

# UE Full Stack

TP3 UE Full Stack

---

<b>1. Introduction</b>	<b>3</b>
1.1. Objet Du Document	3
1.2. Objectifs Du Projet	3
1.3. Contenu Du Document	3
1.4. Convention	3
<b>2. Exigences Fonctionnelles Et Opérationnelles</b>	<b>4</b>
2.1. Généralités et missions du système	4
2.2. Document de référence	4
<b>3. Exigences Techniques</b>	<b>4</b>
3.1. Architecture	4
3.2. Technologies	4
<b>4. Exigences Du Projet</b>	<b>4</b>

## 1. Introduction

### 1.1. Objet Du Document

Le présent document décrit le contexte et le périmètre du TP3 proposé aux étudiants du Master « Génie de l'Informatique Logicielle » pour l'année universitaire 2024-2025 de l'UE Full Stack. Ce document sert de référence pour présenter l'ensemble des exigences fonctionnelles et de qualités associées au TP. Il définit également la liste des fournitures relatives au projet.

### 1.2. Objectifs Du Projet

Ce projet est une refonte d'une application déjà existante qui permet la gestion de boutiques, de produits et de catégories associées aux produits.

Le document qui définit les exigences de réalisation est fourni avec ce document et est intitulé "Projet UE Full Stack".

### 1.3. Contenu Du Document

La première partie du document est consacrée à une description générale du projet et à la définition d'un certain nombre de notions relatives aux fonctionnalités ou aux technologies à mettre en œuvre dans le projet.

La lecture de ces définitions constitue un prérequis indispensable pour la bonne compréhension de la suite du document.

Chacune des fonctions de ce TP fait l'objet d'une description dans laquelle sont exprimées les exigences à saisir. Ce document définit ensuite les exigences techniques applicables à la conception et au développement du projet. Enfin, le document définit les exigences de management qui devront être respectées pour la conduite du projet.

### 1.4. Convention

Le présent document a été rédigé en respectant les règles suivantes :

- Les exigences sont référencées selon le format suivant :
  - la lettre « E » pour « exigence » suivie d'un tiret bas « \_ »,
  - trois lettres pour codifier la catégorie fonctionnelle de l'exigence, suivies d'un tiret bas « \_ »,
  - un numéro d'incrément de 10 en 10.
- Les codes utilisés au deuxième point sont les suivants:
  - BTQ pour les exigences liées à la gestion des boutiques
  - PRD pour les exigences liées à la gestion des produits

- FIX pour les exigences liées aux bugs de l'application
- AME pour les exigences liées à l'amélioration du projet
- ARC pour les exigences liées à l'architecture du système
- TEC pour les exigences techniques de réalisation du système
- PRO pour les exigences du projet

## 2. Exigences Fonctionnelles Et Opérationnelles

### 2.1. Définitions préalables

Dans la suite du document, les définitions ci-après seront utilisées pour la formulation des exigences :

- Long-term support (LTS) : Version spécifique d'un logiciel dont le support est assuré pour une période de temps plus longue que la normale

### 2.2. Généralités et missions du système

Le système doit permettre la gestion de boutiques, de produits et de catégories.

Les fonctionnalités devront être développées au travers d'une IHM dédiées et accessibles au travers d'une application web (SPA) reposant sur des APIs web.

### 2.3. Document de référence

Ce document reprend toutes les exigences du document fourni avec ce TP et est intitulé "Projet UE Full Stack", les exigences supplémentaires concernant le TP3 seront décrites ci-dessous.

### 2.4. Gestion des boutiques

Le système de gestion des boutiques permet d'effectuer des opérations comme la recherche et le tri selon différents critères, la consultation, la modification ou la suppression des boutiques.

Les exigences ci-après sont applicables au système de gestion des boutiques.

**E\_BTQ\_50** Le système permet d'effectuer une recherche paginé sur toutes les boutiques qui sont présente en affichant :

- Leur nom
- Leur date de création

- Le nombre de produits
- Le nombre de catégorie distincte qui sont associées aux produits de la boutique<sup>1</sup>
- Si la boutique est en congé

**E\_BTQ\_65** Le système permet d'effectuer une recherche plein texte via la base de donnée Elasticsearch sur les boutiques avec un ou une combinaison de filtres qui sont :

- Si la boutique est en congé
- Date de création
  - Après une date précise
  - Avant une date précise
  - Entre deux dates précises

## 2.5. Gestion des produits

La gestion des produits permet d'effectuer des opérations comme la recherche et le tri selon différents critères, la consultation, la modification ou la suppression des produits.

