Department of Computer Science.
Faculty of Mathematics and Computer Science.
Batna 2 University.

ING - 3 AI
2024-2025

Distributed Databases

Project

Project Proposal: Cultural Trip Management System (CTMS)

The goal of this project is to develop a web application that allows tourism agencies to organize, schedule, and manage cultural trips. Each agency will have its own dedicated instance of the application to announce events, manage bookings, and communicate with customers. The application will leverage advanced database concepts, including **Object-Relational Mapping (ORM)**, **distributed databases**, and **fragmentation techniques** (horizontal and vertical), using the **Oracle Database System**.

Objective:

- Develop a web application that allows tourism agencies to manage their cultural trips.
- Implement separate instances of the application for each agency to ensure data isolation.
- Utilize the Oracle Database System with Object-Relational concepts.
- Implement distributed database concepts, including horizontal and vertical fragmentation.
- Provide a user-friendly interface for agencies and their customers.
- Enable real-time communication between agencies and customers.

1. System Features

1.1. User Roles

- **Administrator (Agency Owner):**
  - Manage agency profile.
  - Add, update, and delete cultural trips.
  - Manage customer bookings.
  - Communicate with customers.
- **Customers:**
  - Browse available cultural trips.
  - Book and manage reservations.
  - Communicate with the agency.
  - Provide feedback and ratings.

1.2. Key Functionalities

1. **Agency Management:**
   - Each agency has a dedicated instance of the application.
   - Secure authentication and authorization mechanisms.

2. **Trip Management:**
    - Agencies can create, update, and delete cultural trips.
    - Specify trip details such as location, date, itinerary, price, and available slots.
3. **Booking System:**
    - Customers can book trips and receive confirmations.
    - Agencies can manage reservations and cancellations.
4. **Communication Module:**
    - Messaging system between agencies and customers.
    - Notifications for booking confirmations and updates.
5. **Feedback and Ratings:**
    - Customers can leave reviews for trips.
    - Agencies can respond to feedback.
6. **Database Management:**
    - Implementation of Object-Relational concepts in Oracle.
    - Utilization of distributed database techniques.

2. Technical Specifications

2.1. Backend Development

- Programming Language: **Node.js (Express.js)** or **Java (Spring Boot)**
- Database: **Oracle Database 19c**
- Object-Relational Concepts:
    - Use of Object Types, Nested Tables, and REF Types.
- Distributed Database Concepts:
    - **Horizontal Fragmentation:** Data is divided by agencies, ensuring each agency has access only to its data.
    - **Vertical Fragmentation:** Trip-related data is separated from customer-related data for efficient access.

2.2. Frontend Development

- Framework: **React.js** or **Angular**
- UI Components: **Bootstrap / Material-UI**
- Responsive design for (mobile and) desktop views.

2.3. Database Design

- **Agency Table** (Stores agency details)
- **Trips Table** (Stores trip information, fragmented by agency)
- **Customers Table** (Stores customer details)

- **Bookings Table** (Stores reservations, linking customers and trips)
- **Messages Table** (Stores communication records between agencies and customers)
- **Reviews Table** (Stores feedback and ratings from customers)

3. Implementation Plan

1. **Phase 1: System Design**
    - Define database schema and fragmentation strategy.
    - Design the user interface.
    - Define API endpoints.
2. **Phase 2: Backend Development**
    - Develop RESTful API for the application.
    - Implement authentication and authorization.
    - Configure the Oracle database with object-relational features.
3. **Phase 3: Frontend Development**
    - Develop user interfaces for agencies and customers.
    - Implement interaction with the backend.
4. **Phase 4: Integration and Testing**
    - Perform unit testing and integration testing.
    - (Deploy the application and test distributed database functionality.) // for the future
5. **Phase 5: Deployment and Maintenance** // for the future
    - Deploy on a cloud-based or on-premises environment. // for the future
    - Provide documentation for future maintenance. // for the future

4. Expected Outcomes

- A fully functional web application for tourism agencies.
- Implementation of Oracle's object-relational features.
- Demonstration of distributed database concepts.
- Secure, scalable, and user-friendly experience for agencies and customers.

5. Conclusion

This project provides a comprehensive learning experience in web development, database management, and distributed systems. By working on this project, students will gain hands-on experience with Oracle's object-relational concepts, distributed databases, and full-stack development.