# *MANAGEMENT OF TOURISM ACTIVITIES*

By: Marcos Medina Peula (Erasmus)

Index

# 1. Project Description

This project's main objective is to develop a computer system to digitize the current manual system for managing tourist activities. The system will store and manage information related to the different activities and tours within cities. Additionally, the system will automate all activities related to bookings and cancellations of activities by users.

Furthermore, it will offer a wide range of options for businesses to promote their activities with full precision, specifying the location, type of activity, capacity, and target audience, among other details.

The system will therefore increase the public exposure of the involved businesses while facilitating and speeding up users' access, management, and selection of activities, eliminating the need for them to visit physical sales points.

# 2. Case Study – International & National Context

## 2.1. Global Landscape: The Rise of Experience-Marketplaces

Travel-experience spend is estimated at ≈ US $300 billion annually.
Pure-digital platforms dominate this segment:

- Viator (Tripadvisor) – >300 000 bookable experiences worldwide.
- GetYourGuide – Unicorn status; heavy investment in mobile UX and data-driven personalisation.
- Klook / Airbnb Experiences – Regional specialisation and cross-sell with accommodation.

These players demonstrate the value of API-centric architectures, mobile-first web design, and community reviews.

## 2.2. Spain: A Record-Setting but Fragmented Market

- 94 million visitors in 2024 — a 10 % jump versus 2023, marking Spain's best year ever.
- €126 billion in tourist spending — up 16 % , so travellers are ready to pay for extra experiences.
- 2025 keeps climbing — Q1 arrivals already +5.7 % over the same period in 2024.
- Domestic city-breaks thrive — Barcelona, Madrid, Seville and Valencia keep hotels full even in low season.

• “Things-to-do” sector is still offline — most small operators book by phone, email or WhatsApp.
• Civitatis leads but covers < 20 % of supply — the majority of experiences aren’t yet on any major platform.

Take-away: visitor and spending records are rising, but activity booking is stuck in analog mode, perfect timing for a unified digital marketplace.

### 2.3. Pain Points & Opportunity

• Travellers: scattered information, manual confirmations, rigid cancellation policies → Opportunity: one-stop web hub with instant booking and flexible change/cancel flows
• Activity operators: low online visibility, high commissions from foreign → Opportunity: direct-to-consumer sales, dynamic pricing, analytics dashboard, lower fees
• Destination-management bodies: limited data on experience quality and demand → Opportunity: unified data layer for policy decisions and targeted promotion

### 2.4. Project Objectives

OBJ-1. The system must store and manage information related to existing activities, whether upcoming, ongoing, or past, with a maximum retention period (in case of issues with refunds).
OBJ-2. The system will automate all activities related to booking activities (contact with the provider, confirmation to the requester, etc.).
OBJ-3. The system must offer multiple options for managing bookings (cancellation, date changes, data inquiries, ownership transfers, etc.).
OBJ-4. The system will provide all necessary options for businesses to appropriately promote their activities.

## 3. Technical Details: Architecture, Technologies & Marketing Approaches

Similar solutions and what we can learn from them:
TripAdvisor
Tech. Runs on AWS with micro-services in Go and Java. Catalogue stored in MySQL shards; search powered by ElasticSearch. Offers a public REST API and an internal GraphQL layer for its own apps.
Marketing. Dominates SEO for “things to do in …” keywords and cross-sells activities to Tripadvisor’s lodging audience.

GetYourGuide
Tech. Entirely on Google Cloud with Kubernetes. Core services in Go; analytics and recommendations handled by Python + BigQuery. Internal messaging travels through Pub/Sub.
Marketing. Heavy Google-Ads spend and a two-tap mobile checkout. Retains users via personalised e-mails and push notifications.

Civitatis
Tech. Migrating from a PHP monolith to Java with PostgreSQL. Content served from AWS Madrid via CloudFront; search implemented with ElasticSearch.
Marketing. Owns Spanish-language SEO thanks to extensive destination guides, offers lower commissions to attract small local operators, and runs radio and TV campaigns in Spain and Latin America.

Recurring patterns
• Micro-services and event buses for scalability.
• Long-tail SEO and local content to capture cheap traffic.
• Self-service onboarding so any operator can list an activity in minutes.

Opportunity for our project
Thousands of small suppliers are still offline and big platforms charge 20–30 % commission. Using a full-stack Node.js on the back end and React on the front—plus a lower introductory fee, we can mimic the proven technical patterns while winning on price and local proximity.

