Design Document

Veloway
Smart Bicycle Rental System

Authored by:

Emna Belguith

Rim Barnat

Molka Sahli

Nour Krichen

Supervised By:

Dr. Eng. Mohamed-Bécha Kaaniche

Academic Year: 2025

Contents

1. Introduction

1.1 Problematic

The main challenge addressed by this project is the lack of efficient, real-time systems for public bicycle rental management. Current solutions often fail to ensure a seamless user experience that integrates IoT data, mobile applications, and computer vision for damage verification. Users face difficulties locating available bikes, managing their trips, or ensuring the quality of the rented bikes.

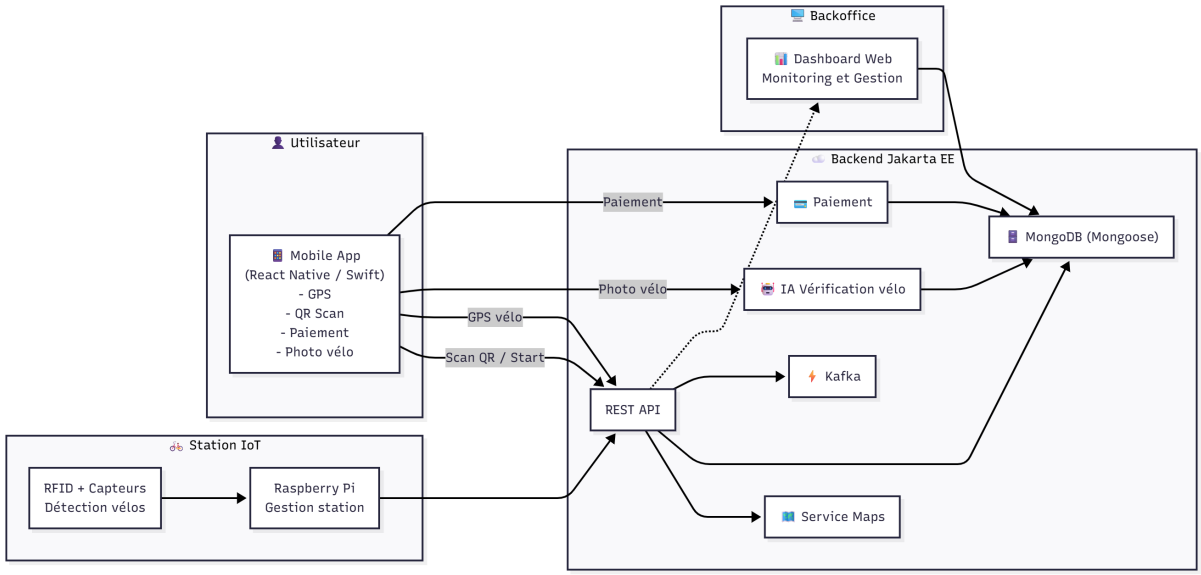
1.2 Context of the Project

This project targets urban mobility services and users seeking a fast, eco-friendly, and automated way to rent bicycles. It leverages IoT and AI technologies to offer a smart and secure rental experience, integrating mobile interaction, payment management, and real-time monitoring of station availability.

1.3 Architecture

The proposed architecture integrates four major components:

