

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 Mobilisés par le projet (simplifiés / adaptés)*	Pas mobilisé	Découverte	Notions	Application	Maitrise	Expertise		
Application / Démo								
C02.1	Complexité : Résoudre un problème complexe en mobilisant les concepts, méthodes et outils informatiques et mathématiques adaptées.	Application statique.	API simple sans identification utilisateur.	Appli avec authentification et API protégés.	Appli avec schéma des données avancés et/ou fonctionnalités originales (son, cartes, vidéos...).	Appli déployé.		
C03.2	Schéma des Données & BackEnd : Concevoir et modéliser le stockage de données	UML ou schema papier simple.	UML réalisé.	Schema implementé et BD instanciée.	BD optimisée.	Efforts au niveau moteur de recherche, schéma complexe, etc ...		
C04.8	FrontEnd : système utile, avec interface cohérente et utilisable, construite avec une approche centrée-utilisateur (conception, évaluation)	Prototype papier ou page Figma simple.	Figma (ou autre prototype) site complète.	Une front basic mais qui fonctionne.	Un front complète, testé par d'utilisateurs.	Front poussé avec composants complexes (ex animations, site responsives) et/ou réflexion et re-conception après retours des utilisateurs.		
Développement / Code								
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 respectées (structure, généricité, noms et contenu des fonctions, ...), et bien commenté.	Code optimisé et possibilité de communication avec des services extérieurs.		
C04.7		Code pas testé.	Test partiel manuel et/ou avec des utilisateurs.	Tests unitaires faites.	Cahier des tests.	Tests end-to-end (e2e), tests TNR.		
Rapport								
C04.3 C06.1	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 refléchi.	Cahier des charges exhaustive, sans planification ou planification pas refléchi. 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 utilisées.	Pas de réflexion personnel et/ou pas de réflexion en groupe.	Réflexion superficielle sur les choix (techno, conception, solution), sur la demande (travail en équipe, gestion, planification), sur le cahier des charges (honefête). Bien citer les ressources utilisées.	Réflexion approfondie sur les choix (techno, conception, solution), sur la demande (travail en équipe, gestion, planification), sur le cahier des charges (honefête). Bien citer les ressources utilisées.	Justifier les choix du projet et personnel avec pertinence, esprit critique, auto-évaluation.	Etre capable de faire une Retour d'Expérience (REX) sur chaque partie du projet (technologie, gestion, front/back, groupe) et être capable d'attrier des leçons dès le prochain projet.		
Communication								
C11.2	Communication: Communiquer et convaincre en s'adaptant aux objectifs et contraintes [...]	Y	Équipe pas préparé.	Avoir fait effort de préparation, mais présentation 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 valeur, les choix faits, etc).	Être capable de convaincre de la maîtrise de son projet et aussi son propre expertise sur le sujet.	
Travail en équipe								
C04.2	Coordination et planification: Mettre en œuvre une méthodologie de projet, planifier votre travail [...]. Communiquer régulièrement la progression auprès des acteurs (ex professeurs). Communiquer la progression à l'écrit.	Pas présent 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'avancé et répartition des tâches à chaque séance. Mettre-en-sace de cérémonie 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+tâches 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: Compétences Mobilisées par le projet (simplifiées / adaptées)*	Pas mobilisé	Découverte	Notions	Application	Maitrise	Expertise	
Application / Démo							
C02.1	Complexité : Résoudre un problème complexe en mobilisant les concepts, méthodes et outils informatiques et mathématiques adaptés.	X	Application statique.	API simple sans identification utilisateur	Appli avec authentification et API protégés.	Appli avec schéma des données avancées et/ou fonctionnalités originales (son, cartes, vidéos...)	
C03.2	Schema des Données & BackEnd : Concevoir et modéliser le stockage de données		UML ou schema papier simple.	UML réaliste.	Schema implementé, MBD instances, BD optimisé.	Appli déployé. Efforts au niveau moteur de recherche, schéma complexe, etc ...	