Planned marketing approach
We will build hyper-local SEO pages (“hidden gems in <neighbourhood>”), launch a referral credit system, and partner with city tourism boards and boutique hotels for embedded booking widgets. New operators receive six months at 0 % commission and a real-time analytics dashboard, letting us grow supply fast while keeping acquisition costs down.

# 4. Technology Landscape & Rationale

## 4.1. Managed-services first (PaaS / SaaS)

We use managed services (Azure PostgreSQL, Confluent Cloud, Front Door) to offload operational tasks like patching and scaling, allowing us to focus on business logic while ensuring high availability.

## 4.2. Technology homogeneity – full-stack TypeScript

React (frontend) and NestJS (backend) share TypeScript, linting rules, and tooling, streamlining development and code reuse.

## 4.3. Elasticity and cost efficiency

Azure Container Apps and Confluent Cloud auto-scale (even down to zero when idle), making them cost-efficient for projects with unpredictable traffic.

## 4.4. Security by design

Strong security: Front Door (WAF) blocks attacks at the edge, API Management enforces JWT validation, and secrets are stored in Azure Key Vault—never in code.

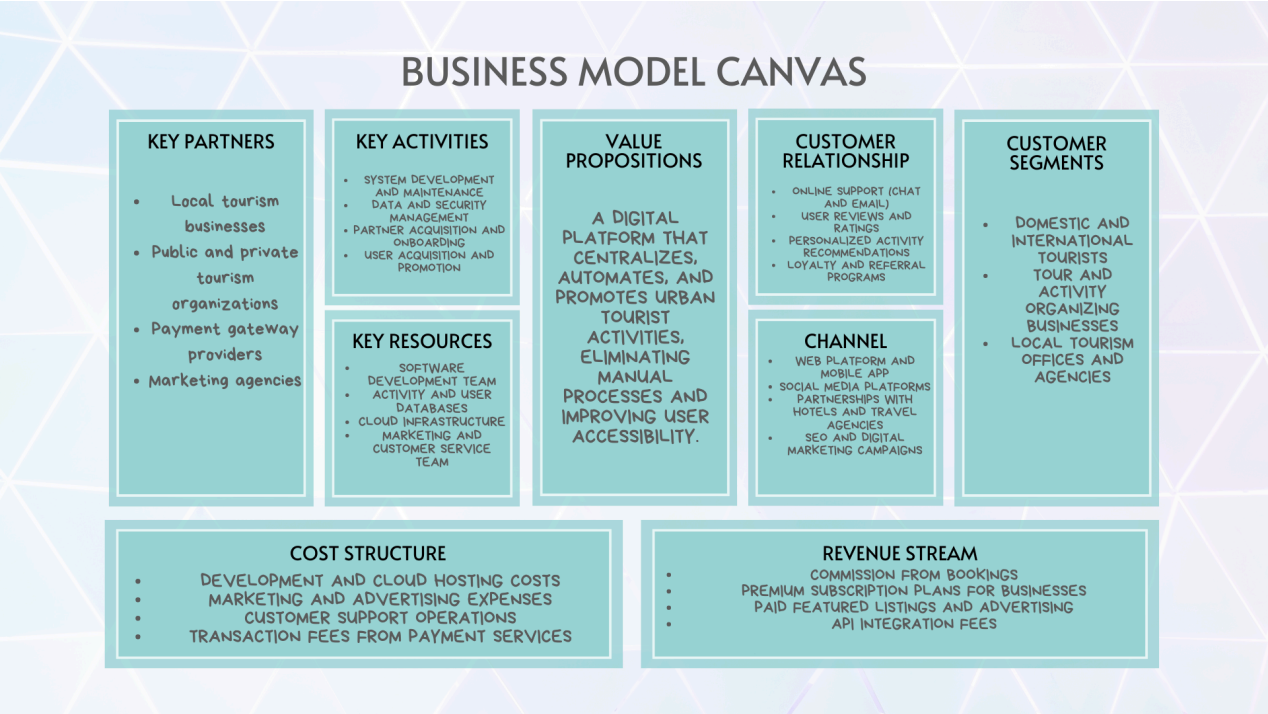
## 4.5. Built-in observability

Monitoring via OpenTelemetry in both frontend and backend, sending traces and metrics to Azure Monitor/Grafana.

