

Université Moulay Ismail Meknès

Rapport de Projet S4

Site Web de Gestion des Conteneurs

Présenté par :

Benjbara Aymen

Soutenu le :09/07/2019.....

Devant le jury :

- M^{me} **Lamini CHAIMAE**.....
- M. **Oubelkacem Ali**
- M. **Benhlime SAID**.....
- M. **Bekri ALI**.....
- M. **Chentoufi Arakil**.....

Année Universitaire : 2018 / 2019

Sommaire

Remerciement	3
Introduction.....	4
Choix techniques du projet	4
<i>Objectifs principaux</i>	5
Cahier des charges	6
Les diagrammes	7
Architecture MVC	10
Réalisation	11
Captures d'écran	12
Conclusion	16

~La qualité est notre premier intérêt~



REMERCIEMENTS

Nous tenons tout d'abord à remercier tous les tuteurs de projet, qui nous ont permis de réaliser notre projet au sein de leurs associations ainsi que pour leurs aides à suivre des instructions claires et précises et de mener à bien notre projet. Ils nous ont aussi indiqué toutes les démarches à suivre tout au long de notre projet mais aussi de nombreux conseils qui nous ont fait avancer rapidement dans la réalisation de notre site internet.

Enfin, nous tenons à remercier l'ensemble des professeurs du département Bachelor qui sans eux, nous n'aurions pas acquis les compétences requises à la réalisation du site web Kargo.

Introduction:

Afin d'appliquer les méthodologies et les notions enseignées jusqu'en deuxième année, nous devons réaliser un projet. Nous avons donc décidé de créer un site web dynamique. Celui-ci permet à nous, étudiants, d'appliquer les connaissances acquises durant notre scolarité.

Le projet nous a été affecté au mois de Janvier 2019 par les biaux de Mr Bekri ALI et Mr Oubelkacem ALI, Ce projet vise la construction d'un Site Internet qui nous permet de faire une gestion globale et professionnelle des Conteneurs.

Choix techniques du projet :

- type de serveur web : TomCat7 & Apache
- module de scripts pour les pages web : JAVA EE
- Type de base de données : SQL

CAHIER DES CHARGES

- **Objectifs stratégiques**

Une fois notre site web mis en place, cela permettra donc de gérer nos conteneurs et nos utilisateurs et de stocker toutes les données dans une base de données et donc pouvoir supprimer et accéder, modifier tout ce qu'on veut. Il sera possible aussi de pouvoir consulter les ports proches et de chercher parmi les conteneurs par trois façons différentes:

Une recherche simple à l'aide de la barre de recherche, ou par le petit box de sélection accoté, ou par une sélection par catégorie dans fenêtre Catégories, ou à l'aide de la « Recherche Avancée » où l'utilisateur peut chercher selon des critères qu'il choisit.

L'utilisateur peut accéder en toute liberté aux conteneurs disponibles dans le site, mais il ne sera pas possible d'effectuer une commande si ce dernier n'est pas identifié.

Une fois l'utilisateur est identifié, il peut donc effectuer des commandes, accéder la liste de ces commandes comme il peut aussi voir ses informations personnels. Il sera automatiquement déconnecté s'il reste inactif pendant 1 heure.

L'Admin a un est un seul compte qui a été spécifier lors de l'implémentation du programme, une fois l'admin est connecté, il peut ajouter des Clients et des Conteneurs, qui seront enregistrés dans la base des données, il peut aussi les modifier et les supprimer. Une liste des Commandes peut être accéder par ce dernier, où l'ID de la commande, l'ID du client et l'ID du Conteneur loué et la date du début de location (qui est toujours la date du jour où la location est effectuée) sont affichés.

Remarque: Tous les conteneurs que l'utilisateur peut y accéder dans notre Site Kargo sont disponibles.

Ces commandes peuvent aussi être supprimées. Même chose pour la liste des dépôts, où l'ID dépôt et ID du conteneur stocké dans ce dépôt sont affichés.

