



# Cahier Des Charges

*M2 MIAGE parcours Ingénierie logicielle pour le web*

## *Développement d'une application mobile de la « planche » parfaite.*



Travail réalisé par :

- Adem MANAA
  - Manel REDISSI
    - Ranya Malek
      - Yacine DIABI

2024/2025



## **Table des matières**

1. Présentation du Projet	3
1. 1. Contexte	3
2. 2.Objectifs	3
2. Description Fonctionnelle	3
2.1 Fonctionnalités principales	3
2.2 Fonctionnalités supplémentaires	4
3. Contraintes Techniques	4
3.1 Technologies	4
3.2 Compatibilité	5
3.3 Performances	5
4. Organisation et Gestion du Projet	5
4.1 Planning prévisionnel	5
4.2 Livrables	5
5. Ressources Nécessaires	6
5.1 Humaines	6
5.2 Matérielles	6
5.3 Logicielles	6
6. Critères de Réussite	6

# 1. Présentation du Projet

## 1. 1. Contexte

L'application vise à accompagner les utilisateurs dans la réalisation correcte de l'exercice de la planche, en utilisant la reconnaissance du corps humain pour évaluer la posture et la tenue sur une durée déterminée. À terme, l'application pourra évoluer pour inclure d'autres exercices de callisthénie tels que les pompes ou autres mouvements fonctionnels.

## 2. 2.Objectifs

- **Court terme** : Développer une application fonctionnelle permettant la reconnaissance de la posture de planche.
- **Long terme** : Offrir une application évolutive avec des fonctionnalités supplémentaires adaptées à différents exercices

# 2. Description Fonctionnelle

## 2.1 Fonctionnalités principales

### A. Analyse de la posture :

- a. Détection du corps à l'aide de technologies de reconnaissance via caméra.
- b. Validation de la posture en temps réel.

### B. Chronométrage :

- a. Enregistrement du temps de maintien de la posture correcte.
- b. Arrêt du chronomètre en cas de posture incorrecte.

### C. Feedback utilisateur :

- a. Notifications sonores/visuelles pour corriger la posture.
- b. Résumé des performances après chaque session.



#### D. Historique des sessions :

- a. Enregistrement des résultats par date.
- b. Statistiques pour suivre la progression.

#### E. Connexion / inscription / déconnexion :

- Création d'un espace personnel avec toutes les informations personnelles ainsi que l'accès à d'autres fonctionnalités comme paramètre et historique
- L'utilisateur doit fournir des informations de base (nom, email, mot de passe, etc.).
- Authentification via email et mot de passe.
- Possibilité de réinitialiser le mot de passe en cas d'oubli.
- Bouton de déconnexion avec suppression des données sensibles en cache.

## 2.2 Fonctionnalités supplémentaires

- Évolutions possibles :
  - Ajout d'exercices supplémentaires comme les pompes.
  - Modes multi-utilisateurs pour les compétitions ou entraînements partagés.
  - Personnalisation des défis (durée, répétitions, etc.).

## 3. Contraintes Techniques

### 3.1 Technologies



Kotlin



- **Langage de programmation** : Kotlin (Android).
- **Base de données** : MySQL pour le stockage des données utilisateur, des informations personnelles, des sessions d'exercices, et de l'historique.
- **Backend** : API REST développée avec un framework comme Node.js pour communiquer entre l'application mobile et la base de données.
- **Versionnement du code** : GitHub pour la gestion de projet et collaboration.



## 3.2 Compatibilité

- Plateforme cible : Android 🤖

## 3.3 Performances

- Fonctionnalités fluides sur des appareils de performance moyenne.
- Analyse en temps réel sans latence significative.

# 4. Organisation et Gestion du Projet

## 4.1 Planning prévisionnel

- **Phase 1 : Analyse et conception (1 à 2 semaines)**
  - Étude des besoins et création des maquettes.
  - Choix des outils de machine learning.
- **Phase 2 : Développement (3 à 4 semaines)**
  - Implémentation des fonctionnalités principales.
- **Phase 3 : Tests et itérations (2 semaines)**
  - Validation technique et retours utilisateurs.
- **Phase 4 : Documentation et livraison (1 semaine)**
  - Rédaction des guides utilisateurs et techniques.

## 4.2 Livrables

- Cahier des charges validé.
- Application fonctionnelle.
- Code versionné et documenté.
- Documentation utilisateur.

## 5. Ressources Nécessaires

### 5.1 Humaines

- Équipe de développement mobile (4 développeurs).
- Spécialiste UX/UI
- Testeur

### 5.2 Matérielles

- Ordinateur performant pour le développement.
- Smartphone pour tester les fonctionnalités.

### 5.3 Logicielles

- Android Studio pour le développement.
- Bibliothèques de ML pour la reconnaissance de posture.



## 6. Critères de Réussite

- Reconnaissance de la posture avec une précision  $>90\%$ .
- Application fluide sur les smartphones cibles.
- Retour utilisateur positif sur l'ergonomie et les fonctionnalités.