

SCALABLE TICKET SELLER

A PROJECT BY

THEJAS BHARADWAJ

SURAJ VISVESWARAIYA

Introduction

- Welcome to a groundbreaking venture aimed at redefining the ticketing experience for sports enthusiasts.
- Our vision is to create a ticket agency that excels in handling events of all sizes, from local matches to international tournaments, with effortless ease.
- At the heart of this project are four key pillars: Scalability, Concurrency, Reliability, and Performance.
- Our mission is to provide users with a seamless, reliable, and high-performance platform, ensuring they enjoy swift and frustration-free ticket purchasing, from event selection to checkout.
- We're committed to robust data persistence, user-friendly interfaces, regulatory compliance, real-time support, and comprehensive documentation.
- Join us on this transformative journey as we set out to revolutionize the world of ticketing.

Use Cases - Actors and Actions

This project seeks to address these challenges by leveraging the power of Akka.

1) **EventManager Actor:** Manages information about events, venues, and schedules.

- Create new events.
- Update event details.
- Retrieve event information.

2) **TicketSeller Actor:** Handles ticket sales for a specific event.

- Process ticket purchases.
- Update available ticket inventory.
- Handle refunds or cancellations.

3) **CustomerActor:** Represents individual customers interacting with the system.

- Purchase tickets for events.
- Receive order confirmations.

Use Cases - Actors and Actions

- 4) **NotificationActor**: Sends notifications to customers.
 - Send order confirmations.
 - Notify customers about upcoming events.
 - Handle communication preferences.
- 5) **PaymentProcessorActor**: Handles payment processing for ticket purchases.
 - Authorize and process payments.
 - Handle payment failures.
 - Manage payment transactions.
- 6) **InventoryManagerActor**: Manages the overall ticket inventory for all events.
 - Track available tickets.
 - Handle restocking of tickets.
 - Monitor and alert on low inventory.

Data Sources

- 1) Event and ticket information from event organizers.
- 2) User profiles and transaction data from customer interactions.

Note : The data magnitude will vary depending on the number of events and users. We anticipate handling tens of events and millions of user records.

Milestones/Sprints

1. System Architecture and Designing - Nov 6 - Nov 12
2. Implementation of Actors - Nov 12 - Nov 25
3. Test cases development - Nov 25 - Nov 30
4. Unit Testing - Dec 1
5. Performance Enhancement - Dec 2 - Dec 4
6. Acceptance Testing - Dec 5
7. Deployment of the system - Dec 6

Acceptance Criteria

1) Scalability Criteria:

- The system must seamlessly handle events ranging from 1,000 attendees to 100,000 attendees, with consistent performance and response times.
- The system should be able to add new events and event details without any significant impact on existing services.

2) Concurrency Criteria:

- The system should be able to handle a peak load of at least 1,000 concurrent users without performance degradation.
- Response times should remain under 2 seconds for 90% of concurrent user interactions.

3) Performance Optimization Criteria:

- Page load times for event listings, seat selection, and checkout should not exceed 1 second for 95% of user interactions.
- The system should be able to process a minimum of 1,000 transactions per minute during peak periods.

4) User Experience Criteria:

- The user interface should be intuitive and easy to navigate, with a user satisfaction rating of at least 4 out of 5.
- Error messages should be descriptive and guide users toward resolving issues effectively.

KEY ASPECTS OF THE PROJECT

- Language: **Scala , Spark.**
- Framework: **Akka** HTTP for building RESTful APIs.
- Database: **PostgreSQL** or **MySQL** for relational data storage.
- **Slick** for database access in a reactive manner

PROJECT PHASES

THE PROJECT WILL BE STRUCTURED IN PHASES, EACH GEARED TOWARDS ACHIEVING SPECIFIC MILESTONES:

1) **SYSTEM DESIGN:** WE WILL DESIGN A SCALABLE ARCHITECTURE THAT CATERS TO EVENTS OF VARYING SIZES AND A USER-FRIENDLY INTERFACE FOR TICKET SALES.

2) **DEVELOPMENT:** WE WILL CREATE THE TICKET AGENCY SYSTEM USING AKKA AND ACTIVATOR, ENSURING IT MEETS THE SPECIFIED REQUIREMENTS WITH THE UNDERLYING LANGUAGE BEING SCALA AND SPARK.

3) **TESTING:** RIGOROUS TESTING WILL BE CONDUCTED TO ENSURE THE SYSTEM'S PERFORMANCE, RELIABILITY, AND SECURITY.

4) **DEPLOYMENT:** THE SYSTEM WILL BE DEPLOYED TO A PRODUCTION ENVIRONMENT, READY TO SERVE SPORTS FANS.

5) **MONITORING AND OPTIMIZATION:** CONTINUOUS MONITORING AND FINE-TUNING WILL BE A PART OF OUR STRATEGY TO MAINTAIN HIGH PERFORMANCE AND RESPONSIVENESS.

PROJECT GOALS

1. On board customers/users into our system(10000 users)
2. Manage Ticket sales(roughly about 100000)
3. Manage concurrency of the users(700 - 1000)
4. Users should be able to request for tickets in numbers
5. Ticket Seller will process user requests
6. Customers will buy tickets on the platform seamlessly
7. Event Manager/ admin will manage the system and performances

We wish to understand the working of Scala Actors and handling concurrency using functional programming!!!