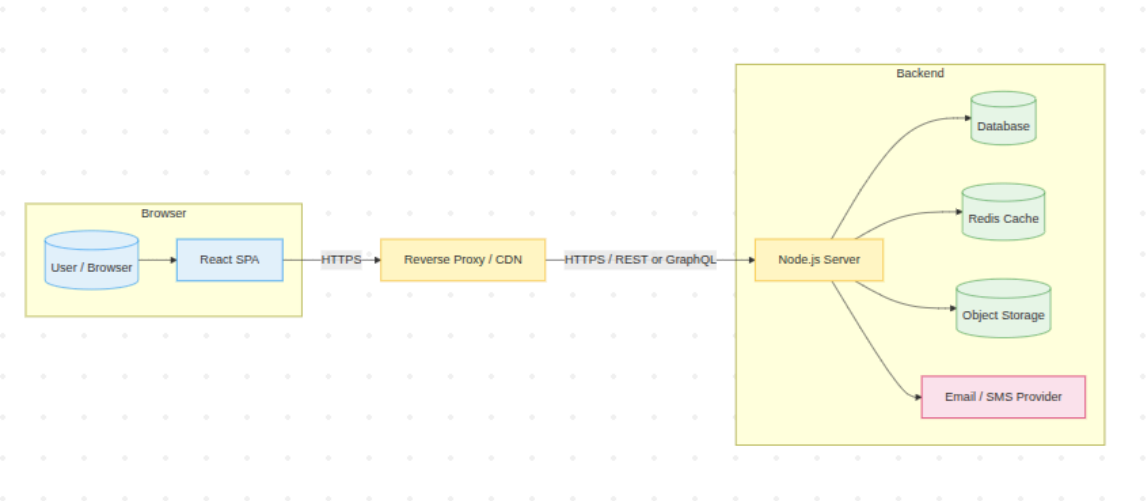
## 4.6. Production-ready, portable architecture

Containerized TypeScript services are portable—easily migrated to AKS, on-prem Kubernetes, or other clouds with minimal changes.

