**User Requirements Document**

**Bike Rental Web Application**

### 1. Overview

This document defines the user requirements for the bike rental web application. The platform aims to deliver a seamless and efficient bike rental experience in Dublin, catering to various user groups, including casual riders, subscribers, business owners, and maintenance staff.

### 2. Goals of the Project

**1. Provide Transport**

The main goal of our web application is to offer a reliable and accessible transport solution for people in Dublin. By providing bicycles, e-bikes, and e-scooters, we aim to reduce dependence on cars and public transport for short-distance travel.

**2. Make City Travels Easier**

Our platform should simplify urban mobility by allowing users to quickly locate and rent vehicles through an intuitive web interface. The system will ensure a smooth rental process, reducing waiting times and making travel hassle-free.

**3. Make Travel Cheaper**

We aim to offer an affordable alternative to traditional transportation methods. Our pricing model will include pay-as-you-go options and subscriptions, making it accessible to students, tourists, and daily commuters.

**4. Good Selection of Vehicles and Subscription Plans**

To meet different user needs, our service will provide a variety of transport options, including bicycles, electric bikes, and scooters. Additionally, we will introduce flexible subscription plans to accommodate occasional and frequent users.

**5. Provide a Flexible and Usable System**

The platform should be user-friendly, with a responsive design that works on different devices. It should also offer multiple payment options, real-time vehicle availability tracking, and easy customer support to enhance user experience.

**3.1 User Roles & Needs**

#### 1. Tourists / Pay-As-You-Go Users

**Role Description:**  
Casual or one-off users who are visiting Dublin and need a convenient transportation option without a long-term commitment.

* **Quick Access & Simplicity:**
* **Efficient Booking Process:**
* **Flexible Payment Options:**
* **Minimal Friction:**

#### 2. Special Categories (e.g., Students and Other Eligible Groups)

**Role Description:**  
Users like students or other special groups who qualify for discounts or special offers.

**Needs:**

* **Discounted Pricing:**
* **Verification Process:**
* **Tailored Subscription Options:**
* **Enhanced Customer Support:**

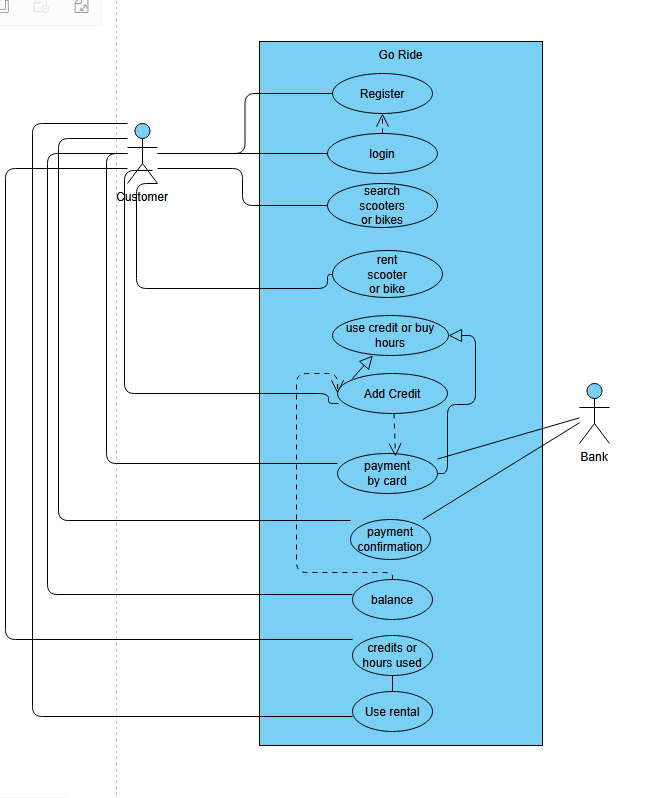
#### 3. Hobbiest’s

**Role Description:**  
Users who lead an active lifestyle, preferring classic bicycles over motorized options. They are motivated by low prices and a focus on fitness and reducing pollution.