C04.8	FrontEnd : système utile, avec interface cohérente et utilisable, construite avec une approche centre-utilisateur (conception, évaluation)	X	Prototype papier ou page Figma simple.	Figma (ou autre prototype) site complète.	Une front basic mais qui fonctionne.	Fondé sur des composants complexes (ex animations, site responsives) et/ou réflexion et re-conception après retours des utilisateurs.	
Développement / Code							
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 ne s'exécute pas.	Code qui tourne avec quelques bugs.	Code complexe qui tourne sans bugs.	Code avec des bonnes pratiques respectées structure, généricité, nommage, etc ...	Code optimisé et possibilité de communication avec des services extérieurs.	
C04.7	Rapport	Test partie manuel et/ou avec des utilisateurs.	Code pas testé	Tests unitaires faites.	Cahier des tests.	Tests end-to-end (e2e), tests TNR.	
C04.3 C06.1	Cahier : Traduire des fonctionnalités attendues en cahiers des charges. [...], Avoir une réflexion sur le cahier initialise les évolutions. Veiller à faire le compte de concurrence (ville scientifique / technologique).	Liste des fonctionnalités haute niveau; pas de planification	Cahier des charges non-exhaustif, planification pas refléchi.	Cahier des charges exhaustive, planification ou prévision de ce qui existe dans le marché (concurrence).	Cahier des charges exhaustive avec planification et connaissance des concurrences.	Cahier des charges exhaustif avec planification détaillée, avec prévisions d'évolution et/ou le cahier des charges, étude des concurrences.	
C14.1 C14.4	Communication	Reflexion : Développer une pratique réflexive sur son projet. Argumenter pour ces décisions : Bien argumenter sur la pertinence du projet, les choix technologique et fonctionnel, l'organisation du travail. Donner des références/citations des sources utilisées.	Pas de réflexion personne et/ou pas de réflexion en groupe.	Réflexion superficielle sur les choix (technologiques, conception, structure).	Reflexion approfondie sur les choix (terminologie, solution), sur la démarche (travail en équipe, gestion, planification), sur le cahier des charges (honnêteté). Bâtir clair les ressources utilisées.	Etre capable de faire une Retour d'Expérience (REX) sur chaque partie du projet (technologie, gestion, frontback, groupe) et être capable d'attirer des leçons denses du prochain projet.	
C11.2	Travail en équipe	Communication: Communiquer et convaincre en s'adaptant aux objectifs et contraintes [...]	X	Équipe pas préparé.	Avoir fait effort de préparation, mais présentation dehors le limites / contraintes données.	Etre capable de présenter le projet dans les contraintes données.	Etre capable de communiquer la maitrise de son projet (son valeur, les choix faits, etc).
C04.2	Coordination et planification: Mettre en œuvre une coordination de projet, planifier votre travail [...] Communiquer régulièrement le progrès auprès des acteurs (ex professeurs). Communiquer la progression de l'avancement du travail 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	Pas présent en cours, pas de communication.	Communiquer dans la classe.	Communiquer dans la classe et mini-rapports envoyés.	Communiquer sur l'avancement et identification des points bloquants.	Communiquer régulièrement (en verbal + mini-rapport) sur l'avancement et la répartition des tâches à chaque séance. Mettre-en-place de cérémonie de type agile et l'expliquer.	
C04.9		Absence des outils de gestion (code, tâches).	Outils initialement peu utilisés.	Outils de gestion code+projet mis-en-place et utilisés régulièrement.	Avoir un git propre (branches, ...) et correction avec Jira et tâches, mise en place d'un Docker.	Avoir une CI/CD - Chaîne d'intégration Chaîne de Déploiement complet.	

Jira:

The Jira board displays the following tasks across three columns:

- A FAIRE (13 tasks):**
 - page d'accueil liste ville (Due: 26 sept. 2025, Assigned: AL-11)
 - interface de connection (Due: 3 oct. 2025, Assigned: AL-9)
 - formulaire de création de compte (Due: 3 oct. 2025)
- EN COURS (4 tasks):**
 - Apprendre le react (Assigned: AL-21, Priority: R)
 - Ajout planning détaillé au cahier des charges (Due: 26 sept. 2025, Assigned: AL-24)
 - Sécuriser les endpoints (Assigned: AL-27, Priority: PB)
 - Implémenter la logique (Assigned: AL-28)
- TERMINÉ(E) (8 tasks):**
 - Apprendre le Spring boot (Assigned: AL-22, Priority: PB)
 - trouver les outils nécessaires (Due: 18 sept. 2025, Assigned: AL-5)
 - Diagramme UML (Due: 18 sept. 2025, Assigned: AL-6)