## 5. Business Model Canvas

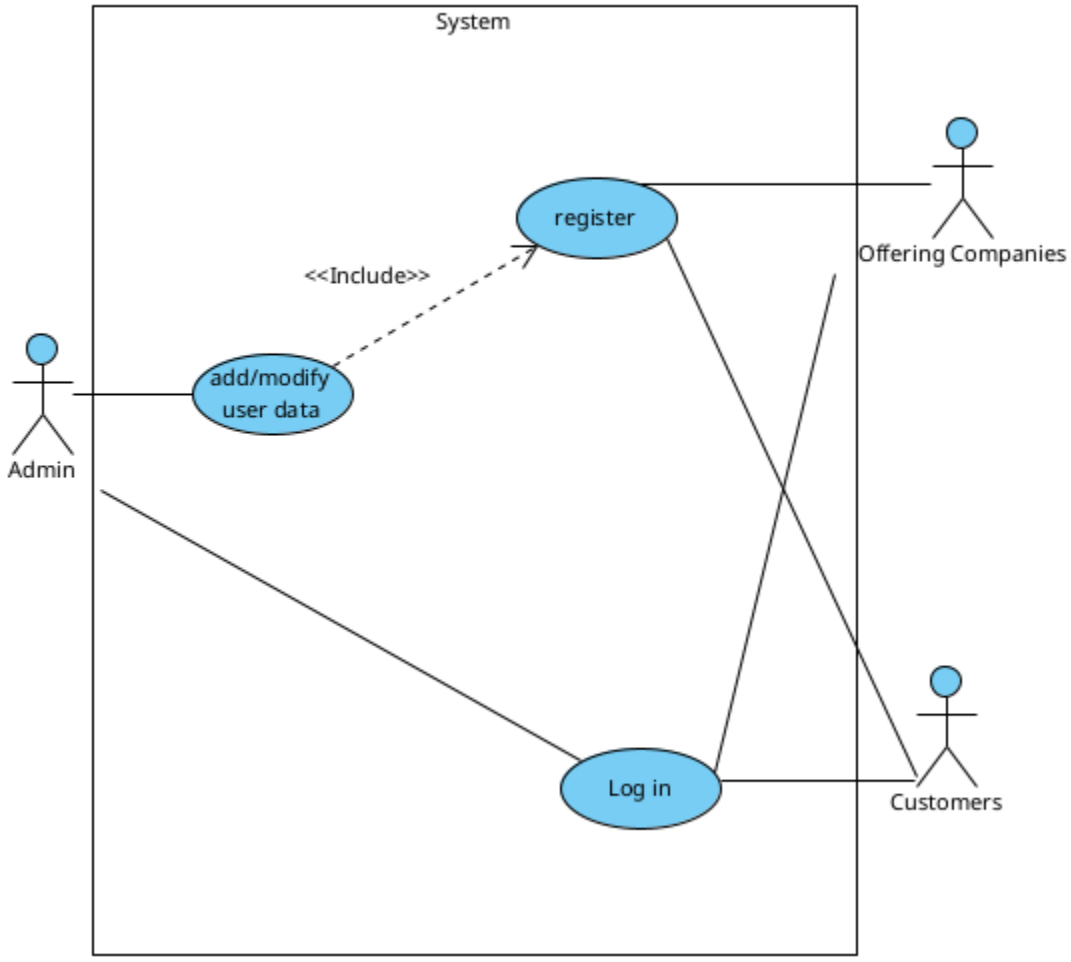



## 6. Solution Architecture Diagram

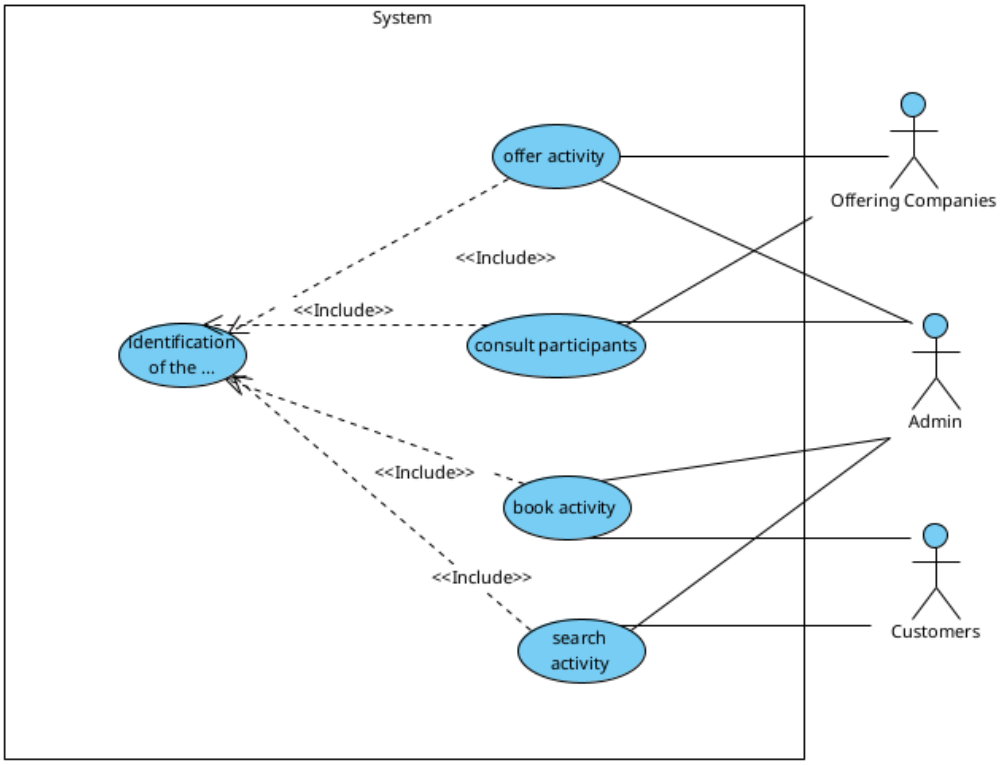


# 7. Project Use-Case Diagrams

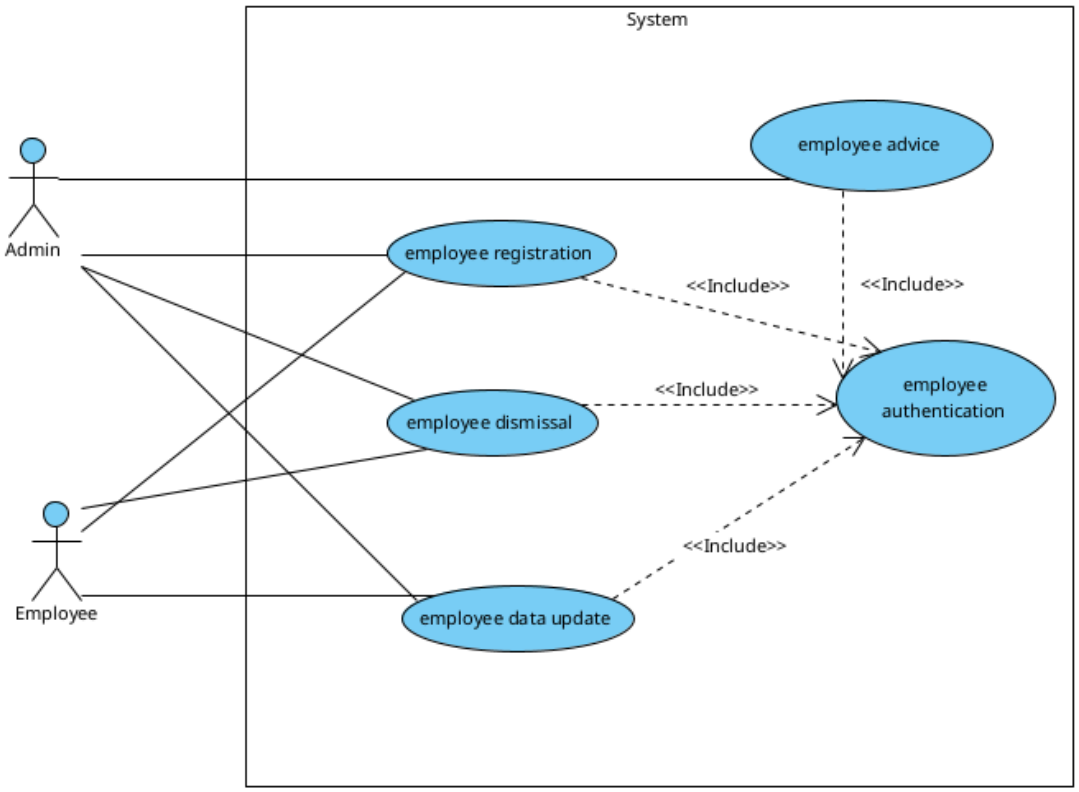
