

Event Management System

CS544 - Enterprise Architecture

Final Project

Awais Waheed - 618677

Requirements

- 1- Have at least 4 entities, with relationships between them.
 - Business Entities: User, Admin, Staff, Customer, Location, Category, Event, Ticket.
 - User is parent class for Admin, Staff and Customer
 - Event has 1-1 unidirectional with Location, Category
 - Event has M-M unidirectional with Staff
 - Customer has 1-M unidirectional with Ticket
 - Ticket has 1-1 unidirectional with Event
 - Payment is embedded into Ticket
 - Address is embedded into Customer
 - Others: Operations, Exceptions
- 2- At least 2 entities must have full CRUD operations (with REST controller, Service, and Repository)
 - Most of the entities have full CRUD operations other than Operations and exceptions
- 3- Have at least one Dynamic Query, one Named query, and one Criteria Query.
 - Named: Get Upcoming Event API
 - Dynamic: Find event with min seats API
 - Criteria API: All the other Event searching APIs
- 4- At least one query involving 3 entities.
 - Get All Events has Criteria API where you can search an event based on Location, Category and Staff.
- 5- You must have JMS (with two projects, one being a sender and the other a receiver of the message)
 - When a ticket is purchased, a message is sent to Consumer which saves that message in the H2 in-memory database. You can get all messages via localhost:8081/api/v1/messages
- 6- You must have content negotiation.
 - Added
- 7- You must have Spring Security with at least one API that is public, and at least two roles (and one API for each role).
 - Login, Refresh Token and Customer Signup are public APIs
 - Rest of the APIs require JWT
- 8- Demonstrate AOP with one use case (using AspectJ)
 - Added API Logging Aspect
 - Added saving all API calls to database Aspect
 - Added saving all exceptions to database Aspect

9- Upload your work to GitHub, but submit your project to Sakai before the deadline, what is uploaded to Sakai will be graded. You will get extra points if you manage to deploy your work to any platform.

- Github Links
 - [Event Management System REST Service](#)
 - [Consumer](#)
- Failed successfully in deployment

10- Demonstrate two profiles (development using H2, and deployment using MySQL)

- dev Profile added with H2 support
- prod Profile added with MySQL support
- data Profile added to generate tables and initial data. **Run in conjunction with dev/prod**

11- Make sure your REST API is stateless.

- Both services are stateless

12- Make sure you address concurrency in all your entities, and demonstrate both types of locking.

- All Entities use optimistic locking.
- While purchasing ticket, Pessimistic Force Increment is used on event entity

Deployment

1. Created database on AWS RDS
2. Created EC2 instance that hosts EMS REST Service and Artemis. [Swagger](#)
3. Created another EC2 instance that hosts Consumer. [Swagger](#)

Others

13- Swagger added to the REST service.

14- Postman Collection is also added beside this document.

15- Role based access set to the best of knowledge of how the APIs would be used.

16- These tables represent all the user accounts created by the DataGenerationService. The passwords are encoded when stored, but these are their plain text values. **The tables and these values will be generated if you run the data profile along with dev/prod.**

Here is the list of roles, emails, and passwords for all the users created in the given code:

Admins

Role	Email	Password
Admin	admin1@example.com	password_1234
Admin	admin2@example.com	password_1234
Admin	admin3@example.com	password_1234

Customers

Role	Email	Password
Customer	customer1@example.com	password1
Customer	customer2@example.com	password2
Customer	customer3@example.com	password3
Customer	customer4@example.com	password4
Customer	customer5@example.com	password5
Customer	customer6@example.com	password6
Customer	customer7@example.com	password7
Customer	customer8@example.com	password8
Customer	customer9@example.com	password9
Customer	customer10@example.com	password10

Staff Members

Role	Email	Password
Event Planner	email_1@example.com	password_1234
Logistics Manager	email_2@example.com	password_1234
Catering Manager	email_3@example.com	password_1234
Marketing Specialist	email_4@example.com	password_1234
Tech Support	email_5@example.com	password_1234
Registration Coordinator	email_6@example.com	password_1234
Event Planner	email_7@example.com	password_1234
Logistics Manager	email_8@example.com	password_1234
Catering Manager	email_9@example.com	password_1234
Marketing Specialist	email_10@example.com	password_1234
Tech Support	email_11@example.com	password_1234
Registration Coordinator	email_12@example.com	password_1234
Event Planner	email_13@example.com	password_1234
Logistics Manager	email_14@example.com	password_1234
Catering Manager	email_15@example.com	password_1234
Marketing Specialist	email_16@example.com	password_1234
Tech Support	email_17@example.com	password_1234
Registration Coordinator	email_18@example.com	password_1234
Event Planner	email_19@example.com	password_1234
Logistics Manager	email_20@example.com	password_1234