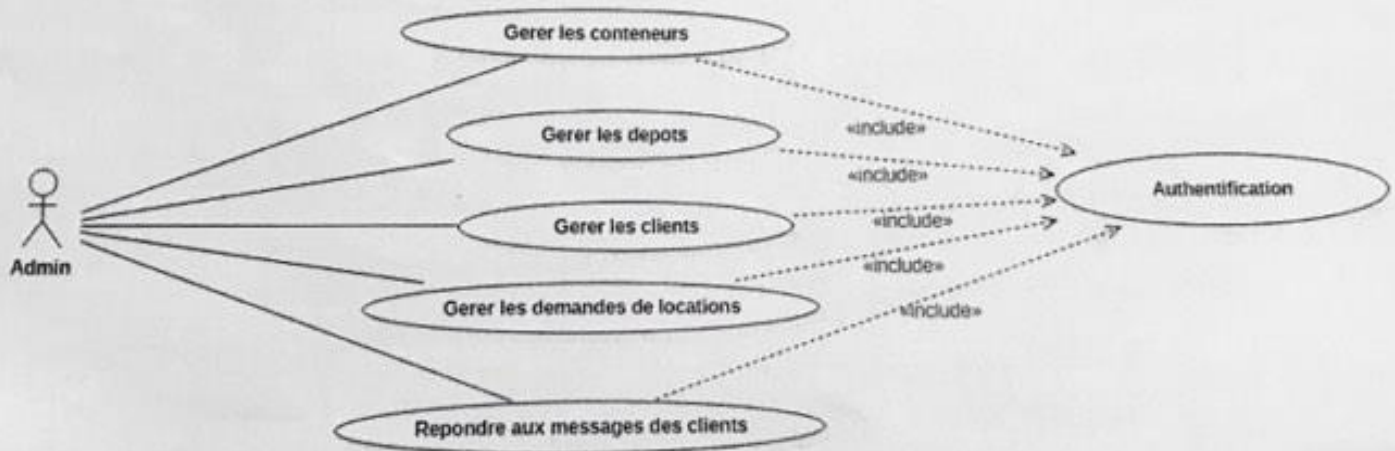
Objectifs principaux:

- *Ouvrir et fermer une Session*
- *Ajouter les conteneurs*
- *Afficher les conteneurs*
- *Modifier les conteneurs*
- *Chercher les conteneurs*
- *Supprimer les conteneurs*
- *Ajouter les clients*
- *Afficher les clients*
- *Modifier les clients*
- *Supprimer les clients*
- *Afficher les dépôts*
- *Supprimer les dépôts*
- *Ajouter les commandes*
- *Afficher les commandes*
- *Supprimer les commandes*