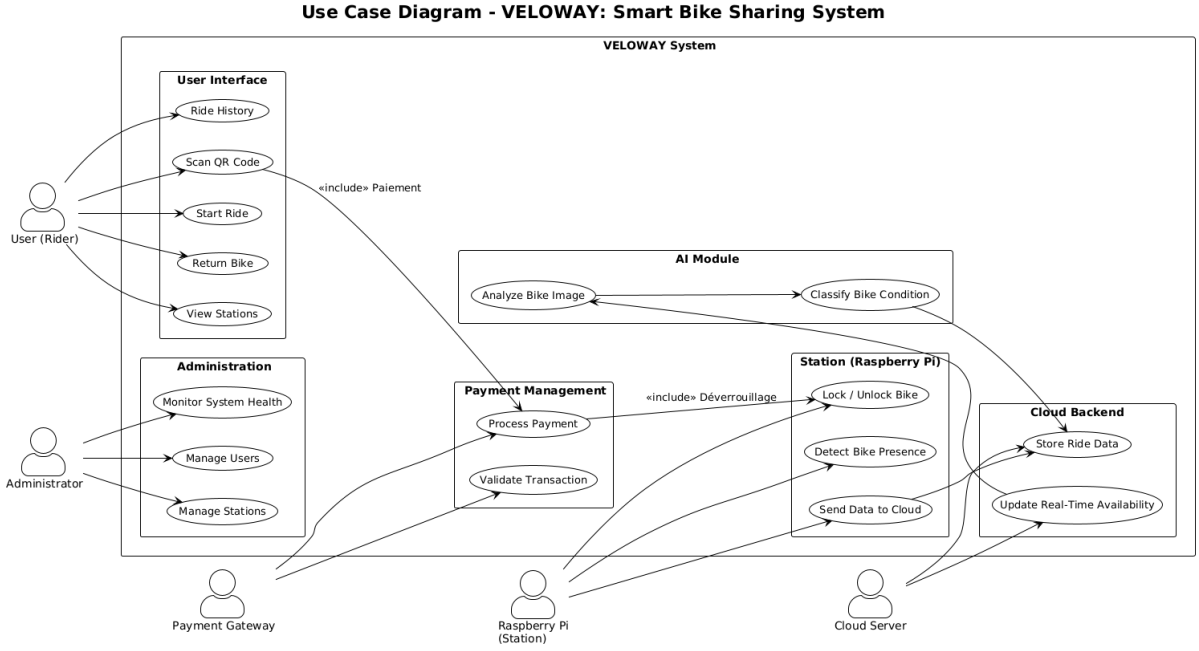
- **Backend (Jakarta EE):** manages authentication, payments, and data exchange between users, IoT devices, and the database.
- **Frontend :** provides the user interface for booking, tracking, and ending trips.
- **IoT Layer:** Raspberry Pi acts as a gateway between the bike station's sensors and the cloud backend.
- **AI Module:** a computer vision model analyzes uploaded bike photos to detect potential damage.


Backoffice
Dashboard Web
Monitoring et Gestion
Utilisateur
Backend Jakarta EE
Mobile App
(React Native / Swift)
- GPS
- QR Scan
- Paiement
- Photo vélo
Paiement
Paiement
MongoDB (Mongoose)
Photo vélo
IA Vérification vélo
GPS vélo
Scan QR / Start
Kafka
REST API
Station IoT
RFID + Capteurs
Détection vélos
Raspberry Pi
Gestion station
Service Maps

1. UML Diagrams

1.1. Use Case Diagram

The Use Case diagram illustrates the main interactions between the actors and the system functionalities. The system has two primary actors: **Users** and **Administrators**. Users can scan QR codes, authenticate, rent a bike, track their route, and return the bike after a visual verification. Administrators can manage station data, oversee IoT devices, and maintain system updates.

