Orange Coding Academy

Masterpiece Project Documentation

EcoRideJordan

Prepared by: Suhaib Abuleil

Contents

Contents

Chapter 1: Introduction

1.1 Objective

MotoRent is a web-based platform that connects motorcycle owners and enthusiasts with convenient rental and maintenance services.
The platform allows users to rent motorcycles online, pay securely through the website, or fill out a service request form to schedule maintenance for their bikes.
It is designed with user experience in mind, providing a smooth, fast, and efficient way to access essential motorcycle services – all through one centralized system.
Additional features and services are also in development to expand the platform's capabilities and offer even more value to its users.

2.2 Technologies Used for EcoRideJordan

Frontend:

- HTML/CSS/
- Bootstrap
- JavaScript

Backend:

- Asp Core MVC
- SQL Server

Chapter 2: Requirements and Analysis

2.1 Functional Requirements

Functional requirements in software engineering define the intended behavior of a system, which may include functions, services, or tasks that the system must perform.

2.1.1 User (Client):

- Home page.
- The client can register for a new account.
- The client can log in to the system.
- The client can browse available bikes for rent.
- The client can fill up a form to request maintainence for his bike.
- The client can filter and search bikes by type, model, or availability.
- The client can view detailed bike specifications and features.
- The client can make secure online payments via credit card.
- Adding notes or additional information for translators.
- The client can view and manage their rental and maintenance history.
- The client can contact support through a built-in chat or contact form.

2.1.2 Admin:

- The admin has his own login and can manage everything in the website.

2.2 Non-Functional Requirements for Tarjim

Non-functional requirements describe the overall qualities and properties of a system, also known as quality attributes.

2.2.1 Security:

- **Data Privacy:** User data must be securely stored and accessible only to authorized roles, such as admins and service providers, while allowing users to view their own bookings and profile information.
- **User Authentication:** The system employs authentication for the login feature, ensuring that only registered users can access their profiles and make bookings.

2.2.2 User-Friendly:

- The system is designed with an intuitive user interface, enabling users to easily browse services, view bookings, and read testimonials without confusion.

2.2.3 Usability:

- The website prioritizes ease of use, featuring simple controls and a straightforward navigation flow for users to explore services, view testimonials, manage their profiles, and make bookings efficiently.

2.2.4 Availability:

- The website will be available 24/7, providing users with continuous access to services for booking venues, managing profiles, and viewing past events.

2.2.5 Responsive Web Design:

- The website is fully responsive, ensuring a seamless user experience across various devices, including desktops, tablets, and mobile phones.

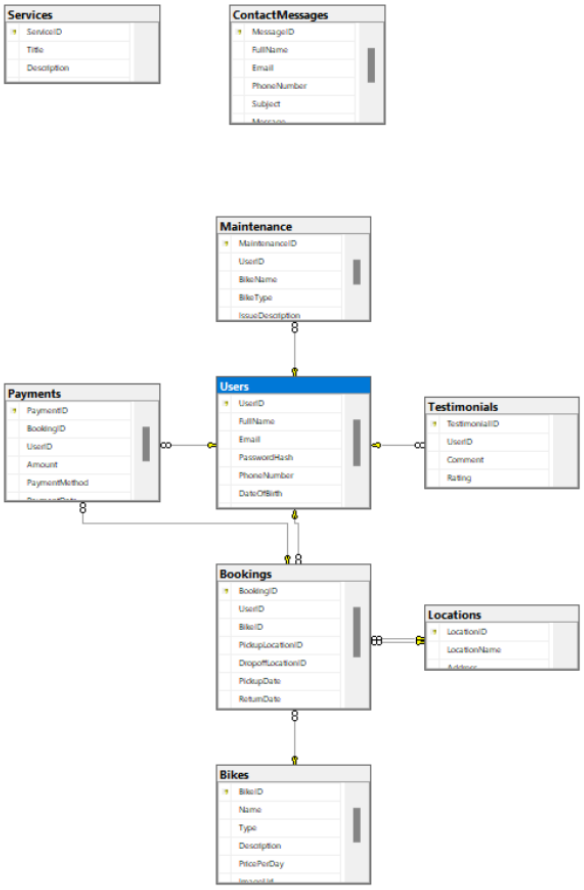
2.3 Test Cases for EcoRideJordan

- **Authentication Requirement for Service Requests:**
  When a user tries to rent a motorcycle or submit a maintenance request without logging in, they will be redirected to the login or registration page to continue.

- **Role-Based Access Control:**
  Only registered users can access their personal dashboards to view bookings, track maintenance, or submit new requests. Unauthenticated users attempting to access these pages will be redirected or shown an unauthorized access message.
- **Admin Access Control:**
  Admin-only features such as managing bikes, viewing booking statistics, updating service listings, or handling user reports are strictly limited to users with admin roles. Any unauthorized access attempt will be denied with proper redirection or alert.

Chapter 3: Design

3.1 Database Schema



Relationships:

- **Users Table**
  - Primary Key: UserId
  - Relationships:
    - One-to-Many with Bookings (One user can create multiple rental bookings)
    - **One-to-Many with Payments (One user can make multiple payments)**
    - **One-to-Many with Maintenance (One user can submit multiple maintenance requests)**
    - **One-to-Many with Testimonials (One user can leave multiple reviews/testimonials)**

- **• Bikes Table**

  - Primary Key: BikeID
  - Relationships:
    - ◦ **One-to-Many with Bookings (One bike can be booked multiple times)**

- **• Bookings Table**

  - Primary Key: BookingID
  - Relationships:
    - ◦ Many-to-One with Users (Each booking is made by one user)
    - ◦ Many-to-One with Bikes (Each booking is for one bike)
    - ◦ **One-to-One with Payments (Each booking has one payment)**

- **Payments Table**

  - Primary Key: PaymentId
  - Relationships:
    - ◦ Many-to-One with Users (Each payment is made by one user)
    - ◦ Many-to-One with Bookings (Each payment is linked to one booking)

- **• Maintenance Table**

  - Primary Key: MaintenanceID
  - Relationships:
    - ◦ Many-to-One with Users (Each maintenance request is submitted by one user)

- **• Testimonials Table**

  - Primary Key: TestimonialID
  - Relationships:
    - ◦ Many-to-One with Users (Each testimonial is written by one user)

- **• Locations Table**

  - Primary Key: LocationID
  - Relationships:
    - ◦ One-to-Many with Bookings (One location can be used for multiple pickups or drop-offs)

- **• Services Table**

  - Primary Key: ServiceID
  - Relationships:
    - No direct foreign key relationships (Used to list platform services/features)

Chapter 4: Conclusion & Future Work

4.1 Conclusion

Tarjim was designed to provide a comprehensive translation service platform that connects clients with qualified translators. The system supports both instant and project-based translation requests, facilitates secure communication, file management, and payment processing. By offering dashboards tailored to each role (Admin, Client, Translator), the platform ensures efficient project tracking, feedback, and control. Tarjim enhances the overall translation workflow and brings convenience and structure to freelance language services.

4.2 Future Work

Implement a real-time chat feature between clients and translators to streamline communication.

Add automated document parsing to estimate project cost and delivery time.

Expand the platform to support translation agencies and team collaboration.

Enhance the review and rating system to include detailed feedback per category (e.g., accuracy, delivery, c

Develop a mobile application version for both clients and translators.

Add multi-language support for platform UI to improve global accessibility.

Integrate AI-powered translation suggestions to help translators work more efficiently.