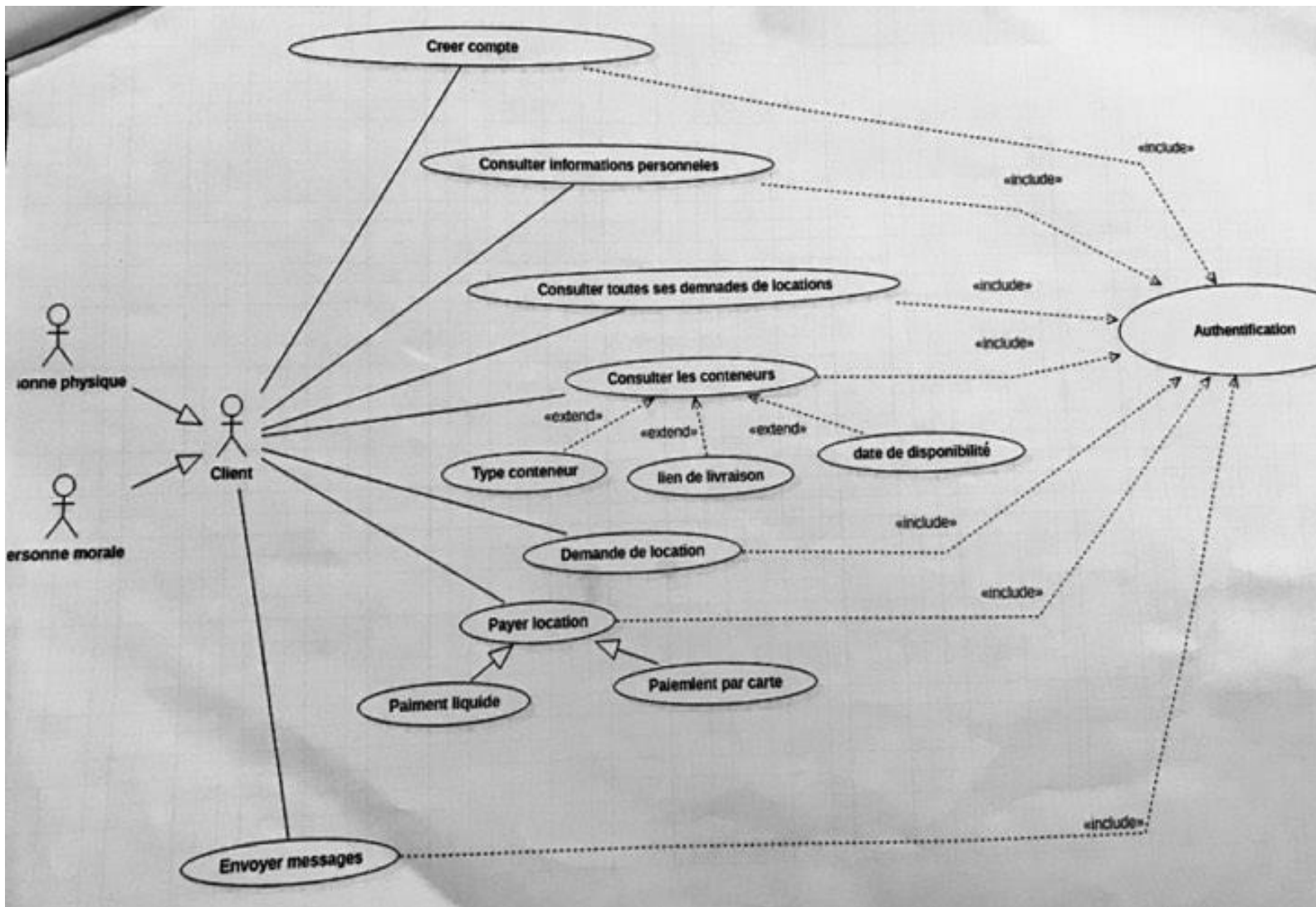
Les diagrammes

- **Diagramme de cas d'utilisation d'administrateur:**

Ce diagramme décrit les utilisations requises d'un système, ou la fonction d'un système. Les acteurs, cas d'utilisation et sujets représentent les principaux concepts de ce diagramme. Un Sujet est un système avec lequel les acteurs et autres sujets interagissent.

