



WebMenu

Nom du candidat : **Esther Pham**

N° de candidat : **151481**

Description générale du projet d'examen :

Application Web et son serveur REST permettant d'afficher sur un écran le prix des consommations (boissons et petite restauration) d'une buvette, d'une cafétéria ou d'un bar. L'écran est positionné en mode portrait et les consommations sont regroupés par type (boisson sans alcool, bière, vin blanc, vin rouge, snack, sandwich...). Chaque article est présenté avec sa description, son volume/quantité/poids et son prix. Il faut prévoir plusieurs pages car cela doit être bien visible par l'assistance, par exemple une page pour les boissons sans alcool et bières, une autre pour les vins et une autre pour les en-cas. Les pages se succèdent/tournent à l'écran de manière automatique toutes les 5-10 secondes configurable. Il doit être possible d'adapter les consommations en temps réel via une page d'administration protégée par un mot de passe. La base de données relationnelle est existante et déjà nourrie.

Technologies : HTML, CSS, JavaScript, PHP ou Java ou NodeJS ou Python, MySQL

Domaine du TPI : (cocher la case de couleur correspondante)

<input checked="" type="checkbox"/> Développement	<input type="checkbox"/> Réseau	<input type="checkbox"/> Systèmes
---	---------------------------------	-----------------------------------

Tracé d'activité de l'apprenti/-e durant la dernière année d'apprentissage :

Stage et travaux sur mandats (projets) en phase de professionnalisation

Coordonnées de l'apprentie :

E-Mail : Esther.Pham@studentfr.ch
Tél. : 076 361 52 55
Prénom : Esther Manh Xuân
Nom : Pham
Adresse : École des Métiers
Chemin du Musée 2
1700 Fribourg

Le supérieur professionnel

Pierre-Alain Mettraux

Lieu et Date

Fribourg, le 8 février 2025

Table des matières

1	DONNÉE DU PROBLÈME	3
2	FORMULATION DU MANDAT – ÉTAT DÉSIRÉ	3
2.1	DESCRIPTION ET OBJECTIFS DU PROJET	3
2.2	MAQUETTES	4
2.3	CONCEPTION DU SYSTÈME	4
2.4	EXIGENCES FONCTIONNELLES	4
2.5	ANALYSE.....	6
2.6	CONCEPTION.....	6
2.7	RÉALISATION ET TESTS	6
3	INFRASTRUCTURE NÉCESSAIRE	6
4	CONNAISSANCES PRÉALABLES.....	6
5	TRAVAUX PRÉPARATOIRES	6
6	STANDARDS D'ENTREPRISE	7
7	DOCUMENTATION OBLIGATOIRE.....	7
7.1	UNE PLANIFICATION	7
7.2	UN JOURNAL DE TRAVAIL.....	7
7.3	UNE DOCUMENTATION D'ANALYSE, DE CONCEPTION ET DE RÉALISATION	7
7.4	UN WEB SUMMARY.....	8
7.5	LA REMISE DE LA DOCUMENTATION	8
8	CRITÈRES D'ÉVALUATION	8
9	DÉLAIS.....	8

1 Donnée du problème

Dans des bars de manifestations, dans des bars éphémères, dans une cafétéria ou dans tout autre lieu de consommation de boissons, il n'est souvent pas simple d'afficher clairement une liste de prix. On peut bien sûr imprimer des feuilles plus ou moins grandes, utiliser un tableau noir ou un tableau blanc mais la visibilité et l'accès à l'information peuvent être compliqués et on ne parle pas de la mise à jour des informations si un article viendrait à être épuisé.

Ce projet propose une alternative informatique qui utilise un navigateur Web qui affiche sur une TV ou sur un écran d'ordinateur la liste des prix des articles proposés aux clients. L'application à développer est basée sur des pages HTML pour le côté client et une application serveur hébergé sur un serveur Web qui accède à la base de données relationnelle pour servir les demandes du client Web. Cela doit fonctionner aussi bien pour un affichage sur une TV que pour plusieurs TV.

2 Formulation du mandat – état désiré

2.1 Description et objectifs du projet

Le but du projet est de développer :

1. L'application cliente Web doit permettre d'afficher la liste des prix des articles disponibles à la vente au public. Ces articles incluent différentes sortes de boissons et d'en-cas, regroupés par type. Parmi ces catégories, on retrouve les boissons sans alcool, les boissons chaudes, les vins (blancs, rosés, rouges, pétillants, ouverts ou en bouteille), les bières (à la pression ou en bouteille), ainsi que les en-cas tels que les sandwiches, les croque-messieurs ou encore les planchettes.