**Needs:**

* **Dedicated Vehicle Options:**
* **Competitive Pricing:**
* **Fitness and Activity Tracking:**
* **Community and Incentives:**

**3.2** **Use Case Diagram for Customers Using The Website**



**3.3 Use Case Descriptions**

|  |  |
| --- | --- |
| **Use Case Id** | 1 |
| **Use Case Name** | Adding credit |
| **Created by** | Yones Ananzeh |
| **description** | The user will try to add credit to their account the user will have to access the system |
| **primary Actors** | Customer |
| **Secondary Actors** |  |
| **Pre-Conditions** | 1. The Customer must be on the website. 2. The Customer must have an account. 3. The Customer must have a debit or credit card. |
| **Post Conditions** | 1. The Customer receives the credits. |
| **Main Path** | 1. The Customer accesses the website 2. The Customer logs in to the website 3. The Customer goes to their balance and adds credits 4. The Customer their payment card to pay for the credits 5. The bank confirms the payment 6. The Customer has added credits to their account |
| **Alternate Path** | 1. The Customer accesses the website 2. The Customer has no account 3. The Customer registers 4. The Customer goes to their balance and adds credits 5. The Customer their payment card to pay for the credits 6. The bank denies the payment 7. The Customer has no credits. |

|  |  |
| --- | --- |
| **Use Case Id** | 2 |
| **Use Case Name** | Renting a scooter |
| **Created by** | Yones Ananzeh |
| **description** | The Customer will try to rent a scooter from the website. |
| **primary Actors** | Customer |
| **Secondary Actors** | Bank |
| **Pre-Conditions** | 1. The customer has access to the website 2. The customer must login 3. The customer must have goride credit or a payment card |
| **Post Conditions** | 1. The customer is able to use the rental |
| **Main Path** | 1. The Customer accesses the website 2. The Customer logs in to the website 3. The customer browses scooters 4. The customer rents a scooter. 5. The customer selects their scooter to rent. 6. The customer uses credit to rent the scooter. 7. The customer can now use the scooter. |
| **Alternate path** | 1. Steps 1-5 of the main path. 2. The Customer selects how much time they will use the scooter for 3. The customer uses payment card. 4. The bank confirms the payment. 5. The User can now use the scooter. |

**4.Functional Requirements**

**Core features**

***1. Search Transport***

**Description:** Users should be able to find available bicycles, e-bikes, and e-scooters near them.

**Implementation**

**2. Login**

**Description:** Users must be able to log in to access booking and payment features.

**Implementation**

**Django Authentication System** will be used for handling user login.

**3. Registration**

**Description:** New users should be able to create an account to book and rent vehicles.

**Implementation**

**4. Payment**

**Description:** Users should be able to pay for their bookings using different payment methods.

**Implementation**

**External Integration:**

Integration with Stripe or PayPal for handling transactions.

**5. Admin Panel**

**Description:** The admin panel will allow admins to manage users, vehicles, and bookings.

**Implementation**

**Django Admin:**

Admin interface to manage User, Vehicle, Booking, and Payment models.

**6. Booking Process**

**Description:** Users should be able to book a vehicle and track their ongoing rental.

**Implementation**

**7. Support**

**Description:** Users should be able to contact support for issues related to booking, payments, or technical problems.

**Implementation**

**Extra features**

#### 1. ****Subscription Plans****

**Description:**  
Expand the payment system by offering various rental options such as daily, weekly, and monthly plans, including special rates for students and a pay-as-you-go model.

**Example:**  
Users can choose a daily pass when traveling short distances or opt for a weekly/monthly plan for regular commuting, with a special discounted rate available for students.

#### 2. ****Map Integration****

**Description:**  
Enhance the user experience by displaying an interactive map to locate available vehicles, view their status, and see service coverage areas (e.g., within Dublin or specific zones).

**Example:**  
Users can see available e-scooters on the map, choose the one nearest to their location, and even get directions to the vehicle or to the designated return spots.

#### 3. ****Cashback System****

**Description:**  
Offer a cashback incentive if users return vehicles to designated parking spots, promoting proper use of the service and enhancing sustainability.

**Example:**  
If a user returns a bike to a verified parking zone (visible on the map), a portion of their fare is credited back to their account as cashback.