## 7.1. User Management



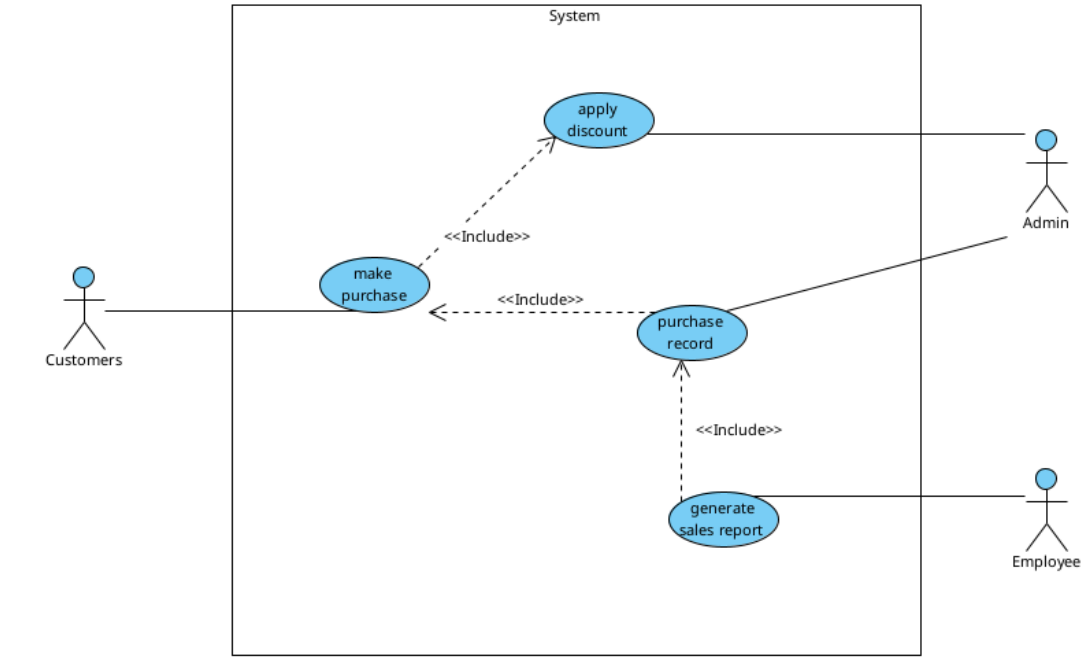

## 7.2. Ticket and Activity Management


System
offer activity
Offering Companies
<<Include>>
<<Include>>
consult participants
Identification of the ...
Admin
<<Include>>
book activity
<<Include>>
search activity
Customers

