

# Rapport d'activité – Projet All Together

Date : 26 septembre 2025

## Avancement de la journée

- **Pablo :**
  - installation et compréhension de docker
  - début du front avec installation de [node.js](#) pour pouvoir commencer à créer l'interface et première page web générée

Prénom, Nom : Équipe :							
Niveau / Qualité de mobilisation: Compétences Mobilisées par le projet (simplifiées / adaptées)*	Pas mobilisé	Découverte	Notions	Application	Maîtrise	Expertise	
<b>Application / Démo</b>							
C02.1	Complexité : Résoudre un problème complexe en mobilisant les concepts, méthodes et outils informatiques et mathématiques adaptés. Schema des Données & BackEnd : Concevoir et modéliser le stockage de données	Application statique.	API sample sans identification utilisateur.	Appli avec authentification et API protégées.	Appli avec schéma des données avancées et/ fonctionnalités originales (son, cartes, videos...).	Appli déployé.	Efforts au niveau moteur de recherche, schéma complexe, etc. ....
C03.2		UML ou schéma papier simple.	UML réaliste.	Schema implementé et BD instansie.	BD optimisé.		Front poussé avec composants complexes (ex animations, site responsives) et/ou reflection et re-conception après retours des utilisateurs.
C04.8	FrontEnd : système utile, avec interface cohérent et utilisable, construite avec une approche centrée-utilisateur (conception, évaluation)	Prototype papier ou page Figma simple.	Figma (ou autre prototype) site complete.	Une front basic mais qui fonctionne.	Un front complete, testé par d'utilisateurs.		
<b>Développement / Code</b>							
C04.10	Qualité Code : Code stable, et qui répond aux besoins demandés [...] Test : Tester un logiciel : concevoir, planifier et exécuter un plan de validation logiciel [...]	Code qui se n'exécute pas.	Code qui tourne avec quelques bugs.	Code complexe qui tourne sans bugs.	Code avec des bonnes pratiques respectés (structure, genericité, noms et contenu des fonctions, ...), et bien commenté.	Code optimisé et possibilité de communication avec des services extérieurs.	
C04.7		Code pas testé.	Test partielle manuel et/ou avec des utilisateurs.	Tests unitaires faites.	Cahier des tests.	Tests end-to-end (e2e), tests TNR.	
<b>Rapport</b>							
C04.3 C06.1	Cahier : Traduire des fonctionnalités attendues en cahiers des charges. [...]. Avoir une réflexion sur le cahier initiale les évolutions. Veille : Tenir en compte de concurrence (veille scientifique / technologique).	Liste des fonctionnalités haute niveau, pas de planification.	Cahier des charges non-exhaustive, planification pas réfléchi.	Cahier des charges exhaustive, sans planification ou planification pas réfléchi. Idée de ce qui existe dans le marché (concurrence).	Cahier des charges exhaustive avec planification, connaissance des concurrences.	Cahier des charges exhaustive avec planification détaillée, avec perspectives d'évolution dans le cahier des charges, étude des concurrences.	
C14.1 C04.4	Réflexion : Développer une pratique réflexive sur son projet. Argumenter pour ces décisions: Bien argumenter sur la pertinence du projet, les choix technologique et fonctionnels, l'organisation du travail. Donner des références/citations des sources utilisés.	Pas de réflexion personnel et/ou pas de réflexion en groupe.	Reflexion superficiel sur les choix (technologiques, conception, structure).	Réflexion approfondis sur les choix (techno, conception, solution), sur la demarche (travail en équipe, gestion, planification), sur le cahier des charges (honnêteté). Bien citer les ressources utilisés.	Justifier les choix du projet et personnel avec pertinence, esprit critique, auto-evaluation.	Etre capable de faire une Retour d'Expérience (REX) sur chaque partie du projet (technologie, gestion, front/back, groupe) et être capable d'attirer des leçons des le prochaine projet.	
<b>Communication</b>							
C11.2	Communication: Communiquer et convaincre en s'adaptant aux objectifs et contraintes [...]	Équipe pas préparé.	Avoir fait effort de preparation, mais presentation dehors les limites / contraintes données.	Être capable de présenter le projet dans les contraintes données.	Être capable de convaincre de la maîtrise de son projet (son valuer, les choix faites, etc).	Être capable de convaincre de la maîtrise de son projet et aussi son propre expertise sur le sujet.	
<b>Travail en équipe</b>							
C04.2	Coordination et planification: Mettre en oeuvre une méthodologie de projet, planifier votre travail [...]. Communiquer régulièrement la progrès auprès des acteurs (ex professeurs). Communiquer la progression à l'écrit.	Pas present en cours, pas de communication.	Communiquer dans la classe.	Communiquer dans la classe et mini-rapports envoyés.	Communiquer sur l'avancé et identification des points bloquants.	Communiquer clairement (en verbal + mini-rapport) sur l'avancement et la répartition des tâches à chaque séance. Mettre en place de cerémonie de type agile et l'expliquer.	
C04.9	DevOps: Gérer le cycle de vie logiciel tout au long des phases de planification, de développement, [...] selon les pratiques DevOps, et mettre en place des architectures orientées services	Absence des outils de gestion (code, tâches).	Outils initiés mais peu utilisés.	Outils de gestion code+projet mis-en-place et utilisés régulièrement.	Avoir un git propre (branches, ...) et connection avec Jira et tâches, mise en place d'un Docker.	Avoir une CI/CD - Chaîne d'intégration Chaîne de Déploiement complet.	