#### 4. ****Promocode Feature****

**Description:**  
Allow users to apply promocodes during the payment process to receive discounts.

**Example:**  
A user enters a promocode to receive a 10% discount on their rental, and the total cost is recalculated accordingly.

#### 5. ****Referral System****

**Description:**  
Encourage word-of-mouth marketing by allowing users to refer friends and earn rewards.

**Example:**  
When a referred friend signs up and completes their first booking, the referrer earns a bonus (e.g., credit towards future rentals).

#### 6. ****Language Versions****

**Description:**  
Support multiple languages to serve a diverse user base, starting with English and Gaeilge , with potential future support for languages like French or Spanish.

**Example:**  
Users can switch between English and Gaeilge via a language dropdown menu, and all text on the site will dynamically change based on the selection.

#### 7. ****Extended Support Features****

**Description:**  
Broaden the support channels by including options like email, messenger integration, and in-app chat for real-time assistance.

**Example:**  
Users facing issues with booking or payment can choose to chat directly within the app instead of waiting for an email response.

#### 8. ****Credit System****

**Description:**  
Introduce an internal wallet allowing users to top up an internal balance, reducing the need for repeated external transactions.

**Example:**  
Users can preload funds into their wallet and then pay for rentals using their account balance, streamlining the payment process.

#### 9. ****Vehicle Access****

**Description:**  
Offer multiple methods for users to unlock vehicles, including QR code scanning and a fallback manual key generation system.

**Example:**  
Users scan the QR code on a vehicle using their mobile device; if the camera is not accessible, they can request a unique access key as an alternative.

#### 10. ****Notification System****

**Description:**  
Provide timely notifications to users via SMS, email, and browser pop-ups, ensuring they stay informed about booking statuses and promotions.

**Example:**  
Users receive an SMS when their rental starts, an email confirmation for payments, and browser pop-ups for timely updates or promotional offers.