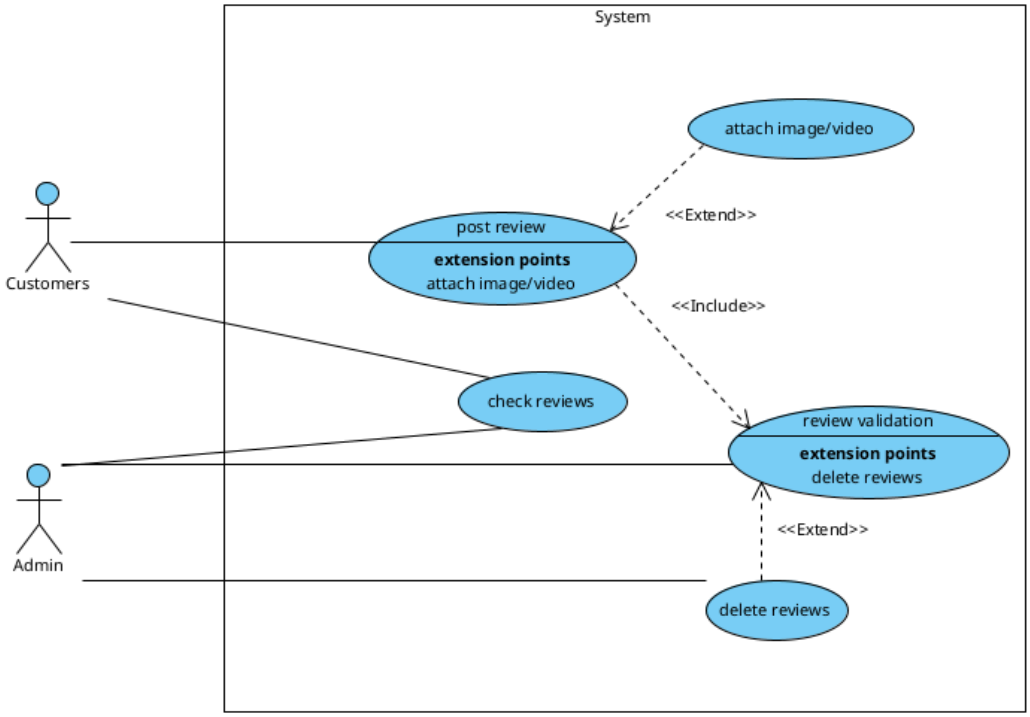
## 7.3. Employee Management



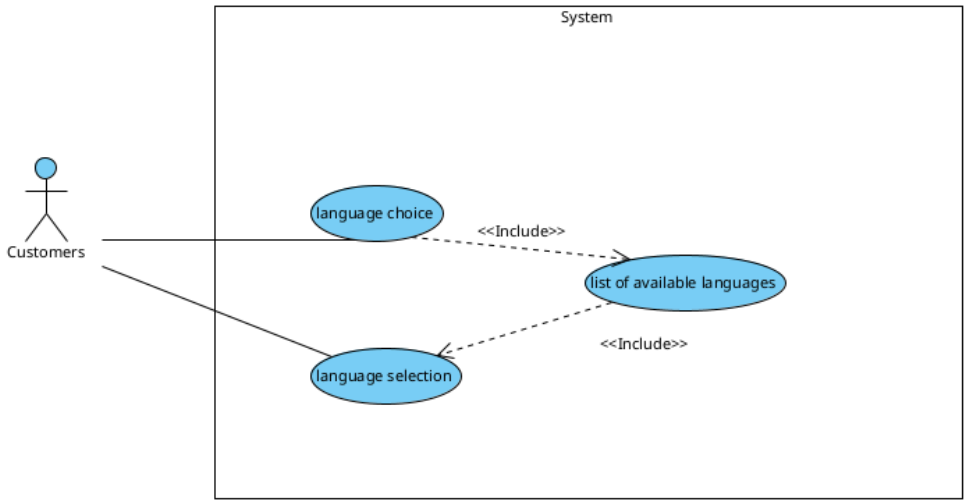
## 7.4. Payment Systems Management



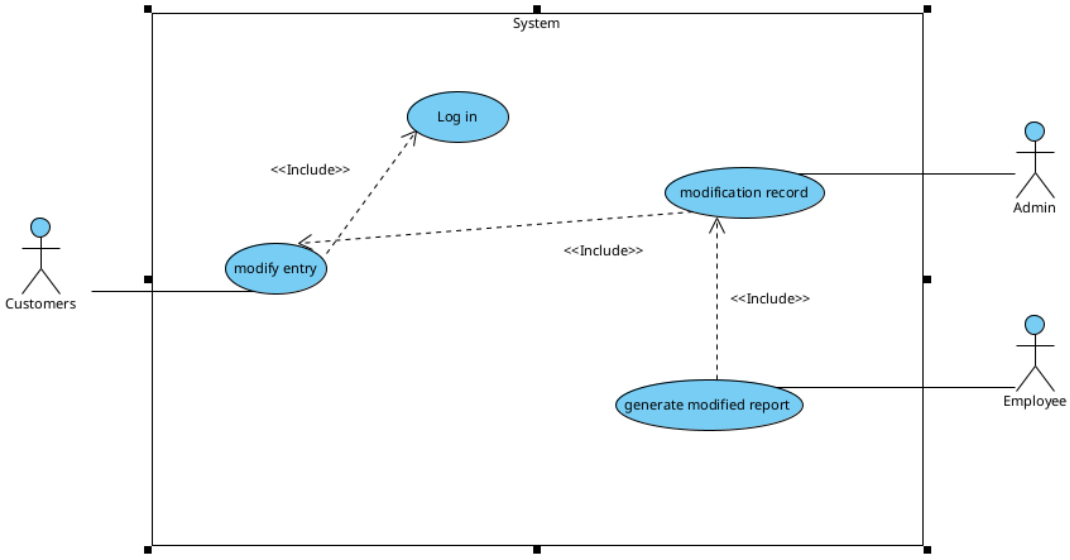

