


1 INFORMATIONS GENERALES

Élève :	Nom:	Prénom:
Lieu de travail :	ETML / Avenue de Valmont 28b, 1010 Lausanne	
Client	Nom: Carrel / Melly	Prénom: XAVIER / JONATHAN
	 xavier.carrel@eduvaud.ch / jonathan.melly@eduvaud.ch	
Dates de réalisation :	2 ^{ème} trimestre	
Temps total :	24 périodes	

2 PROCÉDURE

- Tous les apprentis réalisent le projet sur la base d'un cahier des charges.
- Le cahier des charges est présenté, commenté et discuté en classe.
- Les apprentis sont entièrement responsables de la sécurité et sauvegarde de leurs données.
- En cas de problèmes graves, les apprentis avertissent le client au plus vite.
- Les apprentis ont la possibilité d'obtenir de l'aide externe, mais ils doivent le mentionner.
- Les informations utiles à l'évaluation de ce projet sont disponibles au chapitre 8.

3 TITRE

BitRuisseau

4 SUJET

Médiathèque audio/vidéo partagée avec un protocole P2P

5 MATÉRIEL ET LOGICIEL À DISPOSITION

- Un PC ETML
- Accès à Internet

6 PRÉREQUIS

Modules de programmation de base ainsi que les modules 323 et 321 (en parallèle)

7 CAHIER DES CHARGES

7.1 Gestion de projet

1. Marketplace : <https://apps.pm2etml.ch/jobDefinitions/123>
2. La planification est à faire selon les instructions spécifiques de votre chef de projet.
3. Un journal de travail devra être rendu. Il sera généré automatiquement à partir des commits dans votre repo (git log). Un outil de formatage vous sera fourni. Pour que le résultat soit acceptable :
 - Les commits suivent la convention de nommage « type(scope) :effet ».
 - La première ligne de la description est composée uniquement de deux éléments entre crochets carrés :
 1. L'indication du temps passé à la réalisation du contenu du commit, en minutes
 2. L'état de la tâche : « WIP » ou « DONE »

Exemples pour une tâche faite en 1h45 sur deux jours:

Summary :

feat(Médiathèque) : Permet de supprimer un contenu

Description :

[30][WIP]

Difficultés avec les options de recherche de classe File

Summary :

feat(Médiathèque) : Permet de supprimer un contenu

Description :

[75][DONE]

Problème résolu avec l'option 'readonly'

Ajout de try/catch partout

- Si la tâche est liée à une User Story, la seconde ligne de la description donne la référence à cette dernière

7.2 Qualité

1. Réaliser un programme informatique de qualité
 - Organisé (namespace, classes, commit log,...)
 - Compacté (pas de copié/collé,...)
 - Optimisé (utilisation de structures adaptées)
 - Testé (unitairement et manuellement selon scénarios définis)
 - Commenté judicieusement
 - Complet (code, script DB, maquettes PDF, exécutable, ...)
2. Prouver que vous êtes digne de confiance lorsqu'on vous confie un projet
 - Vous tenez votre journal de travail à jour
 - Vous êtes pro-actif
 - **Posez des questions** au client
 - Faites des démonstrations
 - Faites des propositions
 - Parallélisez vos tâches quand c'est possible (i.e : ne restez pas à ne rien faire si vous êtes en attente de quelque chose)
3. Mettre en œuvre les bonnes pratiques apprises en classe.

7.3 Éléments requis (dans l'ordre)

- a. Analyse UX/UI du MediaPlayer standalone. Elle doit couvrir les User Stories (US) ci-dessous. (elles sont toutes exprimées du point de vue de l'utilisateur unique).
 - i. Je veux définir un dossier dans lequel mes fichiers audio sont enregistrés.
 - ii. Je veux voir dans le MediaPlayer la liste de tous mes fichiers audio (liste personnelle). Chaque fichier est présenté avec les attributs suivants :
 - Titre*
 - Artiste*
 - Année*
 - Durée*
 - Taille*
 - Featuring
- b. Implémentation du MediaPlayer standalone
- c. Définition d'un protocole
 - i. Les messages du protocole permettent de
 1. Demander sur le réseau quels sont les médiathèques qui sont opérationnelles
 2. Annoncer une médiathèque opérationnelle (au démarrage et en réponse aux demandes)
 3. Demander son catalogue à une médiathèque précise
 4. Publier son catalogue (en réponse à la demande)
 5. Demander un fragment de media à une médiathèque précise
 6. Fournir un fragment de media à une médiathèque (en réponse)
 - ii. Publier le protocole dans le canal Teams sous forme de fichier(s) .cs
- d. Analyse UX/UI du MediaPlayer connecté, qui couvre les US :
 - i. Je veux voir la liste des médiathèques disponibles à travers le réseau.
 - ii. Je veux consulter le contenu d'une médiathèque distante.
 - iii. Je veux importer un son d'une médiathèque distante dans ma médiathèque, afin de pouvoir l'écouter même quand la médiathèque distante est arrêtée.
- e. Implémentation du MediaPlayer connecté

7.3.1 Contraintes techniques

- Préférer le fonctionnel / LINQ aux boucles traditionnelles
 - Tout en privilégiant la lisibilité

7.4 Livrables

Une livraison consiste en une release GitHub, à laquelle le journal de travail est joint au format PDF. Un binaire portable est aussi joint à la release.

Lorsqu'une livraison est prête, un lien sur la release est envoyé au destinataire.

8 Évaluation

1. Auto-évaluation challengée par le client basé sur des éléments observables.
2. Jokers.

Vous recevez trois Jokers en début de projet. Ces Jokers peuvent vous être retirés en cours de projet sur observation d'un comportement ou d'une pratique inacceptable, tels que :

- Du code est commité dans votre repo et vous n'êtes pas en mesure de l'expliquer de manière précise et convaincante.
- Abus des libertés qui vous ont données : pause trop longue, discussion extra professionnelles, inactivité, usage non professionnel des outils mis à votre disposition, ...
- Recours inadapté et répété à l'IA
- Comportement dérangeant

À noter : cette liste n'est pas exhaustive. Il est donc possible qu'un joker vous soit retiré pour une raison qui n'est pas listée ci-dessus.