Les exigences ci-après sont applicables au système de gestion des produits :

**E\_PRD\_15** Le système doit gérer dans son système de gestion de données les prix des produits en centimes. (i.e. si dans le système de gestion de donnée le prix du produit est de 1234, alors sur le client il doit être affiché 12,34€)

## 2.6. Bug de l'application

**E\_FIX\_10** Le système vérifie que les horaires des boutiques n'entre pas en conflit. Par exemple, on ne peut pas ajouter un horaire pour une boutique le lundi de 8h à 17h et le lundi de 15h à 19h.

## 2.7. Amélioration du projet

**E\_AME\_10** L'existant pourra être modifiée dans le but de fournir une application maintenable, lisible et/ou cohérente (i.e. nom de variable, fonction, modèle, ...).

**E\_AME\_20** Des index pourront être présents en base de données pour permettre une recherche plus rapide si la base venait à être conséquente<sup>2</sup>.

<sup>1</sup> Donnée non présente dans le projet modifié de cette année

<sup>2</sup> Une bonne pratique est d'ajouter un index sur chaque foreign key en base.

[E\\_AME\\_30](#) Le client du système doit être responsive.

## 3. Exigences Techniques

### 3.1. Architecture

Le système est basé sur une API REST.

[E\\_ARC\\_40](#) Le système utilise une base de données Elasticsearch en plus de son système de gestion données initial.

[E\\_ARC\\_50](#) La base de données Elasticsearch du système utilise le port 9200.

[E\\_ARC\\_60](#) La base de données Postgresql sera initialisée tel que défini dans le fichier `docker-compose.yml` dans le dossier `shop-server` .

### 3.2. Technologies

[E\\_TEC\\_50](#) Le serveur doit être mis à jour avec la dernière version LTS du framework disponible.

[E\\_TEC\\_60](#) Le client doit être mis à jour avec la dernière version LTS du framework disponible.

[E\\_TEC\\_70](#) Les librairies du serveur utilisées doivent être mise à jour avec la dernière version LTS compatible avec le framework dans lequel elles sont utilisées.

[E\\_TEC\\_80](#) Les librairies du client utilisées doivent être mise à jour avec la dernière version LTS compatible avec le framework dans lequel elles sont utilisées.

[E\\_TEC\\_90](#) Les boutiques déjà présentes dans le système de gestion de données initial doivent aussi être présentes dans la base de données Elasticsearch au lancement de l'application ou par tout autre moyen expliqué dans la documentation fourni lors du rendu du projet.

[E\\_TEC\\_100](#) Lors de la mise à jour du système, une interruption du serveur est autorisée.

## 4. Exigences Du Projet

[E\\_PRO\\_170](#) Des outils de configuration doivent être mis en place. Ils intègrent :

- 
- Une configuration docker qui permet de lancer le client. En plus de l'exigence E\_PRO\_10

[E\\_PRO\\_180](#) Le client doit tourner sur le port 4200.

[E\\_PRO\\_190](#) Le client doit avoir une gestion des exceptions et des erreurs qui permettent de prévenir l'utilisateur de sa mauvaise utilisation du système.

[E\\_PRO\\_200](#) Une documentation de mise à jour du système doit être disponible et fournie dans la documentation du projet<sup>3</sup>.

[E\\_PRO\\_210](#) Le projet sera rendu au format tar.gz. Le nom du dossier sera le nom du binôme ou de l'étudiant qui aura réalisé ce projet, il contiendra :

- Le code source et le git du client
- Le code source et le git du serveur
- Un rapport par participant du projet comme décrit pour [E\\_PRO\\_60](#)
- La configuration du docker
- La documentation du projet
- Et toute autre documentation que vous jugerez utile.

---

<sup>3</sup> Pour noter ce projet, je vais lancer le docker compose avec les fichiers initiaux, puis avec la documentation que vous m'aurez fourni je procéderais à la mise à jour du client, du serveur et des bases de données.