## 7.5. Review Management



## 7.6. Multilingual Support Management

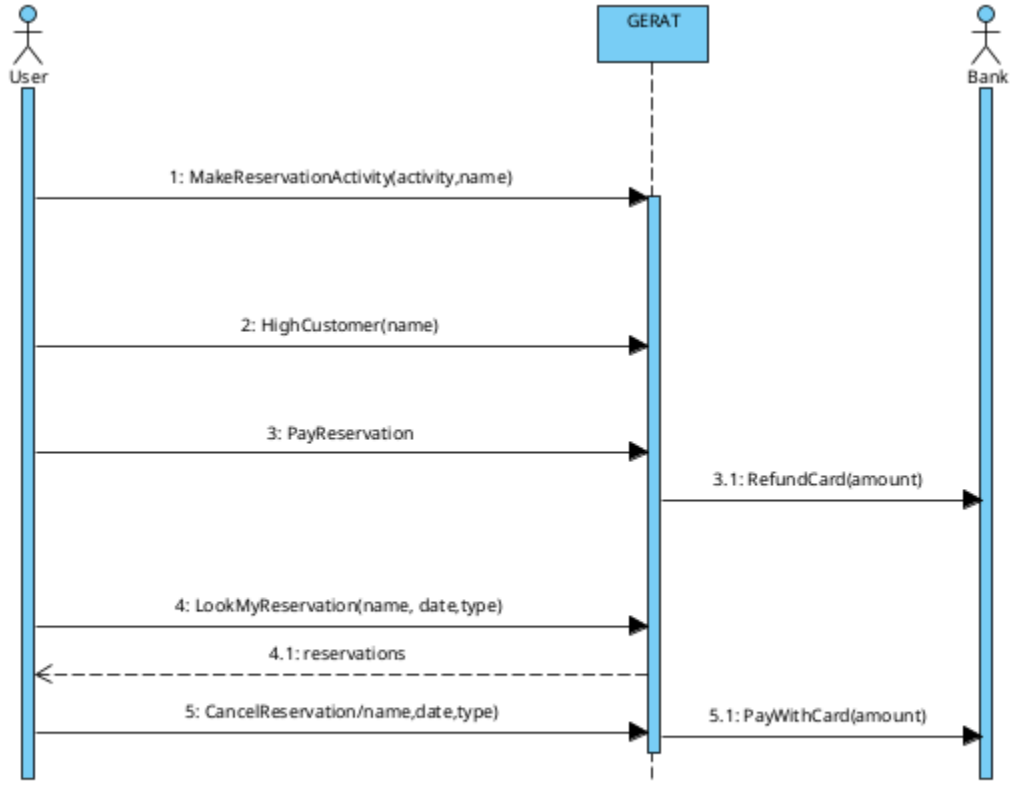



## 7.7. Booking Management

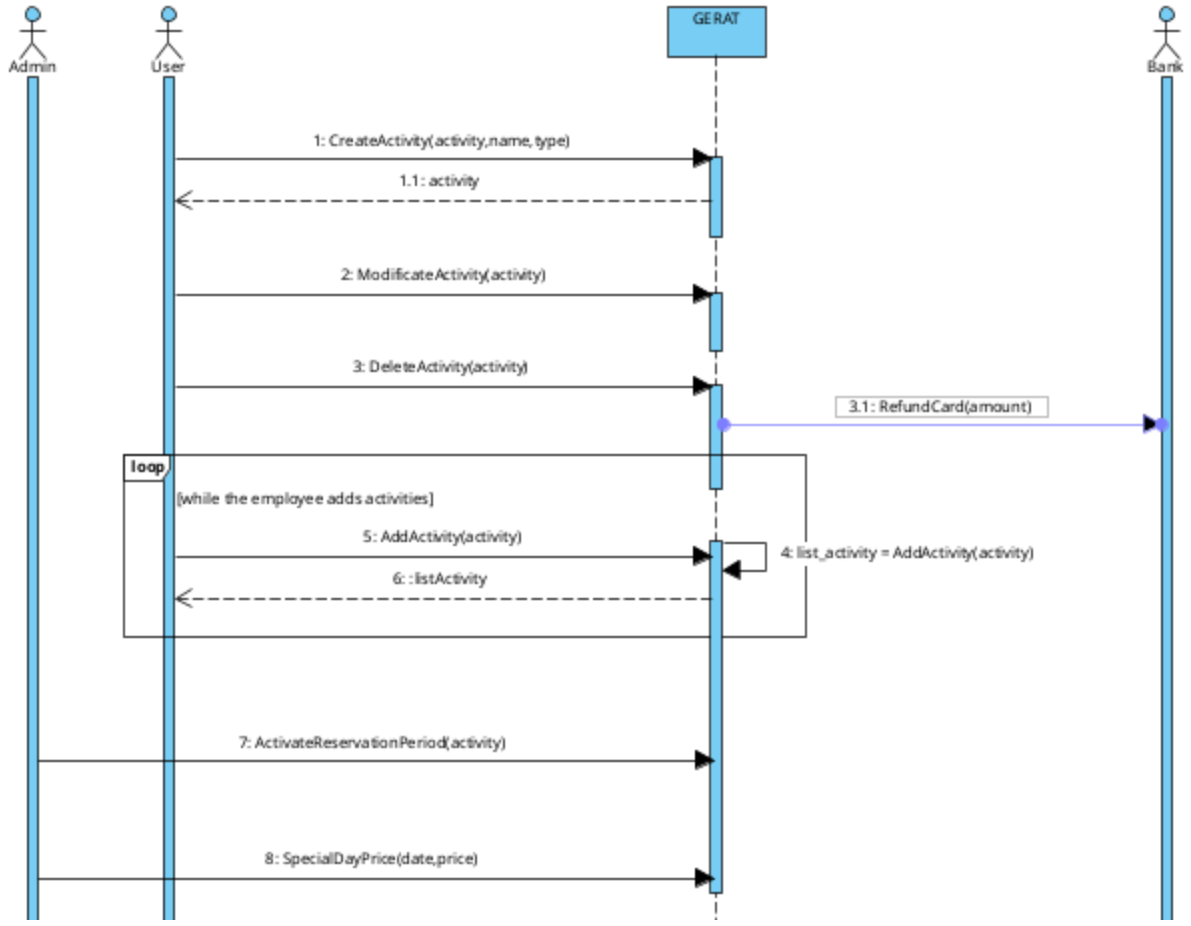


# 8. API Documentation (OpenAPI in SwaggerHub & Functional Flows )

## 8.1. Reservation Management



## 8.2. Activity Management



## 8.3. User Access Management