

# Cahier des Charges

## Application Web de gestion des réparations de téléphones

---

Projet : Fast Repair

## 1. Présentation générale

Les magasins de réparation de téléphones, notamment dans des zones comme à Rabat ou Casablanca, gèrent souvent leurs activités de manière manuelle.

L'objectif du projet est de créer une application Java connectée à une base SQL, permettant aux clients, réparateurs et administrateurs de suivre et gérer efficacement les réparations.

## 2. Objectifs du projet

Objectif principal :

Digitaliser la gestion des réparations de téléphones pour améliorer la communication entre clients, réparateurs et administrateurs.

Objectifs spécifiques :

- Permettre aux clients de consulter l'avancement de leur réparation.
- Donner aux réparateurs un accès direct à leurs tâches.
- Offrir à l'administrateur une interface de supervision et de statistiques.
- Centraliser les données de réparation dans une base SQL fiable et sécurisée.

## 3. Périmètre fonctionnel

Profils utilisateurs :

- Admin (propriétaire) : gère le magasin, les réparateurs et les réparations.
- Réparateur : gère uniquement les téléphones qui lui sont assignés.
- Client : consulte l'état de ses téléphones et les dates de fin prévues.

## 4. Fonctionnalités détaillées

Côté Client :

- Suivi d'une demande de réparation.
- Consultation du statut (en cours, prêt, annulé).

Côté Réparateur :

- Connexion sécurisée.
- Liste des réparations assignées.
- Mise à jour de l'état (en cours, prêt, annulé).
- Saisie de commentaires techniques.

Côté Admin :

- Authentification sécurisée.
- Gestion des réparateurs (ajout, modification, suppression).
- Assignation des réparations.
- Suivi global et statistiques.

## 5. Architecture technique

Langage : Java

Base de données : MySQL / SQL Server

Architecture : MVC

Sécurité : Authentification par rôles (admin, réparateur), les clients n'auront pas à faire une authentification.

## 6. Schéma de base de données

```
admin(id, nom, email, mot_de_passe, nom_magasin, adresse, téléphone)
reparateur(id, admin_id, nom, email, mot_de_passe, specialite, actif)
reparation(id,reparateur_id,appareil,marque,probleme_signale,etat,date_depot,
date_prevue,date_fin,commentaire,code_reparation)
```

## 7. Interfaces utilisateurs

Espace Client :

- \_ Etat de la réparation après la saisie du code de la réparation

Espace Réparateur :

- Liste des téléphones attribués.
- Mise à jour d'état et notes techniques.

Espace Admin :

- Tableau de bord global, statistiques et gestion des utilisateurs.

## **8. Sécurité**

- Accès restreint selon rôle.
- Mots de passe chiffrés (bcrypt ou SHA256).
- Validation des données côté client et serveur.

## **9. Exigences non fonctionnelles**

Performance : chargement < 2 secondes.

Disponibilité : 99 %.

Maintenance : code structuré selon MVC.

## **10. Évolutions futures**

- Envoi de SMS aux clients.
- Génération de factures PDF.
- Tableau de bord statistique dynamique.

## **11. Livrables**

- Cahier des charges (ce document)
- Diagrammes UML (cas d'utilisation, classes, événement)
- Script SQL de création de la base
- Code source Java complet
- Rapport de tests