


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 :	1 ^{er} trimestre	
Temps total :	32 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 ».
 - Si la tâche est liée à une User Story, la seconde ligne de la description donne la référence à cette dernière
 - La première ligne de la description est composée de trois éléments entre crochets carrés :
 1. L'indication du temps passé à la réalisation du contenu du commit
 2. L'état de la tâche : « WIP » ou « DONE »
 3. Une référence à la User Story correspondante, s'il y a en a une

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

Summary :

```
feat(Médiathèque) : Permet de supprimer un contenu
```

Description :

```
30m-WIP-#4
```

```
Difficultés avec les options de recherche de classe File
```

Summary :

```
feat(Médiathèque) : Permet de supprimer un contenu
```

Description :

```
1h15-DONE-#4
```

```
(blabla)
```

- Les activités qui ne produisent pas de contenu (écoute de consigne, théorie, lecture d'article, visionnage de tuto vidéo, réunion, ...) font également l'objet d'un commit. Dans un cas pareil, le contenu sera un fichier « notes.md » situé à la racine qui regroupera, par date, les éléments utiles.

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
 - Journal de travail à jour
 - Pro-activité
 - **Poser des questions** au client
 - Faire des démonstrations
 - Utiliser un système de versioning de code (GIT)

7.3 Éléments requis (dans l'ordre)

- a. Analyse UX/UI Mediaplayer
 - i. Gérer le contenu de la médiathèque (dossier local)
 - ii. Afficher la liste des contenus disponibles dans la communauté
- b. Définition d'un protocole commun à toute la classe
 - i. Messages
 1. Demander sur le réseau quels sont les médiathèques qui sont opérationnelles
 2. S'annoncer opérationnelle
 3. Demander son catalogue à une médiathèque
 4. Publier son catalogue
 5. Demander un fragment de media à une médiathèque
 6. Fournir un fragment de media à une médiathèque
 - ii. Format
 - iii. Le protocole doit être conçu pour supporter des évolutions.
- c. Implémentation des catalogues (suite à partie a et b)
- d. Analyse UX/UI
 - i. Echanger des contenus
 - ii. Jouer des contenus
- e. Implémentation de l'ajout de contenu

7.4 Éléments additionnels (si le temps le permet)

- f. Déploiement extra-muros (sur Internet)
- g. Recherche

7.4.1 Contraintes techniques

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

7.5 Livrables

Un lien vers une release Github v1.0.0, **d'un dépôt github commençant par votre prénom suivi d'un tiret (exemple bob-monSuperDepot)**, contenant tous les documents de travail ainsi qu'un bref rapport PDF avec au moins les chapitres suivants (si le rapport n'est pas dans les sources, l'attacher directement à la release) :

- a. Introduction
- b. Planification
- c. Journal de travail
- d. Rapport de tests (unitaires et manuels)
- e. État des lieux et conclusion
- f. Chapitre explicatif de l'usage fait de l'IA dans ce projet

8 Évaluation

1. Auto-évaluation challengée par le client basé sur des éléments observables.
2. Le recours à des outils en ligne d'intelligence artificielle (ex. : Chat GPT) doit être mentionné et ne peut servir que d'inspiration à la réalisation. Chaque développeur doit être à tout moment en mesure d'expliquer le code de manière précise et convaincante.
En cas d'abus, l'évaluation du projet en tiendra compte.