

Projet Mobile / Web / Jeu – Simulation de Garage

Contexte

Projet académique **S5 DESIGN – Promotion 4** à réaliser **en groupe de 4**.

L'objectif est de concevoir un **système complet** composé : - d'une **application mobile** - d'une **application web** - d'un **jeu HTML**

Ces trois parties communiquent entre elles via une **API REST** et une **base de données**.

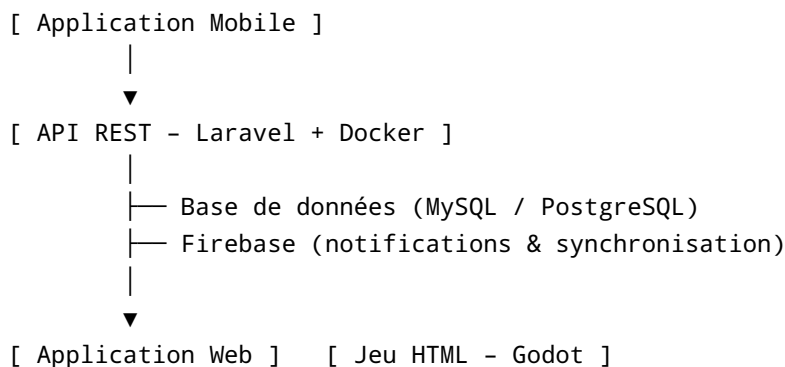
 **17** Date limite de livraison : 16 février

Objectif Général

Simuler le fonctionnement d'un **garage automobile** :

- Le **client** décrit les pannes de sa voiture via le **mobile**
 - Les **réparations** sont effectuées dans le **jeu**
 - Le **garage** suit les réparations et statistiques via le **web**
-

Architecture Globale



Application Mobile (Client)

Rôle

Permettre aux **clients du garage** de gérer leurs voitures et réparations.

Fonctionnalités minimales

- Inscription
- Connexion
- Décrire les pannes de sa voiture
- Voir la liste des voitures à réparer
- Recevoir une **notification** quand la réparation est terminée
- **Payer** pour récupérer la voiture

Technologies

- React Native **ou** Ionic
- Connexion à l'API REST
- Firebase pour les notifications



Application Web



BackOffice (avec login – Garage / Admin)

Fonctionnalités : - Connexion (login simple) - Gestion des **interventions** : - type (frein, vidange, pneu, etc.) - prix - durée (en secondes) - Tableau de statistiques : - montant total des interventions - nombre de clients - Synchronisation avec Firebase



Règle importante : **seulement 2 voitures peuvent être réparées en même temps.**

FrontOffice Web (sans login)

- Voir la liste des clients
- Voir les réparations en cours



Jeu HTML – Godot

Principe du jeu

- Le personnage se déplace dans le garage
- Il voit la liste des voitures
- Il place une voiture dans **1 des 2 slots de réparation**
- Il sélectionne les réparations **une par une**
- Chaque réparation a :
 - une durée
 - une barre de progression
- Il peut réparer une autre voiture pendant l'attente

Fin de réparation

- La voiture va dans le **slot d'attente (1 seul)**
- Le client doit payer pour récupérer la voiture

- Le slot se libère

Réparations imposées (8)

1. Frein
2. Vidange
3. Filtre
4. Batterie
5. Amortisseurs
6. Embrayage
7. Pneus
8. Système de refroidissement



Backend & Technique

- Laravel + Docker
- API REST
- Base de données relationnelle (MySQL ou PostgreSQL)
- Firebase :
- notifications
- synchronisation
- Authentification

Documentation Technique (Obligatoire)

À fournir avec le projet : - MCD (Modèle Conceptuel de Données) - Scénarios détaillés (avec explications)
- Captures d'écran **uniquement des fonctionnalités terminées** - Liste des membres du groupe : - Nom
- Prénom - Numéro étudiant

Évaluation

Les critères : - Fonctionnalités - Qualité du code (GitHub / GitLab public) - Design - Documentation - Gestion des aléas

Prochaines Étapes Conseillées

1. Définir le rôle de chaque membre du groupe
2. Concevoir le **MCD**
3. Définir l'API (routes principales)
4. Mettre en place la base de données
5. Commencer le backend (Laravel)
6. Développer Mobile / Web / Jeu en parallèle



Ce fichier sert de référence générale pour ne rien oublier du projet.