

ATTESTATION D'ETUDES COLLÉGIALES

Conception et programmation de sites Web
(NWE.OF - 2020)

Devis du projet

Groupe xxxx

Projet Web 2
582-41W-MA

90 heures

Session xxxx

Enseignant :

Nom Prénom

nom@cmaisonneuve.qc.ca

COLLÈGE DE MAISONNEUVE
3800 rue Sherbrooke Est
Montréal, Québec, H1X 2A2
Téléphone : 514 254-7131



Collège de
Maisonneuve

Devis du projet

Ce projet a pour objectif l'avancement et la bonification d'un projet Web transactionnel répondant aux demandes d'un client. En équipe de 3 à 5 personnes et en fonction d'un devis (ce document) présentant les exigences et des divers items de backlog qui vous seront exposés, vous devrez poursuivre le développement de l'application. Le site devra être compatible avec les diverses plateformes disponibles (Mac, PC, Android et iOS) et les principaux navigateurs (Edge, Chrome, Firefox et Safari).

Le suivi du projet et sa réalisation seront structurés en fonction de la méthodologie de gestion de projet AGILE/SCRUM, de l'utilisation de l'outil de gestion de code Git et du travail à distance.

Mandat

À partir du code source fourni et des items de *backlog*, vous devrez faire évoluer le projet Web « Vino ». À terme, ce site devrait permettre à des usagers de gérer leur(s) cellier(s) de vin. En répondant aux divers items de *backlog* à réaliser, chaque équipe produira une version bonifiée de Vino qui offrira différentes fonctionnalités, toutes en lien avec l'ajout de « valeur d'affaire », un des principes à la base de la gestion Agile.

L'objectif du projet est de permettre à un usager de gérer ses bouteilles de vin à partir d'une application Web. De base, avec l'application, un usager peut consulter les bouteilles qu'il possède dans son cellier et y modifier la quantité de bouteilles s'y trouvant. Il est implicite que l'application doit être pensée en fonction d'un utilisateur « dans l'action » avec un appareil mobile et non un utilisateur assis devant un écran large. L'approche de création *Mobile first* devra donc être utilisée et une attention particulière devra être portée sur l'utilisation simple de l'interface.

Le travail sera réalisé en 3 phases. La première vous permettra de vous familiariser avec le projet, les outils de gestion du code (git), la méthodologie de gestion du projet (Agile) et le code source. À cette étape, vous aurez à définir les éléments du design, restructurer le code source, régler quelques bugs et ajouter des fonctionnalités jugées essentielles.

Les deux autres phases, les sprints 2 et 3, serviront à poursuivre le développement du projet en fonction des items de *backlog*. Chaque équipe devra donc sélectionner et prioriser les éléments qui seront réalisés. Ainsi, trois sprints de 2 semaines permettront à chaque équipe de produire un prototype de plus en plus avancé du site « Vino ».

Un suivi journalier sera fait pour s'assurer que chaque membre de l'équipe fasse un travail équitable, et ce, autant en front-end qu'en backend. Chaque membre de l'équipe devra donc avoir livré une fonctionnalité complète : intégration HTML/CSS, programmation de l'interface (javascript) et programmation de la réponse serveur (php/mysql).

Phases détaillées :

Sprint 1 – Analyse et prise en main (30%) – Semaine 1 et 2

Ce sprint permet aux équipes de se familiariser avec le projet, le code source, les outils de travail et les tâches à réaliser.

Modélisation de la base de données et liste des opérations des modèles

Les équipes devront remettre une mise à jour des schémas de la base de données et faire évoluer ce schéma tout au long des 3 sprints du projet. De plus, une liste des opérations des modèles devra être produite et mise à jour pendant le développement du projet. Cette liste permettra de saisir rapidement les fonctions devant être produites dans les modèles (Le M, dans le patron de conception MVC). Par exemple, un modèle Bouteille pourra avoir Bouteille::ajouter(), Bouteille::effacer(), Bouteille::modifier(), Bouteille::getListe(), etc.