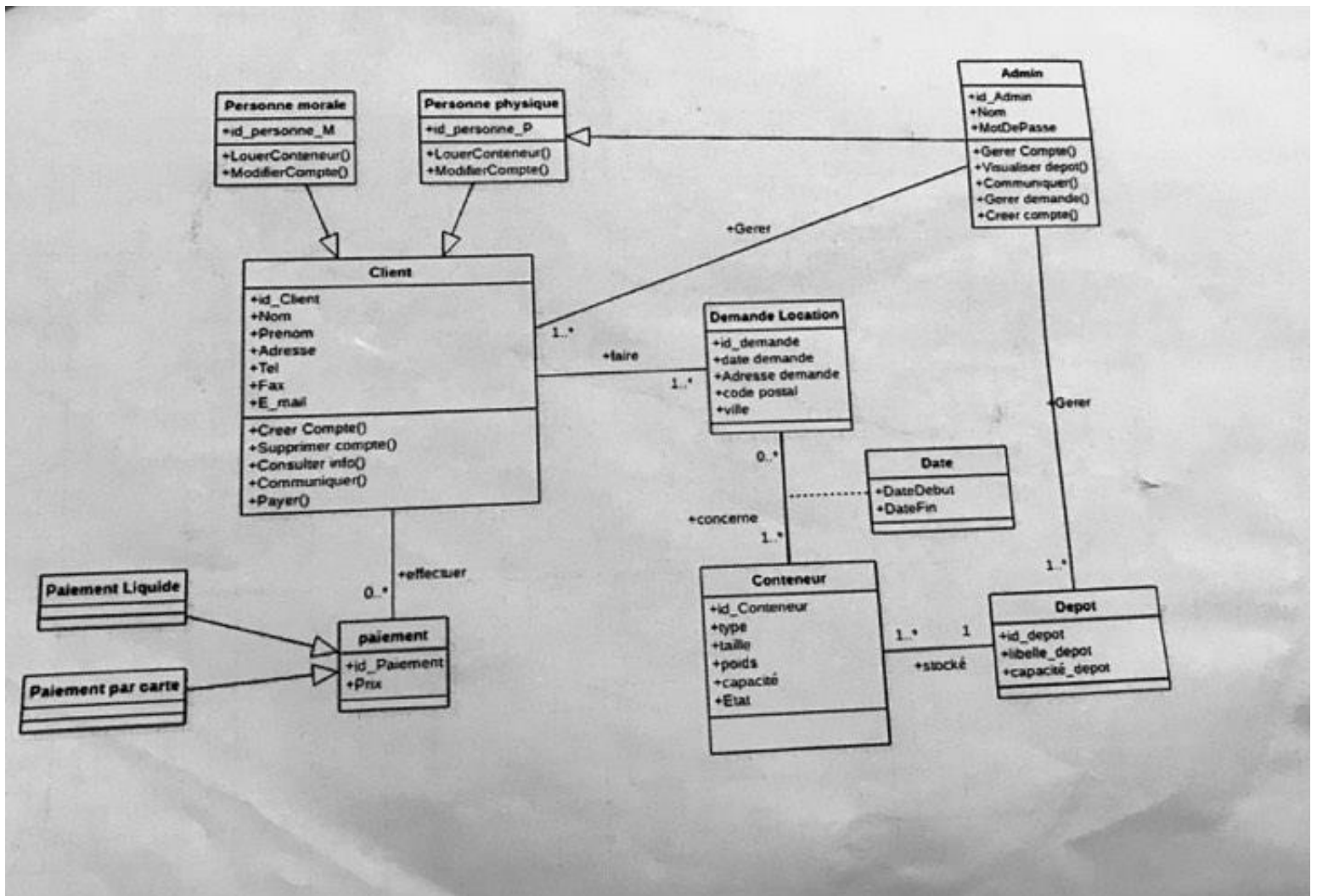


- Diagramme de cas d'utilisation d'utilisateur:



- **Diagramme de classe:**

Ce diagramme consiste à présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