### ****11. User Stats & Insights****

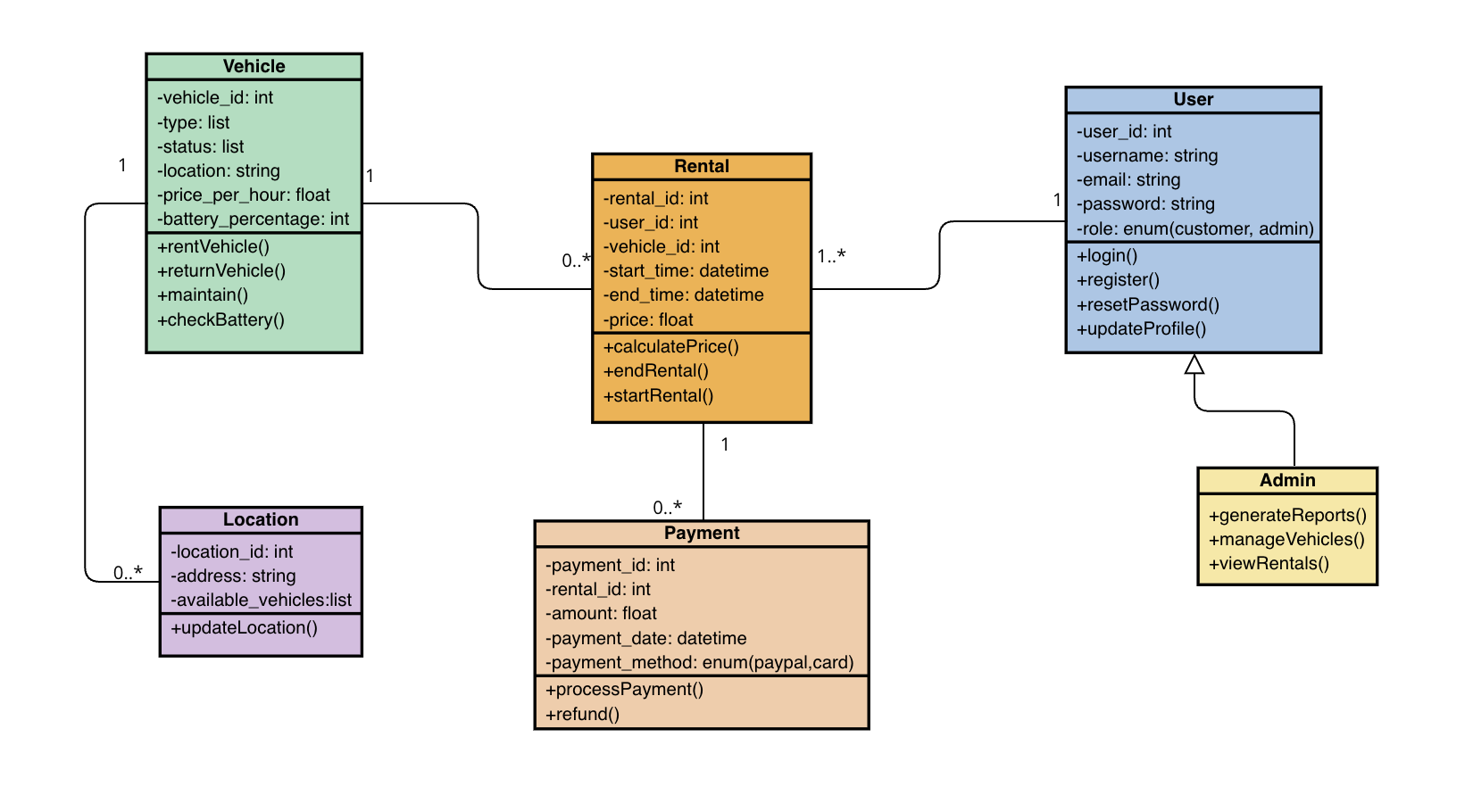
#### ****Description:****

Enhance user experience by providing detailed statistics about their trips, including total distance traveled, number of rides, average speed, and personal insights. Additionally, an annual summary will highlight key riding habits, favorite transport types, and the user’s ranking in the community.

### ****Example:****

John, a frequent e-bike user, checks his dashboard and sees that he has taken **150 rides**, traveled **500 km**, and saved **80 kg of CO₂** this year. His favorite transport type is an **electric scooter**, and his most active day was **May 10th**. At the end of the year, he receives a **personalized Wrapped-style summary** comparing his stats to other users.

**4.6 Class Diagram**

****

**5. Non-Functional Requirements**

**5.1 Performance & Scalability**

* The system must handle **high user traffic** and support future expansion.
* Optimized for **fast loading times** and **real-time updates.**

**5.2 Security & Compliance**

* **End-to-end encryption** for user data & transactions.
* **GDPR compliance** for data protection.
* **Regular security audits** to prevent vulnerabilities.

**5.3 User Experience (UX) & Accessibility**

* **Mobile-first design** for seamless access across devices.
* **Intuitive user interface** for effortless navigation.
* **Accessibility features** to accommodate users with disabilities.

**6. Project Feasibility**

**6.1 Timeline & Development Plan**

* **Three-month development period (Jan 27 – Apr 28)** with a **prototype demo every 3 weeks**.
* **Prioritize core features** (authentication, rentals, payments, mapping) before enhancements.
* **Development milestones:**
  + **Week 3:** Basic booking system.
  + **Week 6:** Payment integration.
  + **Week 9:** Admin dashboard & analytics.

**6.2 Potential Risks & Contingency Plans**

1. **Limited Team Capacity**
   * **Risk:** A high workload could lead to burnout.
   * **Solution:** Focus on MVP features & ensure knowledge-sharing within the team
   * .
2. **Technical Challenges**
   * **Risk:** Potential issues with database, APIs, or security.
   * **Solution:** Allocate debugging time & maintain thorough documentation.
3. **Strict Deadlines & Unexpected Delays**
   * **Risk:** Bugs or unforeseen problems affecting timelines.
   * **Solution:** Include buffer time in the final weeks to resolve last-minute issues.
4. **User Testing & Feedback Implementation**
   * **Risk:** Major usability issues discovered late in development.
   * **Solution:** Conduct early-stage testing with simulated users.