Conception graphique

Afin de clarifier le rendu proposé pour le Product Owner et les membres de l'équipe, un exemple des principales vues devra être remis. Minimale, la vue de la page principale de l'application et une vue comprenant des

formulaire (comme la page de création de compte) devraient être remises. Toute vue jugée importante peut être remise ou remplacer une vue demandée. Le format est au choix de l'équipe (HTML/CSS, psd, figma, sketch, adobeXD), mais une version standardisée (jpg, png ou pdf) devra être fournie.

Code source refactorisé, commenté, déployé et testé

En plus de se retrouver sur Github, le code source commenté doit être remis sur Léa et déployé sur Webdev. Celui-ci doit avoir été testé avant la présentation au Product owner. La présentation consiste à expliquer les éléments qui ont été ajoutés, corrigés ou faits durant le sprint. Chaque membre de l'équipe peut présenter les éléments qu'il ou qu'elle a réalisés ou bien un membre peut présenter l'ensemble des éléments. Lors de la présentation du produit réalisé, une revue de sprint devra être faite. Il s'agit simplement d'un retour sur le sprint, principalement sur les aspects techniques réalisés durant celui-ci. Finalement, les objectifs du prochain sprint seront exposés ainsi que les items de *backlog* choisis.

Voici un tableau synthèse de ces éléments :

Remise ou travail	Pondération (30 % total)	Échéance
La modélisation de la base de données et liste des opérations prévues dans les modèles Les principales vues (HTML/CSS, psd, figma, sketch, adobeXD)	10 % équipe/individuel ¹	Fin de la semaine 2
Le code source refactorisé (si nécessaire), intégrant la correction des bugs et l'ajout des fonctions de base requises (voir liste des story en annexe). Intégration des vues, si possible Code en ligne sur Webdev pour la présentation. Présentation en direct du produit réalisé au Product owner	20 % équipe/individuel	Fin de la semaine 2

Sprint 2 et 3 (35 %)

Pour chacun des deux derniers sprints, vous devez remettre :

- Le « *sprint backlog* » mis à jour et le suivi du sprint réalisé sur Scrumdesk/Clickup.
- La planification du « *sprint backlog* » du prochain sprint (pour le sprint 1). Celui-ci doit impérativement contenir les tâches non finalisées du « *sprint* » précédent.
- Le produit en ligne sur Webdev dont les tâches prioritaires du « *sprint backlog* » ont été implémentées, testées et validées.
- Schémas MCD et MPD mise à jour
- Liste des opérations des modèles mise à jour
- Comme pour le sprint 1, à la fin des deux sprints, les équipes doivent présenter leur produit au *Product owner*.

Remise ou travail	Pondération (35 % total)	Échéance
-------------------	--------------------------	----------

¹ Bien que l'évaluation se fasse en équipe, une note individuelle pourra être attribuée dans les cas d'une iniquité dans la production. Par exemple, une personne ne faisant pas sa partie du travail pourra être pénalisée.

Mise à jour des informations de base de données et des opérations sur les modèles	10 % équipe/individuel	Fin de chaque sprint
Le code source refactorisé et commenté Code en ligne sur Webdev pour la présentation. Présentation en directe du produit réalisé au Product owner	25 % équipe/individuel Notez que tous les membres de l'équipe doivent participer équitablement, et ce, autant en <i>front-end</i> , <i>backend</i> qu'en intégration/design.	Fin de chaque sprint

Précisions :

- Le nom de vos fonctions et variables doit être en français.
- Les commentaires doivent être dans un bon français.
- Le HTML/CSS doit respecter le balisage sémantique et être commenté.
- Vous devez commenter votre code (PHP et Javascript) en utilisant le standard du Département TIM².
- Aucun JavaScript dans les balises HTML (pas de onclick=`clique()`)
- Pas de alert(), prompt(), confirm()
- L'usage de jQuery est interdit (donc bye bye Bootstrap...).