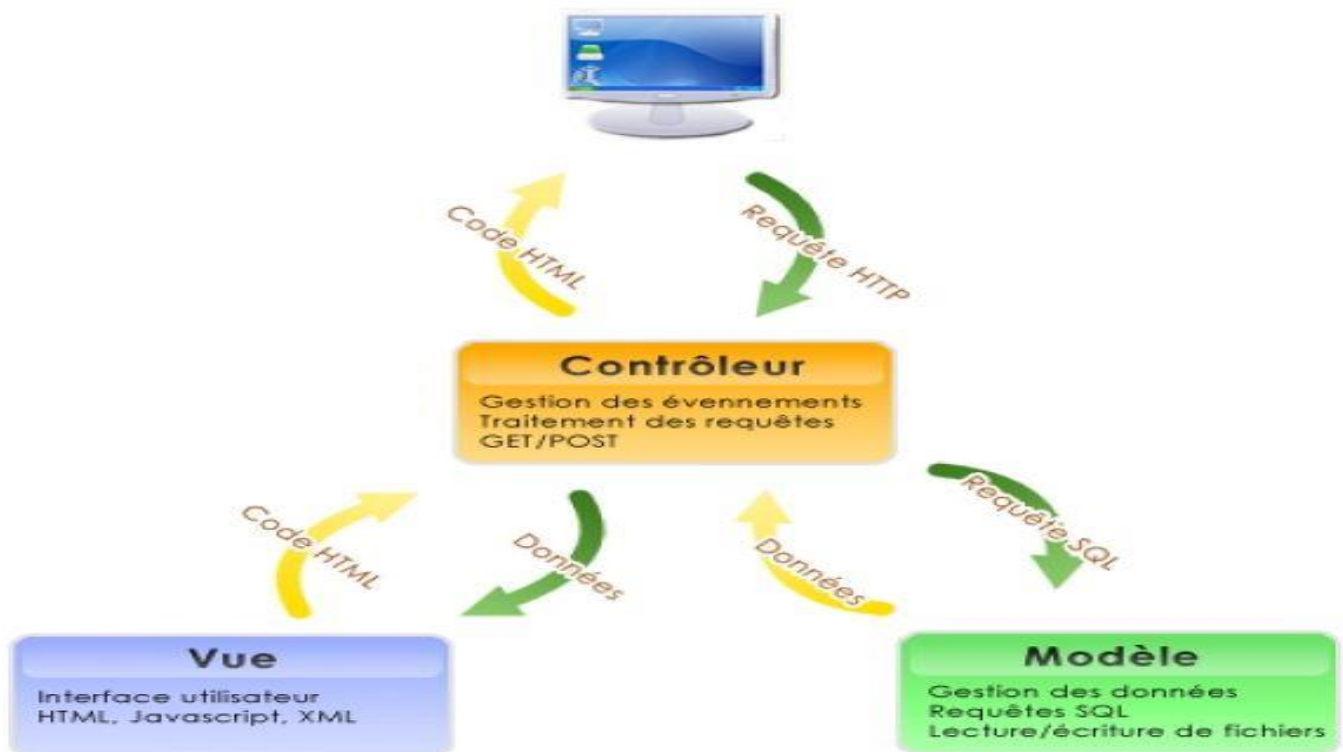


Architecture MVC

Le patron Modèle-Vue-Contrôleur, tout comme les patrons Modèle-vue-présentation ou Présentation, abstraction, contrôle, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

Ce paradigme regroupe les fonctions nécessaires en trois catégories :

- un modèle (modèle de données),
- une vue (présentation, interface utilisateur)
- un contrôleur (logique de contrôle, gestion des événements)



Réalisation

- **Environnement Matériel:**

Ci-dessous les caractéristiques de l'ordinateur sur lequel nous avons développés Kargo pour donner une idée sur les conditions du travail:

- Processus : Intel® Core™ i5 CPU 430M @ 2.26 GHz
- Mémoire installé(RAM) : 8.00 Go
- Type de système : système d'exploitation Windows 64 bits.
- Disque dur : 1 To.
- Ecran : 15,6 pouces cassé ☺.

- **Environnement Logiciel:**

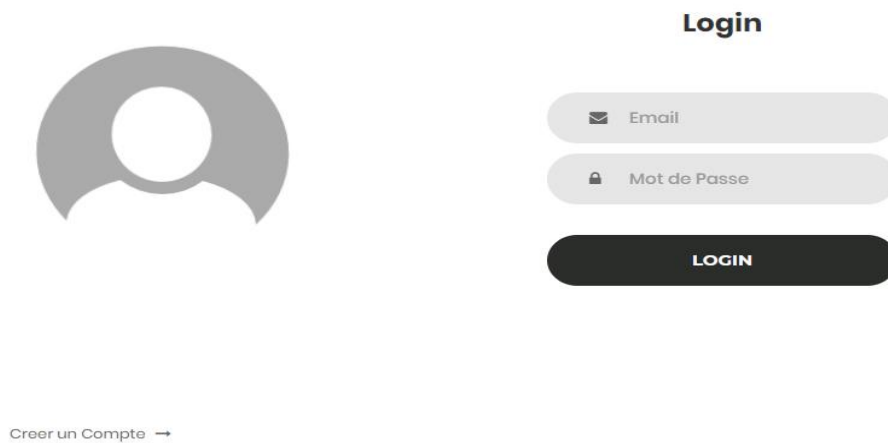
Au cours de cette partie, nous énumérons les différents outils utilisés tout au long de ce projet pour l'étude et la mise en place de notre application.

- Eclipse JEE
- XAMPP
- Autre sites web et forums

Captures D'écran

1) Login

Cette interface présente les champs requis pour s'identifier



The login interface features a large, light gray circular placeholder for a profile picture on the left. To the right, the title "Login" is centered above two input fields: "Email" (with an envelope icon) and "Mot de Passe" (with a lock icon). Below these fields is a dark gray button labeled "LOGIN". At the bottom left, there is a link that says "Créer un Compte" followed by a right-pointing arrow.