Chaque article est présenté avec une description, son volume, sa quantité ou son poids, ainsi que son prix. Si un article devient épuisé, il doit disparaître automatiquement de l'affichage après mise à jour dans la base de données.

L'écran d'affichage doit être en mode portrait afin d'afficher une seule colonne d'articles avec une mise en page claire et suffisamment lisible par le public. L'affichage est réparti sur plusieurs pages pour garantir une bonne visibilité des informations. Par exemple, une page peut être dédiée aux boissons sans alcool et aux bières, une autre aux vins et une autre aux en-cas. Ces pages se succèdent automatiquement toutes les cinq à dix secondes, une durée qui doit être configurable depuis la base de données.

Les paramètres généraux de couleur et de police de caractères sont stockés dans la base de données. Chaque groupe d'articles peut également posséder sa propre couleur et sa propre police afin de mieux les distinguer. L'ordre des articles et la répartition des groupes sur les différentes pages d'affichage sont aussi enregistrés en base de données.

Enfin, à chaque cycle complet d'affichage des pages, l'application cliente doit interroger le serveur afin de récupérer d'éventuelles modifications effectuées par le responsable du débit de boissons. Cela permet de garantir que les informations affichées sont toujours à jour et correspondent à la disponibilité réelle des articles.

2. L'application serveur REST a pour rôle de répondre aux requêtes du client Web et d'accéder à la base de données afin de fournir les informations nécessaires à l'affichage. Lorsqu'un client demande une page, le serveur doit lui transmettre les groupes de consommations ainsi que les articles associés. Chaque article est accompagné des couleurs et des polices définies en fonction des préférences enregistrées, et l'ensemble des données est structuré dans le bon ordre. Ces informations sont envoyées au client sous forme de messages JSON. Cette partie de l'application ne nécessite aucune authentification.

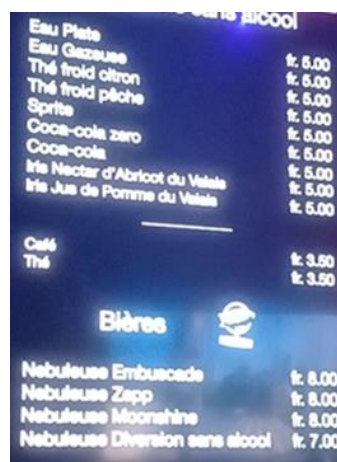
Le responsable doit également pouvoir modifier les articles stockés dans la base de données. Il doit, par exemple, être en mesure d'indiquer qu'un article est épuisé afin que cette information soit répercutée sur l'affichage du client Web. Pour cela, une ou plusieurs pages doivent être prévues afin de gérer les articles.

Étant donné que ces pages permettent la modification des données, un accès sécurisé est nécessaire, tant au niveau de l'interface client que des points d'entrée du serveur. Dans cette

première version de l'application, un mot de passe stocké en base de données est suffisant pour l'authentification. Un jeton généré sera utilisé pour la vérification des accès au serveur, garantissant ainsi une sécurité minimale pour la gestion des articles.

2.2 Maquettes

Les maquettes seront proposées par le candidat. Avec la base de données existante, il est possible de déduire les contours de nos affichages. Voici tout de même, une représentation d'un affichage qui peut donner des premières pistes.



2.3 Conception du système

L'application cliente doit se baser sur un modèle à couches avec à son sommet la couche vue avec ses fichiers HTML/CSS, en dessous la couche contrôleur en JavaScript, ensuite la couche service aussi en JavaScript. Chaque vue doit posséder son contrôleur. Les technologies à utiliser sont en lien avec ce qui a été vu durant la formation à l'EMF. On peut citer HTML, JavaScript, CSS. Si des compétences sur des technologies ou des frameworks ont été acquises par le candidat lors de projet ou de stage durant sa dernière année, elles sont bienvenues lors de la réalisation de ce TPI.

L'application serveur doit aussi se baser sur un modèle à couches qui structure une couche contrôleur où sont positionnées les méthodes d'accès aux Webservice/REST et une couche métier pour le traitement et les accès à la base de données. Les technologies à utiliser sont aussi en lien avec ce qui a été vu durant la formation à l'EMF. On peut citer le PHP, MySQL/MariaDB mais aussi le Java. Si des compétences sur des technologies ou des frameworks ont été acquises par le candidat lors de projet ou de stage durant sa dernière année, elles sont aussi bienvenues.

