**🛴 Scooter Rent Web App (PYM Scooter)**

This is a minimal **Scooter Rent Web App** template for managing scooter rentals across multiple stations in Basel. The project serves as a reference implementation and can be customized based on specific requirements.

🚧: **This is a template project**: Adapt this documentation and the source code according to your needs. Remove all placeholder comments (🚧) before the final submission.

**📌 Contents:**

* [Analysis](#analysis)
  + [Scenario](#scenario)
  + [User Stories](#user-stories)
  + [Use Case](#use-case)
* [Design](#design)
  + [Wireframe](#wireframe)
  + [Prototype](#prototype)
  + [Domain Model](#domain-model)
  + [Business Logic](#business-logic)
* [Implementation](#implementation)
  + [Backend Technology](#backend-technology)
  + [Frontend Technology](#frontend-technology)
* [Execution](#execution)
* [Deployment](#deployment)
* [Project Management](#project-management)
  + [Roles](#roles)
  + [Milestones](#milestones)

**🔍 Analysis**

**📖 Scenario**

**PYM Scooter** is a lightweight scooter rental web application designed to allow users to rent scooters at various locations across Basel. The app enables users to find available scooters, view rental stations, track service disruptions, and access discounts and offers. Admins can manage scooter availability, rental pricing, and station information.

The scooter rental service includes **four main stations**:

1. **Basel SBB**
2. **Basel Badischer Bahnhof**
3. **Basel Bankveria**
4. **Basel Claraplatz**

Additionally, the system provides:

* **Live map integration** for station locations
* **News updates** (e.g., road closures or special promotions)
* **Contact options** for customer support via phone and email

**👥 User Stories**

1. As a **User**, I want to see a list of scooter stations so that I can choose the closest one.
2. As a **User**, I want to view available scooters at each station.
3. As a **User**, I want to rent a scooter for a specific duration.
4. As a **User**, I want to see **discount offers** at specific locations.
5. As a **User**, I want to check for **station updates** (e.g., closures or maintenance).
6. As an **Admin**, I want to **add, update, or remove scooters** from stations.
7. As an **Admin**, I want to manage rental prices and discount campaigns.
8. As an **Admin**, I want to track active rentals and monitor scooter availability.

**📌 Use Cases**

* **UC-1: Show all Stations** → Users can view all available scooter stations on a map.
* **UC-2: Show Station Details** → Users can check the number of scooters available at a specific station.
* **UC-3: Rent a Scooter** → Users can select a scooter, specify rental duration, and proceed with booking.
* **UC-4: Show Special Offers** → Users can view promotional discounts at different locations.
* **UC-5: Manage Scooters (Admin)** → Admins can add, update, and remove scooters from stations.

**🎨 Design**

**📌 Wireframe**

🚧: The wireframe includes a **header, menu bar, map, station list, rental options, news section, and contact details**. The menu allows users to navigate between different sections, and a **dropdown menu**provides additional options.

**📌 Prototype**

🚧: The initial prototype consists of placeholder elements and can be built using a low-code tool like **Budibase**.

**📌 Domain Model**

The **domain model** consists of the following entities:

* **User**: Contains user details such as name, contact info, and rental history.
* **Admin**: Manages scooter stations, pricing, and availability.
* **Station**: Represents a scooter rental station (Basel SBB, Basel Badischer Bahnhof, etc.).
* **Scooter**: Contains information about each scooter (model type, battery level, availability).
* **Rental**: Manages user rentals, including duration, cost, and rental status.

**📌 Business Logic**

**Special Offers by Location**

* If the station is **"Basel Badischer Bahnhof"**, users receive **20% off** rentals.
* If the station has **track work** (e.g., road closures), users see a **service disruption notice**.
* Otherwise, standard rental pricing applies.

**API Endpoint Example**:

pgsql

CopyEdit

GET /api/stations?location="Basel Badischer Bahnhof"

Response: { "discount": "20%", "message": "Special offer on all scooter rentals!" }

**🛠 Implementation**

**📌 Backend Technology**

* **Spring Boot (Java)** for API development
* **Spring Data JPA** for database interactions
* **H2 Database** for testing (or PostgreSQL for production)
* **Swagger API Documentation**
* **JWT Authentication** for user logins

**Example Dependency for H2 Database**:

xml

CopyEdit

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

**📌 Frontend Technology**

* **React.js / Vue.js** for UI development
* **Map API Integration** (Google Maps or OpenStreetMap)
* **Axios** for API requests
* **Tailwind CSS / Bootstrap** for styling

**🚀 Execution**

**Steps to Run Locally**

1. **Clone the repository:**

bash

CopyEdit

git clone https://github.com/your-username/scooter-rent.git

cd scooter-rent

1. **Start the Backend:**

bash

CopyEdit

mvn spring-boot:run

1. **Start the Frontend:**

bash

CopyEdit

cd frontend

npm install

npm start

1. **Open the web app at:** http://localhost:3000

**☁️ Deployment**

**📌 PaaS Deployment (Optional)**

The backend can be deployed using **Render** or **Heroku**.

**Steps:**

1. Deploy the backend on **Render** or **AWS**.
2. Update the API URL in the frontend.
3. Deploy the frontend on **Netlify** or **Vercel**.

**📊 Project Management**

**📌 Roles**

* **Backend Developer:** [Your Name]
* **Frontend Developer:** [Your Name]

**📌 Milestones**

1. **Analysis**: Define user requirements and use cases.
2. **Prototype & Design**: Wireframe creation and initial frontend setup.
3. **Domain Model & API Design**: Define backend structure and endpoints.
4. **Implementation**: Develop backend services and frontend UI.
5. **Testing & Deployment**: Ensure security and performance optimizations.
6. **(Optional) Cloud Deployment**: Deploy to PaaS (Render, Netlify, etc.).

**📝 Maintainer**

* **[Your Name]**
* **GitHub:** [your-username](https://github.com/your-username)
* **Email:** contact@yourdomain.com

**📜 License**

This project is licensed under the **Apache License 2.0**. See the [LICENSE](blob/master/LICENSE) file for details.