

**EWUAME Mawuéna Ezechiel**

**Chargé du cours :**

**ing. LAMBONi**

**Documentation du Projet Java :**

**LAVAUTO**

**Table des matières**

1. Guide Utilisateur
   1. Introduction
   2. Interface Principale
   3. Gestion des Services
   4. Gestion des Clients
   5. Gestion des Réservations
   6. Gestion des Factures
2. Documentation Technique
   1. Architecture du Projet
   2. Base de Données
   3. Technologies Utilisées
3. Installation et Configuration
   1. Prérequis
   2. Installation
   3. Configuration de la Base de Données
   4. Maintenance

**1. Guide Utilisateur**

**1.1. Introduction**

LavAuto est une application de gestion pour les services de lavage automobile. Elle permet de gérer les clients, les services, les réservations et les factures de manière simple et efficace.

**1.2. Interface Principale**

L'application est divisée en quatre onglets principaux :

* **Services** : Gestion des services proposés
* **Clients** : Gestion de la base clients
* **Réservations** : Gestion des rendez-vous
* **Factures** : Gestion de la facturation

**1.3. Gestion des Services**

**Ajouter un service**

1. Cliquez sur l'onglet "Services"
2. Cliquez sur le bouton "Ajouter"
3. Remplissez les informations :
   * Nom du service
   * Prix (en FCFA)
4. Cliquez sur "Enregistrer"

**Modifier un service**

1. Sélectionnez le service dans la liste
2. Cliquez sur "Modifier"
3. Mettez à jour les informations
4. Cliquez sur "Enregistrer"

**1.4. Gestion des Clients**

**Ajouter un client**

1. Cliquez sur l'onglet "Clients"
2. Cliquez sur "Ajouter"
3. Remplissez les informations :
   * Nom
   * Prénom
   * Téléphone
4. Cliquez sur "Enregistrer"

**Modifier un client**

1. Sélectionnez le client dans la liste
2. Cliquez sur "Modifier"
3. Mettez à jour les informations
4. Cliquez sur "Enregistrer"

**1.5. Gestion des Réservations**

**Créer une réservation**

1. Cliquez sur l'onglet "Réservations"
2. Cliquez sur "Ajouter"
3. Sélectionnez :
   * Le client
   * Le service
   * La date et l'heure
   * Le statut (en attente, en cours, termine)
4. Cliquez sur "Enregistrer"

**Suivi des réservations**

Les réservations sont affichées avec leur statut :

* **En attente** : Réservation future
* **En cours** : Service en cours de réalisation
* **Termine** : Service terminé

**Note importante :** Une facture est automatiquement créée lorsqu'une réservation passe au statut "termine"

**1.6. Gestion des Factures**

**Consulter les factures**

1. Cliquez sur l'onglet "Factures"
2. La liste affiche :
   * Numéro de facture
   * Client
   * Date
   * Montant
   * Statut de paiement

**Marquer une facture comme payée**

1. Sélectionnez la facture
2. Cliquez sur "Payer"
3. Confirmez le paiement

**2. Documentation Technique**

**2.1. Architecture du Projet**

Le projet est structuré en plusieurs couches :

* **UI** : Interface utilisateur (Swing)
* **DAO** : Accès aux données
* **Model** : Classes métier
* **DB** : Gestion de la base de données

**2.2. Base de Données**

|  |  |
| --- | --- |
| **Table** | **Description** |
| clients | Stockage des informations clients |
| services | Liste des services disponibles |
| reservations | Gestion des rendez-vous |
| factures | Suivi des paiements |

**2.3. Technologies Utilisées**

* Java 8+
* Swing pour l'interface graphique
* MySQL pour la base de données
* JDBC pour l'accès aux données

**3. Installation et Configuration**

**3.1. Prérequis**

* Java JDK 8 ou supérieur
* MySQL Server
* Maven (optionnel)

**3.2. Installation**

1. Clonez le projet
2. Importez la base de données
3. Configurez les accès à la base de données dans DatabaseConnection.java
4. Compilez le projet
5. Lancez l'application

**3.3. Maintenance**

* Sauvegarde régulière de la base de données recommandée
* Vérification périodique des logs d'erreur
* Mise à jour des pilotes JDBC si nécessaire