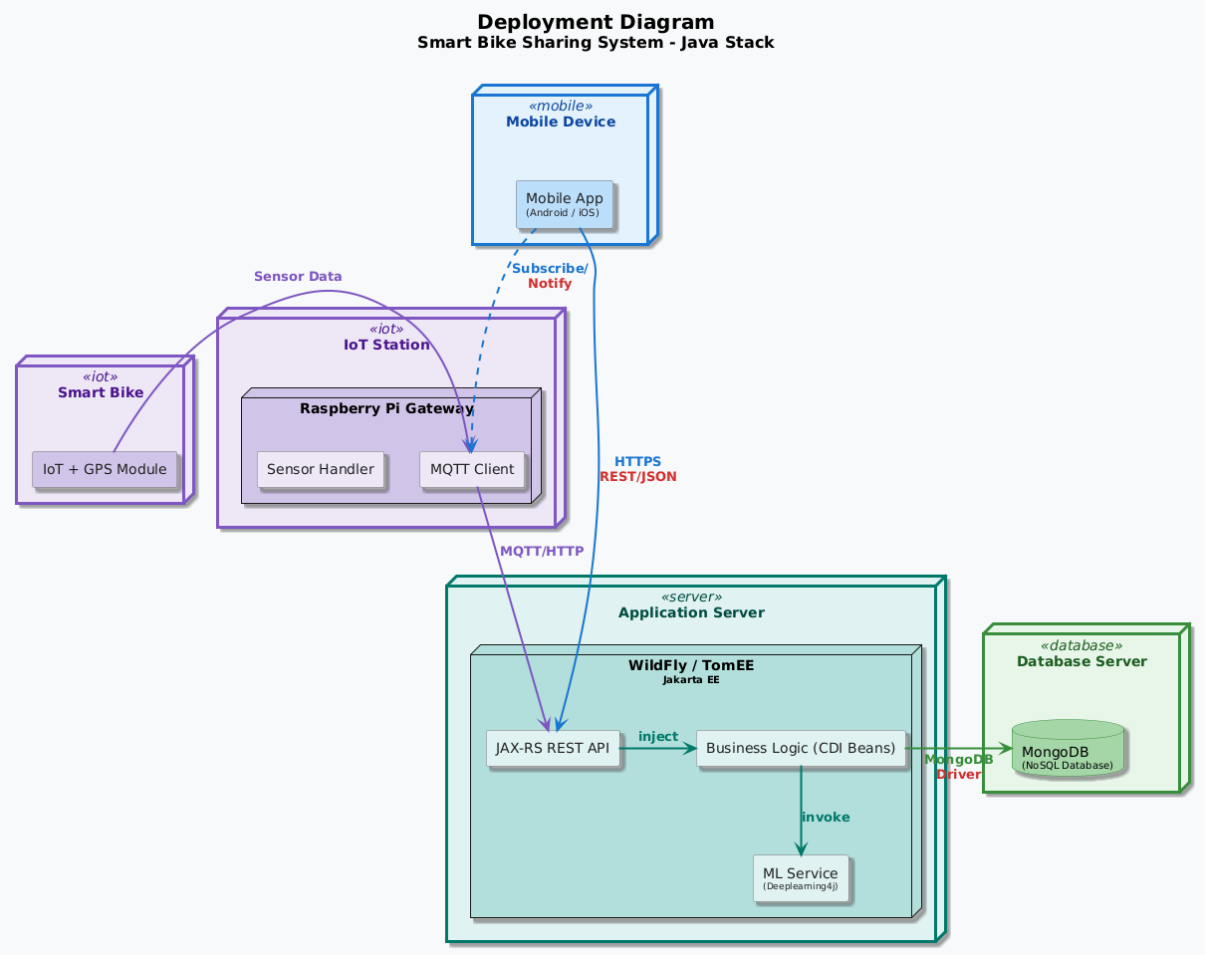


1.2. Deployment Diagram

This figure shows the deployment diagram illustrating how software components are distributed across physical devices. The architecture includes:

- **User Device:** runs the front-end app with map and camera access.
- **Raspberry Pi:** collects IoT data from the station and communicates with the backend.

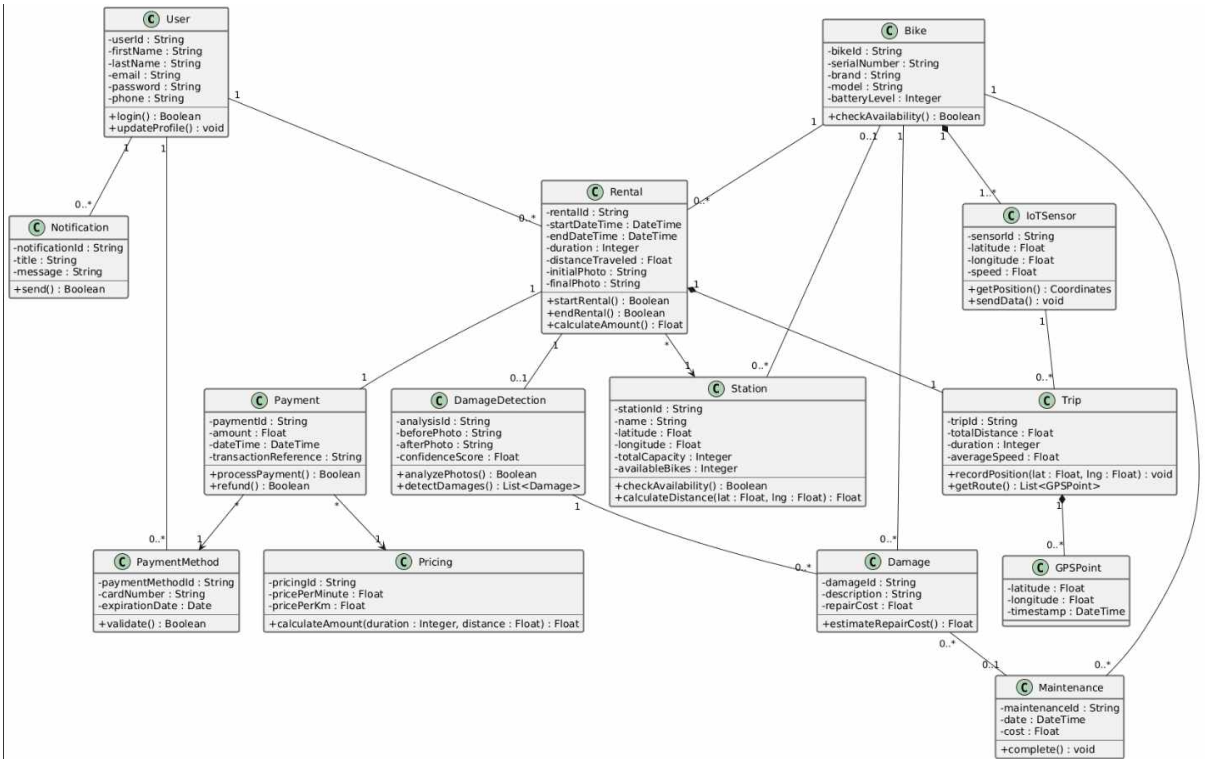
- **Cloud Server:** hosts the Jakarta EE backend and database.



