

Projet #Mobile Musician

Application Mobile Musicians - Projet sur 5 jours en équipe

HETIC

Sommaire	
Contexte	2
Présentation	2
Contraintes	2
Libertés	2
Objectifs	3
Rendu final	3
Evaluations	4
Compétences à valider	4

Contexte

Vous êtes embauché en tant que développeur pour travailler sur une application mobile **visant à mettre en relation des musiciens**.

L'objectif est de permettre aux musiciens de s'inscrire, se connecter, envoyer des messages, créer et participer à des événements musicaux.

Une carte type google maps permettra de repérer facilement les Jams dans votre région (géolocalisation).

Le projet inclut la conception du **back-end** pour gérer les utilisateurs, les messages et les événements, ainsi que le développement de l'application mobile pour une expérience utilisateur fluide et interactive.

Contraintes

- **Travail en groupe de 5 maximum**

Le projet doit être réalisé en équipe de 5 développeurs maximum. Une répartition claire des tâches doit être établie pour assurer une gestion efficace du projet, avec une bonne communication entre les membres de l'équipe (un peu d'AGILITE).

- **Technologie mobile**

L'application mobile peut être développée avec **React Native (conseillé)** ou une autre technologie mobile hybride ou native selon les préférences et les compétences de l'équipe.

Les choix doivent être justifiés et permettre une intégration fluide avec le back-end.

Vous pouvez (conseillé) utiliser des framework comme Tamangui.dev pour la création rapide des composants stylisés en React Native

- **Technologie back-end**

Le back-end du projet doit être développé avec Symfony ou FastAPI. Ces technologies sont choisies pour leur robustesse et leur flexibilité dans la gestion des utilisateurs, des messages, et des événements.

Pensez à utiliser des ORM pour les données comme Doctrine avec Symfony.

- **Normes de sécurité**

Des normes de sécurité strictes doivent être appliquées tout au long du projet. Cela inclut l'utilisation de pratiques sécurisées pour l'authentification et l'autorisation des utilisateurs (par exemple, JWT pour la gestion des sessions), la validation des entrées, la gestion des données sensibles et la protection contre les attaques courantes telles que les injections SQL, les attaques XSS et CSRF.

- **Documentation complète**

Chaque étape du projet, du développement à la mise en production, doit être documentée. Pensez à commenter votre code également.

- **Utilisation de GitHub ou GitLab**

Le code source du projet doit être versionné sur GitHub ou GitLab. Une gestion rigoureuse des branches (ex : GitFlow) doit être mise en place. Les commits doivent être bien rédigés.

- **Respect des délais**

Le respect des jalons définis et des dates de livraison est impératif !

Chaque membre de l'équipe doit gérer son temps et ses tâches de manière à respecter les objectifs du projet, avec un suivi régulier de l'avancement (réunions de synchronisation, gestion de projet avec des outils comme Trello, Notion ou Jira). Pensez à définir un Lead Dev.

- **Containerisée**

L'application peut être containerisée à l'aide de **Docker**. Les services backend et base de données pourront être encapsulés dans un conteneur distinct pour assurer une isolation des processus et faciliter le déploiement. Si vous utilisez React Native pensez à ne pas mettre cette partie dans un conteneur (plus technique pour les **Live Reload** dans la phase de développement).

- **Figma**

Vous devez utiliser Figma pour faire les maquettes de l'application mobile. Vous pouvez vous inspirer des [Maquettes exemples](#)

Libertés

- **Choix de la base de données**

Vous êtes libres de choisir la base de données la plus adaptée à vos besoins, que ce soit **MySQL**, **PostgreSQL**, ou **MongoDB**.

- **Utilisation d'API existantes**

Vous pouvez intégrer des API tierces pour enrichir l'application, comme API Platform pour faciliter la gestion des entités ou utiliser des services comme Twilio pour les notifications. Vous pouvez également utiliser des API externes pour la géolocalisation des musiciens ou la gestion des événements.

- **GitFlow**

L'utilisation d'un GitFlow **est recommandée** pour la gestion de version. Chaque fonctionnalité doit être développée dans une branche dédiée, et les changements doivent être fusionnés dans la branche principale (main) après validation (Pensez à le faire avec le Lead de votre équipe). Des Pull Requests peuvent être utilisés pour la revue du code et assurer la qualité et la cohérence du projet (si vous pensez avoir le temps).

Objectifs

- **Prototype fonctionnel**

Développer une application mobile permettant aux musiciens de se connecter, échanger des messages, créer et participer à des événements musicaux, même si l'application n'est pas terminée pensez à écrire les étapes suivantes pour continuer le projet, c'est un prototype !

- **Conception du système**

Rédiger un dossier de conception de 5/10 pages maximum, incluant :

- Modélisation des données.
- Description des API utilisées.
- Spécifications fonctionnelles
- Spécifications techniques

- **Tests et qualité**

Effectuer des tests pour assurer la fonctionnalité et la stabilité de l'application, en incluant des tests unitaires et fonctionnels.

- **Déploiement**

- Déployer le backend sur AlwaysData en utilisant la version gratuite et sécurisée en HTTPS.
- Soumettre l'application mobile sur les stores (Google Play, Apple App Store).

- **Documentation**

Fournir une documentation complète sur le code, le processus de développement, et le déploiement.

Données

Vous utiliserez des données d'exemple créées avec un faker de votre choix.

Rendu final

Prototype fonctionnel : Développement d'une application mobile avec les fonctionnalités de base (inscription, messagerie, création d'événements) en React Native ou une technologie équivalente, intégrée à un backend utilisant Symfony ou une autre technologie.

Dossier de conception

Document de 5/10 pages maximum

Détaillez les choix architecturaux (Clean Architecture, MVC ...).

Présentez les modèles de données (diagramme de la base de données)

Détaillez vos choix techniques.

Détaillez les spécifications fonctionnelles.

Documentez un extrait de code.

Détaillez un test fonctionnel.

Commentez un extrait de code.

Décrivez le déploiement

Evaluations

Prototype : Fonctionnalité et pertinence de l'application.

Dossier de conception : Clarté, structuration et justifications des choix techniques.

Respect des délais : vendredi vous devez présenter votre projet à la classe ... Même si votre application n'est pas terminée, essayez au maximum d'anticiper la "livraison" du prototype.

Aides techniques échanges

Un Discord a été créé pour le projet [Projet Mobile](https://discord.gg/8vYtkNHp) (<https://discord.gg/8vYtkNHp>)

Un exemple de Docker Compose pour le projet [Docker Compose](#)

Compétences à valider

1	Choix des architectures de l'application Front et Back (Clean Architecture, MVC ...)
2	Élaborer des modèles de données logiques et physiques (entité-relation, les modèles de données en étoile...) qui correspondent au projet
3	Choix des Stacks techniques (API librairie Framework)
4	Qualité du dossier et de la présentation du projet. Pertinence
5	Versionning et organisation du développement de l'application en équipe
6	Déploiement

Bon courage à tous