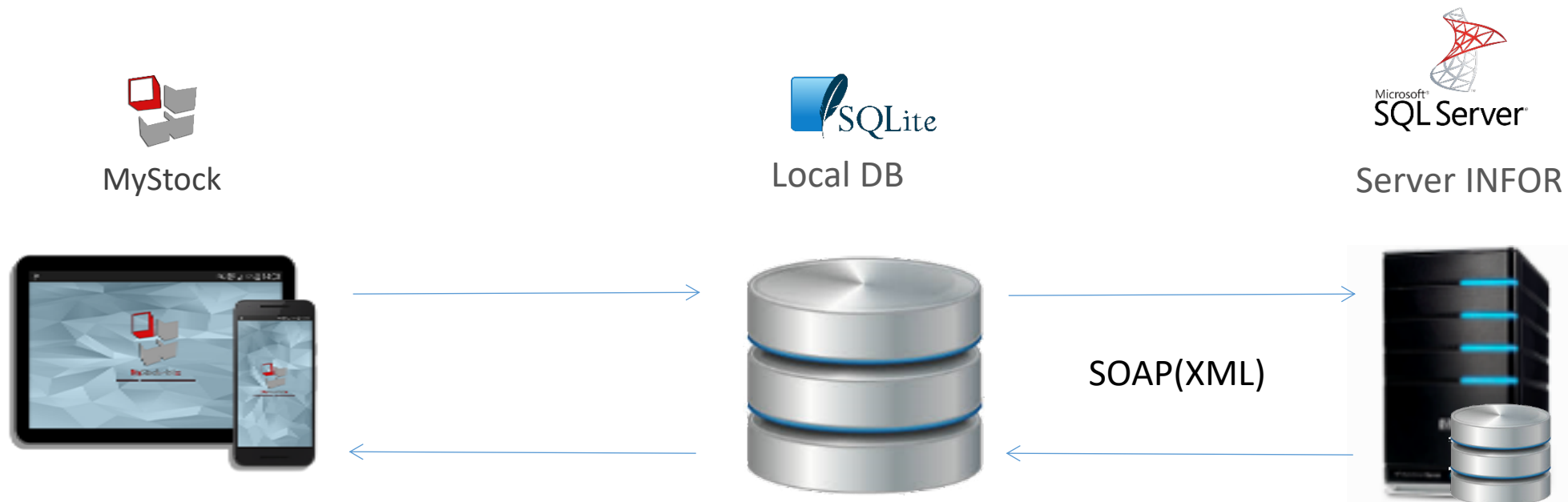
MyStock

Local DB

Server INFOR
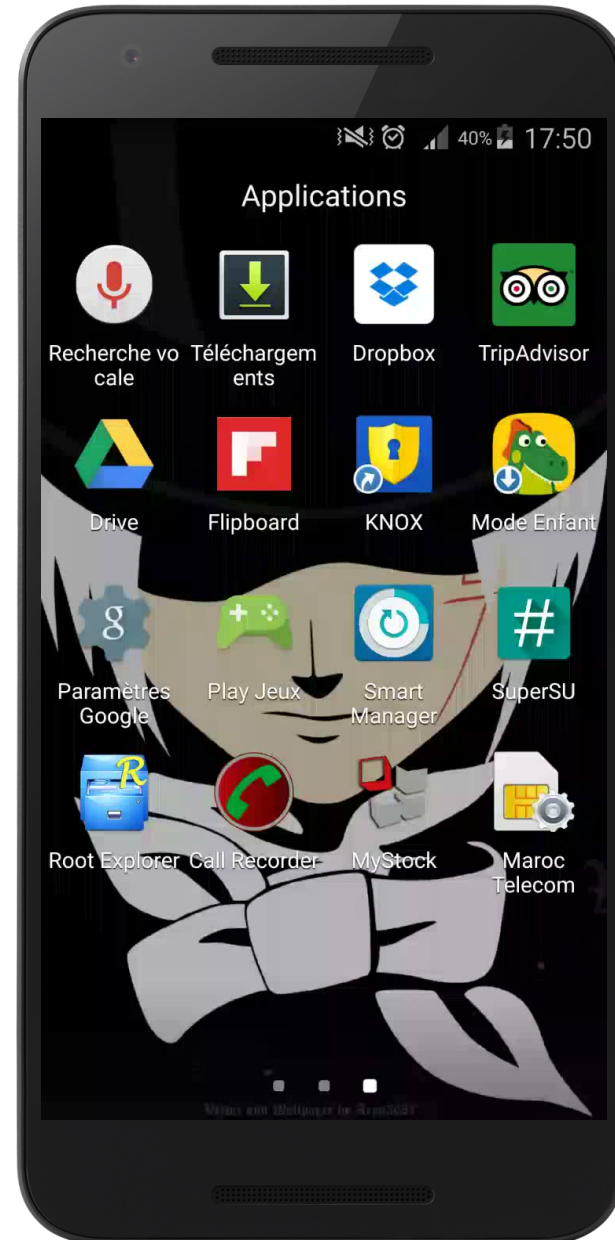
SOAP(XML)