- **Paolo :**
  - Installation Docker
  - Création des Packages repository, controller, service
  - Création des classes associées
  - Risques et prochaines étapes du back:
    1. Configurer Spring Security et JWT :
    2. Ajouter les endpoints manquants :
    3. Sécuriser les endpoints :

## 4. Implémenter la logique anti-doublons :

Auto évaluation:

Niveau / Qualité de mobilisation:		Pas mobilisé	Découverte	Notions	Application	Maîtrise	Expertise
<b>Compétences Mobilisées par le projet (simplifiées / adaptées)*</b>							
<b>Application / Démo</b>							
C02.1	Complexité : Résoudre un problème complexe en mobilisant les concepts, méthodes et outils informatiques et mathématiques adaptés.	✗	Application statique.	API sample sans identification utilisateur.	Appli avec authentification et API protégées.	Appli avec schéma des données avancés et fonctionnalités originales (son, cartes, vidéos...).	Appli déployé.
C03.2	Schema des Données à BackEnd : Concevoir et modéliser le stockage de données.		UML ou schema papier simple.	UML réaliste.	Schema implémenté et BD instancée.	BD optimisé.	Efforts au niveau moteur de recherche, schema complexe, etc.
C04.8	FrontEnd : système utile, avec interface cohérent et utilisable, construite avec une approche centrée-utilisateur (conception, évaluation).	✗	Prototype papier ou cagui Figma simple.	Figma (ou autre prototype) site complete.	Une front basic mais qui fonctionne.	Un front complete, testé par d'utilisateurs.	Front poussé avec composants complexes (ex animations, site responsive) et/ou reflection et re-conception après retours des utilisateurs.
<b>Développement / Code</b>							
C04.10	Qualité Code : Code stable, et qui répond aux besoins demandés [...] Test : Tester un logiciel : concevoir, planifier et exécuter un plan de validation logiciel [...]	•	Code qui se n'exécute pas.	Code qui tourne avec quelques bugs. Test partiel manuel et/ou avec des utilisateurs.	Code complexe qui tourne sans bugs.	Code avec des bonnes pratiques respectées structure, générée, nommée et contenue des fonctions.	Code optimisé et possibilité de communication avec des services extérieurs.
C04.7			Code pas testé.		Tests unitaires faites.	Cahier des tests.	Tests end-to-end (e2e), tests TNR.
<b>Rapport</b>							
C04.3	Cahier : Traduire des fonctionnalités attendues en cahiers des charges, [...]. Avoir une réflexion sur le cahier initial les évolutions. Veille : Tenir en compte de concurrence (veille scientifique / technologique).		Liste des fonctionnalités haute niveau, pas de planification.	Cahier des charges non-exhaustive, planification pas réfléchi.	Cahier des charges exhaustive, sans planification ou planification pas réfléchi. Idée de ce qui existe dans le marché (concurrence).	Cahier des charges exhaustive avec planification détaillée, avec perspectives d'évolution et le cahier des charges, étude des concurrences.	Cahier des charges exhaustive avec planification détaillée, avec perspectives d'évolution et le cahier des charges, étude des concurrences.