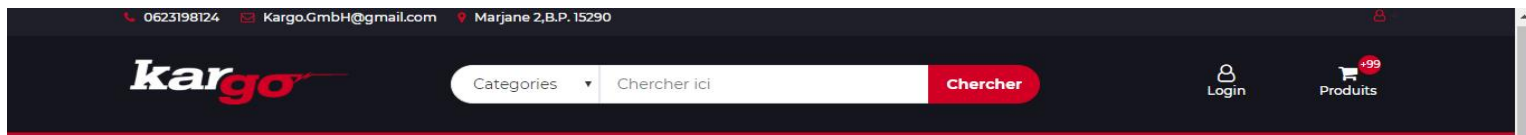
2) Inscription

Cette interface présente les champs requis pour s'inscrire



The registration interface is divided into two main sections. On the left is a black and white photograph of a modern, multi-story building with a glass facade and balconies, illuminated at night. On the right, the title "INSCRIVEZ VOUS" is centered above a series of input fields. These fields are labeled "Nom", "Prenom", "Mot de passe", "Adresse Email", "Adresse", "Telephone", and "Fax", each with a corresponding icon (envelope, lock, house, telephone, and fax). Below the "Fax" field is a dark gray button labeled "Ajouter" followed by a right-pointing arrow.

3) Accueil



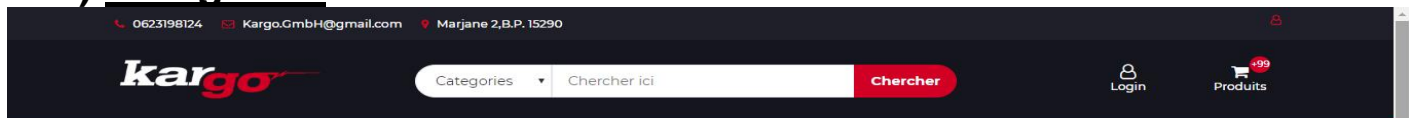
[Accueil](#) [Nos Conteneurs](#) [Categories](#) [Ports Proches](#) [Contactez Nous](#)



NOUVEAUX PRODUITS

_ Cette interface affiche l'accueil de Kargo

4) Categories



[Accueil](#) [Nos Conteneurs](#) [Categories](#) [Ports Proches](#) [Contactez Nous](#)


ACCUEIL / CATEGORIES



VOIR AUSSI

5) Espace Client

Cette interface affiche liste des commandes effectués



Categories ▼ Chercher ici Chercher

[Mon Compte](#) [Déconnexion](#)


[Mes Commandes](#) Informations Personnels

Numero de Commande	Conteneur n°	Client n°	Date Debut de Location	Actions
4	1	11	2019-07-04	Supprimer
5	4	11	2019-07-04	Supprimer
19	1	11	2019-07-07	Supprimer

Showing 1 of 2 Précédent 1 2 ... Suivant

6) Espace Admin

Cette interface affiche la liste des conteneurs disponibles



Welcome **Aymen** | [Déconnexion](#)

[Gestion des Conteneurs](#) [Gestion des Clients](#) [Gestion des Commandes](#) [Gestion des Depots](#)

[Dashboard](#) > [Conteneurs](#)

Management

[Ajouter un Conteneur](#)[Ajouter un Client](#)

Liste des Conteneurs

id_Conteneur	titre	Type	Taille	Prix	Poids	Capacite	Etat	Actions
1	CONTENEUR de Kargo Special edition	DRY	20.0	500.0	1200.0	65.0	disponible	Supprimer Modifier
2	CONTENEUR DE PENSEE AWS 40 OPEN TOP	OPEN TOP	30.0	4500.0	1400.0	30.0	disponible	Supprimer Modifier
3	ABRI Dry Conteneur 20ft	Dry	30.0	1500.0	900.0	55.0	dipsonible	Supprimer Modifier
4	CONTENEUR PLATEAU STOCKAGE SECHE	FLAT RACK	20.0	4500.0	800.0	20.0	disponible	Supprimer Modifier
5	CONTENEUR OUVERTE D'EXPÉDITION	OPEN TOP	30.0	6500.0	1900.0	30.0	disponible	Supprimer Modifier
6	CONTENEUR PLATEAU STOCKAGE SECHE	FLAT RACK	26.0	4000.0	1000.0	26.0	disponible	Supprimer Modifier
-	MSC FLAT RACK 1/50	FLAT	Supprimer Modifier