L'interface est en français et il n'est pas nécessaire de gérer le multilinguisme. La base de données ne possède pas d'utilisateurs mais un simple mot de passe pour sécuriser les modifications de la base de données. Il faut donc prévoir un popup de saisie du mot de passe et surtout faire la gestion de la sécurité d'accès. Normalement les commandes HTTP GET sont en accès libre sur le serveur et les commandes HTTP POST, UPADTE, DELETE et PATCH doivent être protégées.

2.4 Exigences fonctionnelles

Exigence fonctionnelle	Description	Priorité Haute Moyenne Basse	Temps estimé (jours)
Administration	Administration, planning, visites des experts et séances avec le supérieur professionnel.	Haute	0.5 jour

Analyse / Conception	<p>À partir de ce document et de la discussion avec le supérieur professionnel, analyse de la base de données existantes, la structuration du client et du serveur ainsi que la création des mockups.</p> <p>La documentation doit être traitée en parallèle avec l'analyse et la conception</p>	Haute	1.5 jour
Réalisation des WebServices GET (lecture des articles)	<p>L'implémentation de cette exigence doit permettre :</p> <ul style="list-style-type: none"> De créer la structure du serveur De se connecter à la base de données De créer les différents entrypoints (GET) du serveur De lire les données de la base de données, les traiter et les mettre au format JSON D'implémenter la logique des pages et des tris <p>La documentation doit être traitée en parallèle avec les développements.</p>	Haute	2 jours
Réalisation des pages d'affichage des articles	<p>L'implémentation de cette exigence doit permettre :</p> <ul style="list-style-type: none"> De créer la structure du projet côté client De récupérer les données du serveur et de les mettre en page De gérer la rotation des pages De gérer la relecture de des données et la mise à jour des pages <p>La documentation doit être traitée en parallèle avec les développements.</p>	Haute	2 jours
Réalisation des WebServices POST/UPDATE/DELETE (mise à jour des articles)	<p>L'implémentation de cette exigence doit permettre :</p> <ul style="list-style-type: none"> De gérer le login (mot de passe) et la sécurité des entrypoints De créer les différents entrypoints serveur De gérer les données avec les opérations CRUD sur la base de données <p>La documentation doit être traitée en parallèle avec les développements.</p>	Haute	2 jours
Réalisation des pages de modification des articles	<p>L'implémentation de cette exigence doit permettre :</p> <ul style="list-style-type: none"> Se connecter avec le mot de passe De créer une page pour modifier les articles existants (description, prix et disponibilité les données du serveur et de les mettre en page) De créer une page pour ajouter un article dans un groupe De créer une page pour effacer un article dans un groupe <p>La documentation doit être traitée en parallèle avec les développements.</p>	Haute	2 jours
Réalisation supplémentaire sur la modification des groupes	<p>L'implémentation de cette exigence doit permettre :</p> <ul style="list-style-type: none"> D'ajouter, de modifier ou d'effacer un groupe d'articles De gérer les groupes sur les différentes pages De gérer les couleurs de l'application, des groupes et des articles De changer la vitesse de rafraichissement des pages. <p>La documentation doit être traitée en parallèle avec les développements.</p>	Basse	

2.5 Analyse

Cette phase de travail permet d'analyser les besoins tels qu'ils ont été décrits dans la donnée du problème, en les représentant par des schémas adéquats.

Durant la phase d'analyse, l'apprentie accomplit les tâches suivantes :

- Récupération du modèle de base de données
- Cas d'utilisation UML
- Analyse des différentes situations nécessaires au fonctionnement de l'application
- Analyse des flux entre le client et le serveur
- Maquettes (scans de documents manuscrits est une possibilité)
- Documentation

2.6 Conception

Durant la phase de conception, l'apprentie accomplit les tâches suivantes :

- Planification des tests fonctionnels de l'application
- Définitions des modèles à couche côté client et côté serveur
- Documentation

2.7 Réalisation et tests

Durant la phase de réalisation, l'apprentie accomplit les tâches suivantes :

- Création du serveur
- Mise en place des couches et programmation de la lecture des articles
- Création du client
- Mise en place des couches et programmation l'affichage des articles
- Programmation de la mise à jour des articles côté serveur avec l'aspect sécurité à gérer
- Programmation de la mise à jour des articles côté client avec l'aspect sécurité à gérer
- Réalisation des tests fonctionnels de l'application
- Documentation tout au long du projet

3 Infrastructure nécessaire

Pour la réalisation du projet, l'apprentie disposera de l'infrastructure suivante :

- Un PC Windows 11 (poste standard de l'atelier A45)
- Un écran permettant un affichage en mode portrait connecté au PC de l'atelier

4 Connaissances préalables

Les compétences JavaScript, HTML, CSS, PHP, Java, JPA et SQL ont été travaillées dans différents modules. Les projets de 4^e année ont amené des expériences supplémentaires. Les outils et les technologies sont suffisamment maîtrisés pour mener ce projet à terme.