C04.1	Reflexion : Développer une pratique réflexive sur son projet. Argumenter pour ces décisions: Bien argumenter sur la pertinence du projet, les choix technologiques et fonctionnels, l'organisation du travail. Donner des références/citations des sources utilisées.		Pas de réflexion personnel et/ou pas de réflexion en groupe.	Reflexion superficielle sur les choix (technologiques, conception, structure).	Reflexion approfondie sur les choix (techno, conception, solution), sur la démarche (travail en équipe, gestion, planification), sur le cahier des charges (honnêteté). Bien citer les ressources utilisées.	Justifier les choix du projet et personnel avec pertinence, esprit critique, évaluation.	Etre capable de faire une Retour d'Expérience (REX) sur chaque partie du projet (technologie, gestion, front/back, groupe) et être capable d'attirer des leçons dès le prochain projet.
<b>Communication</b>							
C11.2	Communication: Communiquer et convaincre en s'adaptant aux objectifs et contraintes [...]	✗	Equipe pas préparé.	Avoir fait effort de preparation, mais presentation dehors les limites / contraintes données.	Etre capable de présenter le projet dans les contraintes données.	Etre capable de convaincre de la maîtrise de son projet (son valeur, les choix faites, etc).	Etre capable de convaincre de la maîtrise de son projet et aussi son propre expertise sur le sujet.
<b>Travail en équipe</b>							
C04.2	Coordination et planification: Mettre en œuvre une méthodologie de projet, planifier votre travail [...] Communiquer régulièrement la progrès auprès des acteurs (ex professeurs). Communiquer la progression à l'écrit.		Pas present en cours, pas de communication	Communiquer dans la classe.	Communiquer dans la classe et mini-rapports envoyés.	Communiquer sur l'avancement et la répartition des tâches à chaque séance. Mettre-en-place de ceremonie de type agile et l'expliquer.	Communiquer clairement (en verbal + mini-rapport) sur l'avancement et la répartition des tâches à chaque séance. Mettre-en-place de ceremonie de type agile et l'expliquer.
C04.9	DevOps: Gérer le cycle de vie logiciel tout au long des phases de planification, de développement, [...] selon les pratiques DevOps, et mettre en place des architectures orientées services		Absence des outils de gestion (code, tâches).	Outils initiés mais peu utilisés.	Outils de gestion code+projet mis-en-place et utilisés régulièrement.	Avoir un git propre (branches, ...) et connecté avec Jira et tâches, mise en place d'un Docker.	Avoir une CI/CD - Chaîne d'intégration Chaîne de Déploiement complet.

Jira:

A FAIRE 13

page d'accueil liste ville

🕒 26 sept. 2025

☒ AL-11

⬆

interface de connexion

📅 3 oct. 2025

☒ AL-9

⬆

formulaire de création de compte

📅 3 oct. 2025

☒

⬆

+ Créer

EN COURS 4

Apprendre le react

☒ AL-21

⬆

R

Ajout planning détaillé au cahier des charges

🕒 26 sept. 2025

☒ AL-24

=

R

Sécuriser les endpoints

☒ AL-27

⬆

PB

Implémenter la logique

☒

⬆

+ Créer

TERMINÉ(E) 8

Apprendre le Spring boot

☒ AL-22

✓

⬆

PB

trouver les outils nécessaires

📅 18 sept. 2025

☒ AL-5

✓

=

Diagramme UML

📅 18 sept. 2025

☒ AL-6

✓

⬆

+ Créer