Critères de correction :

Planification et suivi adéquat du projet selon les méthodes actuelles	5 %
Qualité du design et de l'adaptation des maquettes	10 %
Qualité, structure et précision de l'intégration et de la programmation des interfaces animées et interactives	20 %
Qualité du traitement et de l'intégration des éléments textuels et médiatiques	15%
Qualité, structure et optimisation adéquate des éléments de programmation serveur du site	20 %
Précision et rigueur des tests de validation	5 %
Optimisation adéquate des fichiers sources	5 %
Mise en ligne adéquate sur un serveur distant	5 %
Clarté de l'expression et de l'argumentation lors de la présentation	5 %
Qualité du travail d'équipe	5 %
Démonstration adéquate de l'autonomie de l'étudiant.	5 %
Total :	100 %

² https://docs.google.com/document/d/13113Kp_pBdoE70j-lcjBpApYWNW1DPz9meZn5NwaOT4/pub

Annexe

Type	Titre du « user story »	Description
bug	La quantité affichée ne change pas dans le cellier (Ajouter, Boire)	Après avoir appuyé sur le bouton Ajouter ou Boire, la quantité dans l'interface ne se met pas à jour, mais dans la DB oui. Il faut recharger la page pour voir la modification.
bug	L'importation des produits ne conserve pas le prix de la bouteille correctement.	Le prix est tronqué à l'entier. Les valeurs décimales ne sont pas conservées.
feature	Modification d'une bouteille dans le cellier	Le bouton modifier n'a pas de fonctionnalité. En cliquant sur le bouton, il faudrait pouvoir modifier les informations de la bouteille qui est dans le cellier, sans modifier la bouteille dans la table d'importation, ni pour les autres usagers.
improvement	Refaire l'interface graphique	Revoir l'interface graphique pour la rendre conviviale, adaptative, ergonomique et moderne.
improvement	Revoir la structure des données pour avoir au moins 1 cellier par usager	Dans le prototype de base, la table cellier représente 1 seul cellier. Le prototype fonctionne sur la base de 1 seul cellier et 1 seul usager.
user_story	En tant qu'utilisateur, je peux gérer mon cellier personnel (ajout, modification, retrait de bouteille)	
user_story	En tant qu'administrateur, je peux importer les bouteilles du site de la SAQ	
user_story	En tant qu'utilisateur, je peux me créer un compte et le gérer	
user_story	En tant qu'administrateur, je peux gérer la liste des usagers	
user_story	En tant qu'utilisateur, je peux trier mon cellier	Tri par nom, par type, par quantité, par pays, par millésime, etc
user_story	En tant qu'utilisateur, je peux faire des recherches dans mon cellier	par nom, par type, par quantité, par pays, par millésime, etc
user_story	En tant qu'utilisateur, je peux créer plusieurs celliers	
user_story	En tant qu'utilisateur, je peux partager sur les réseaux sociaux les bouteilles que j'ai achetées ou bu.	
user_story	En tant qu'utilisateur, je peux ajouter des bouteilles "non listées" dans mon cellier	

Type	Titre du « user story »	Description
user_story	En tant qu'administrateur, je peux sortir des statistiques sur les usagers et leurs bouteilles	<p>Connaitre le nombre d'usager, le nombre de cellier, le nombre de cellier par usager, le nombre de bouteille par cellier et par usager.</p> <p>Le nombre de bouteille bu et partager dans un temps donné.</p> <p>Le nombre de bouteille ajouté dans un temps donné.</p> <p>Le nombre de nouvel usager</p> <p>La valeur total des bouteilles (Tous, par usager, par cellier)</p>
user_story	En tant qu'administrateur, je peux modifier les informations du catalogue	Je peux corriger les données erronées du catalogue importé
user_story	En tant qu'usager, je peux indiquer des erreurs dans le catalogue aux administrateurs	
user_story	En tant qu'usager, je peux créer une liste d'achat	
user_story	En tant qu'usager, je peux laisser des notes de dégustation sur les bouteilles que j'ai consommé.	