1.3. Class Diagram

The Class diagram represents the object-oriented structure of the system. Key classes include:

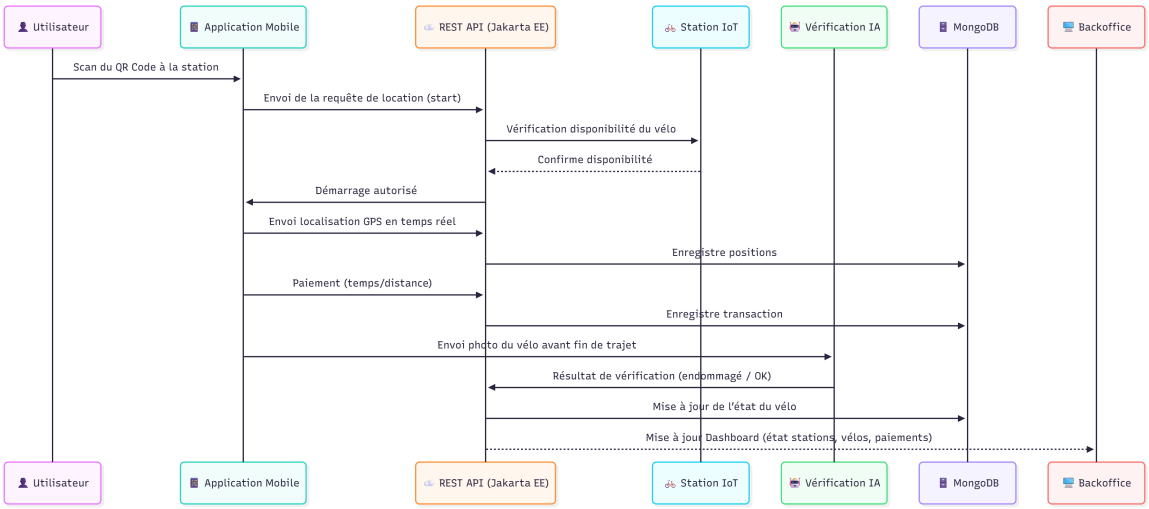
- **User:** manages account and rental information.
- **Bike:** stores bike details and condition.
- **Station:** handles available bikes and empty slots.
- **Rental:** links users to their rented bikes.
- **AIService:** processes uploaded images for damage detection.



1.4. Sequence Diagram

The sequence diagram (Figure 4) illustrates the process from scanning a QR code to completing a rental:

1. The user scans a QR code at the station.
2. The app connects to the backend to authenticate and initialize the rental.
3. IoT data confirms bike availability.
4. During the trip, the user's position is tracked via GPS.
5. Upon completion, the user takes a photo of the bike.
6. The AI module evaluates the image to detect any damage before closing the rental.


Utilisateur
Application Mobile
REST API (Jakarta EE)
Station IoT
Vérification IA
MongoDB
Backoffice
Scan du QR Code à la station
Envoi de la requête de location (start)
Vérification disponibilité du vélo
Confirme disponibilité
Démarrage autorisé
Envoi localisation GPS en temps réel
Enregistre positions
Paiement (temps/distance)
Enregistre transaction
Envoi photo du vélo avant fin de trajet
Résultat de vérification (endommagé / OK)
Mise à jour de l'état du vélo
Mise à jour Dashboard (état stations, vélos, paiements)
Utilisateur
Application Mobile
REST API (Jakarta EE)
Station IoT
Vérification IA
MongoDB
Backoffice

Conclusion

In conclusion, the Smart Bicycle Rental System provides an integrated and scalable solution for urban bike-sharing management. By combining IoT, AI, and modern web technologies, it ensures:

- Seamless user experience from station to app.
- Secure authentication and payment.
- Real-time monitoring of stations and available bikes.
- Automated verification of bike condition using computer vision.

This system demonstrates the potential of IoT and AI to enhance sustainable mobility and improve user satisfaction in smart city environments.