7) Ajout d'un Conteneur

Cette interface présente les champs requis pour ajouter un Conteneur

Ajouter un nouveau Client (page réservé à l'administrateur)

Nom	<input type="text" value="Nom"/>
Prenom	<input type="text" value="Prenom"/>
Mot de Passe	<input type="text" value="Mot de Passe"/>
Adresse Mail	<input type="text" value="Email"/>
Adresse	<input type="text" value="Adresse"/>
Telephone	<input type="text" value="Telephone"/>
Fax	<input type="text" value="Fax"/>
<input type="button" value="Ajouter"/>	

[Retourner à la liste des commandes](#)

8) Un petit aperçu de Code

```
ConteneurServlet.java
1 package Web;
2
3 import java.io.IOException;
4
5 /**
6  * Cette servlet gère les conteneurs
7  * @author Ayman
8  */
9
10 // @WebServlet("/ConteneurServlet")
11 public class ConteneurServlet extends HttpServlet {
12     private IClient client;
13     private IAdmin admin;
14     private IDepot depot;
15     private IConteneur conteneur;
16     private IDemandeLoc Demande_Location;
17     private ICommande commande;
18     // private IDate Date;
19     // private IPers_Mo Personne_Morale;
20     // private IPers_Ph Personne_Physique;
21
22     @Override
23     public void init() throws ServletException {
24         client=new ClientImpl();
25         admin=new AdminImpl();
26         depot= new DepotImpl();
27         conteneur= new ConteneurImpl();
28         Demande_Location= new DemandLocImpl();
29         // Date=new DateImpl();
30         commande= new CommandeImpl();
31         // Personne_Morale=new Pers_Mo_Impl();
32         // Personne_Physique= new Pers_Ph_Impl();
33     }
34 }

deals.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
4 <%@ page import="entities.Conteneur" %>
5 <%@page import="java.util.ArrayList"%>
6 <%@page import="java.util.List"%>
7 <%@page import="Web.ConteneurServlet"%>
8 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://
9   <html lang="en">
10     <head>
11       <meta charset="utf-8">
12       <meta http-equiv="X-UA-Compatible" content="IE=edge">
13       <meta name="viewport" content="width=device-width, initial-scale=1">
14       <!-- The above 3 meta tags *must* come first in the head; any
15       other meta tags should follow. -->
16       <title>Kargo</title>
17
18       <!-- Google font -->
19       <link href="https://fonts.googleapis.com/css?family=Montserrat" rel="stylesheet">
20
21       <!-- Bootstrap -->
22       <link type="text/css" rel="stylesheet" href="css/bootstrap.min.css">
23
24       <!-- Slick -->
25       <link type="text/css" rel="stylesheet" href="css/slick.css"/>
26       <link type="text/css" rel="stylesheet" href="css/slick-theme.css">
27
28       <!-- nouislider -->
29       <link type="text/css" rel="stylesheet" href="css/nouislider.min.css">
30
31       <!-- Font Awesome Icon -->
32       <link rel="stylesheet" href="css/font-awesome.min.css">
33     </head>
34     <body>
35     </body>
36   </html>
```

Conclusion

Le but de ce projet est de développer une Application Web pour le management des Conteneurs. Une telle Application permettra l'intervention d'un Utilisateur avec un coût minimal.

Tout au long de ce Rapport, nous avons présenté les différentes étapes de réalisation de l'Application Web Kargo. Pour développer cette Application nous avons utilisé le langage UML, ce qui nous a permis de mener correctement la tâche d'analyse des besoins à l'aide du Diagramme de Classe et de Cas d'utilisation. Nous avons aussi utilisé la technologie JAVA que nous avons jugé nécessaire pour accomplir ce Projet.

Ce Projet représente une opportunité pour s'initier à la vie professionnelle dans un milieu réel et avoir un début d'expérience significative, et il nous a appris comment compter sur soi pour résoudre les problèmes au cas où ils se présentent, comment être rigoureux dans notre travail, comment être bien organisés pour accomplir dans les meilleurs délais, et meilleures conditions de travail qui nous a été confiés.

Le Projet est dans le nuage sous l' adresse URL suivante:

<https://mega.nz/#F!TT4WzCJQ!weZmjVJVW41gJ-jkhz4S8Q>