Technical environment
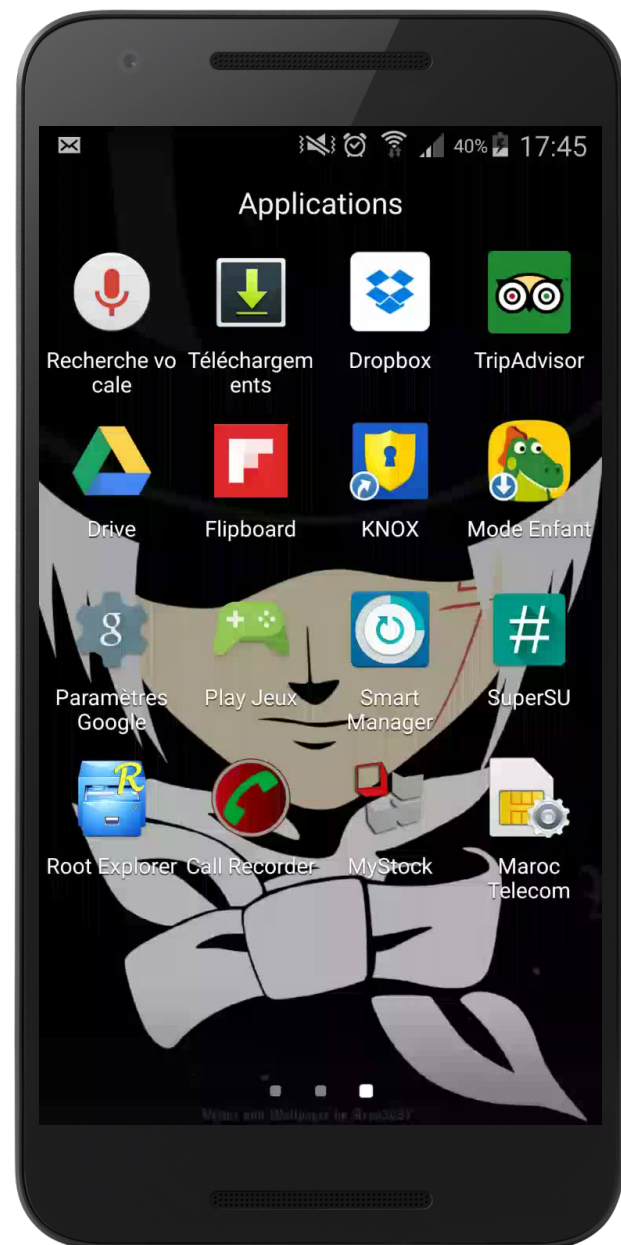
Online Mode

Offline Mode

Conclusion
&
Perspectives

Thank you for your attention