5 Travaux préparatoires

- Installation des outils de développement qui sont laissés au choix du candidat et des outils d'administration
- Mise en place d'outil de dépôt GIT pour gérer les versions et les sauvegardes

- Préparation d'un serveur avec un accès public pour l'hébergement de la partie serveur du projet
- Test d'une librairie JavaScript pour la rotation de pages avec si possible avec un effet agréable à l'œil au moment du changement de page
- Test d'un procédé permettant de gérer l'accès aux points d'entrée du serveur qui modifient les données de la base de données.

6 Standards d'entreprise

Les diagrammes UML permettront de comprendre la structure du programme.

Les standards utilisés sont ceux de l'EMF, mobilisés durant les projets de développement et la phase de professionnalisation :

- Tous les paramètres configurables doivent se trouver dans un fichier de configuration externe
- Une logique unique doit être respectée quant au nommage (pertinence des noms et logique des noms) des variables, classes et méthodes
- L'utilisation des versions récentes de JavaScript est demandée (let, fonction flèche ...)

7 Documentation obligatoire

À la fin du projet, l'apprentie doit fournir différents documents.

7.1 Une planification

Cette planification doit être réalisée au début du projet avant toute autre action (selon modèle fourni). Elle décrit les étapes importantes du projet ainsi que la durée estimée correspondante. Elle doit être validée par le supérieur professionnel.

7.2 Un journal de travail

Ce document décrit les diverses étapes et activités liées au projet (selon modèle fourni).

7.3 Une documentation d'analyse, de conception et de réalisation

Ce document est basé sur le modèle proposé par la commission d'examen et accessible sur pkOrg. Ce document est composé de :

La première partie concerne les généralités et est composée de :

1. Introduction
2. Contexte de l'application

La seconde partie concerne les travaux préliminaires et est composée de :

1. Travaux effectués
2. Résultats pertinents pour la suite du projet

La troisième partie du document détermine les exigences et contraintes du projet et permet la justification des choix pour la réalisation du travail demandé. Cette partie est composée de :

1. Les informations nécessaires à la réalisation de l'application
2. Les flux entre le client et le serveur
3. Maquettes

La quatrième partie du document concerne la documentation de la conception et de la réalisation et a pour objectif de faciliter la reprise du projet et la maintenance. Elle doit contenir les informations suivantes :

1. Spécificité logicielle

- a. Librairie x (description, version, source, installation, configuration...)
- 2. Conception
 - a. Modèle de la base de données
 - b. Tests fonctionnels prévus
- 3. Réalisations
- 4. Tests fonctionnels
- 5. Remarques et commentaires
 - a. Configuration particulière ou fonctionnement particulier
 - b. Problèmes résiduels et améliorations possibles
- 6. Conclusion
 - a. Commentaires personnels sur l'application
 - b. Commentaires personnels sur le développement
 - c. Autoévaluation
- 7. Bibliographie
- 8. Annexe

7.4 Un Web Summary

Ce document a pour objectif de présenter le projet de manière succincte.

7.5 La remise de la documentation

La documentation est remise selon les instructions du manuel i-FR, partie C, point 1.9.1 « Périmètre du rapport, Tips ».

Le manuel est disponible dans PkOrg sous : Documents → Documents pour tous : <http://manuel.ict-fr.ch>

8 Critères d'évaluation

<http://criteres.ict-fr.ch>

Compétences professionnelles globales (selon la grille d'appréciation)

Résultat et efficience

- 1. **Solution / Produit** (selon la grille d'appréciation)
- 2. **Critères spécifiques au projet**
 - 225 - Gestion des versions
 - 124 - Cas de test (programmation)
 - 164 - Codage : Traitement des erreurs
 - 166 - Style de codage ; Lisibilité du code
 - 250 - Séparation des couches
 - 119 - Utilité (application)
 - 165 - Implémentation de solutions
- 3. **Documentation** (selon la grille d'appréciation)
- 4. **Journal de travail** (selon la grille d'appréciation)

Présentation et entretien professionnel (selon la grille d'appréciation)

9 Délais

Le projet y compris la remise du projet aux experts se terminera le 2 juin 2025 à 17 heures. Le dernier délai pour la présentation du projet est fixé au 18 juin 2025 à 12 heures selon le planning ICT.

La documentation est déposée sur PkOrg dans les temps.