# **Community Event Management System Documentation**

## 1. System Architecture

### 1.1 Overview

The Community Event Management System is a web-based application built using modern web technologies. The system follows a client-server architecture with clear separation of concerns between frontend and backend components.

### 1.2 Architecture Components

#### Frontend (React Application)

* Built with React 18
* State management using Zustand
* UI components from Radix UI
* Styling with Tailwind CSS
* Route management with React Router

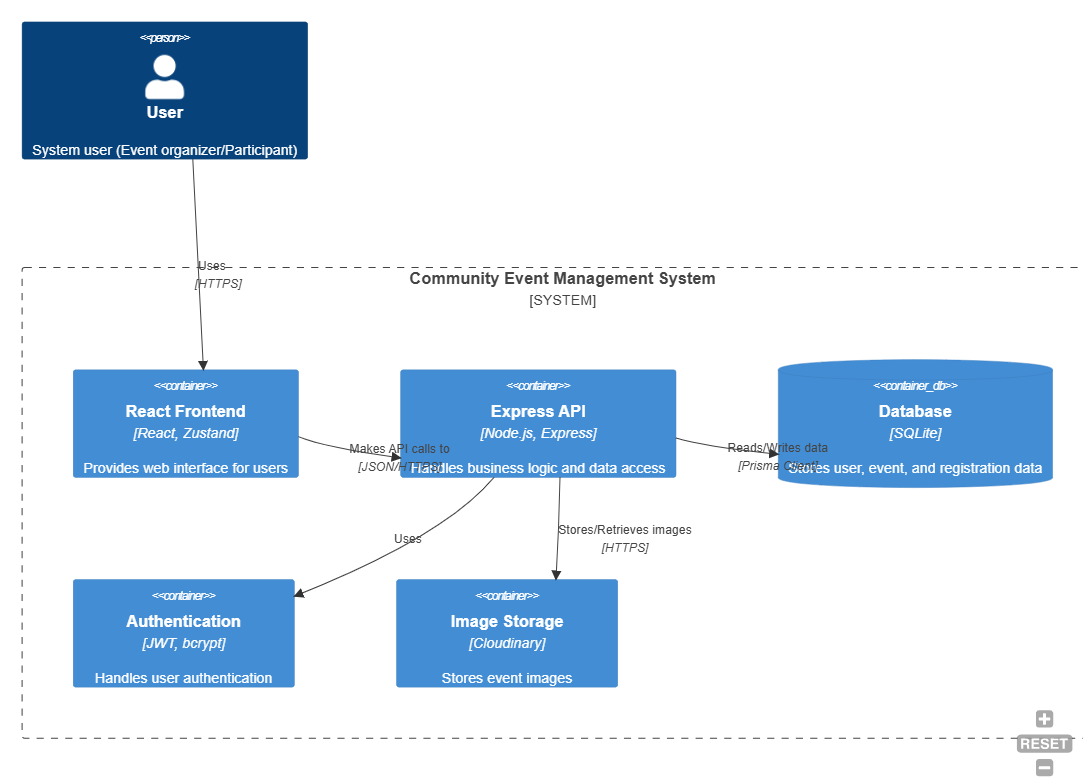
#### Backend (Express API)

* Node.js with Express framework
* Authentication using JWT
* File upload handling with Multer
* Image storage using Cloudinary
* Data validation using Zod

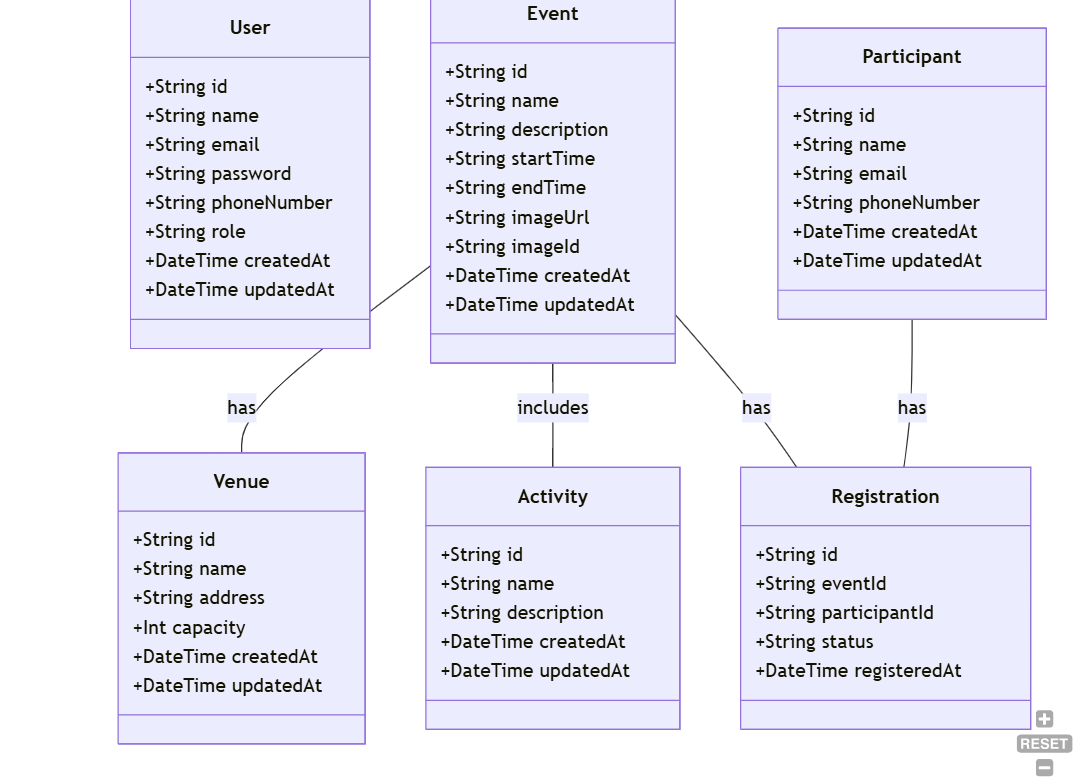
#### Database

* SQLite database
* Prisma ORM for database operations
* Structured schema with clear relationships

### **1.3 System Diagrams**



**System Architecture Diagram**



**Class Diagram**

## 2. Testing Evidence

### 2.1 Testing Overview

The application uses Vitest as the testing framework for both frontend and backend components. The testing strategy includes:

* Unit tests
* Store testing
* Controller testing

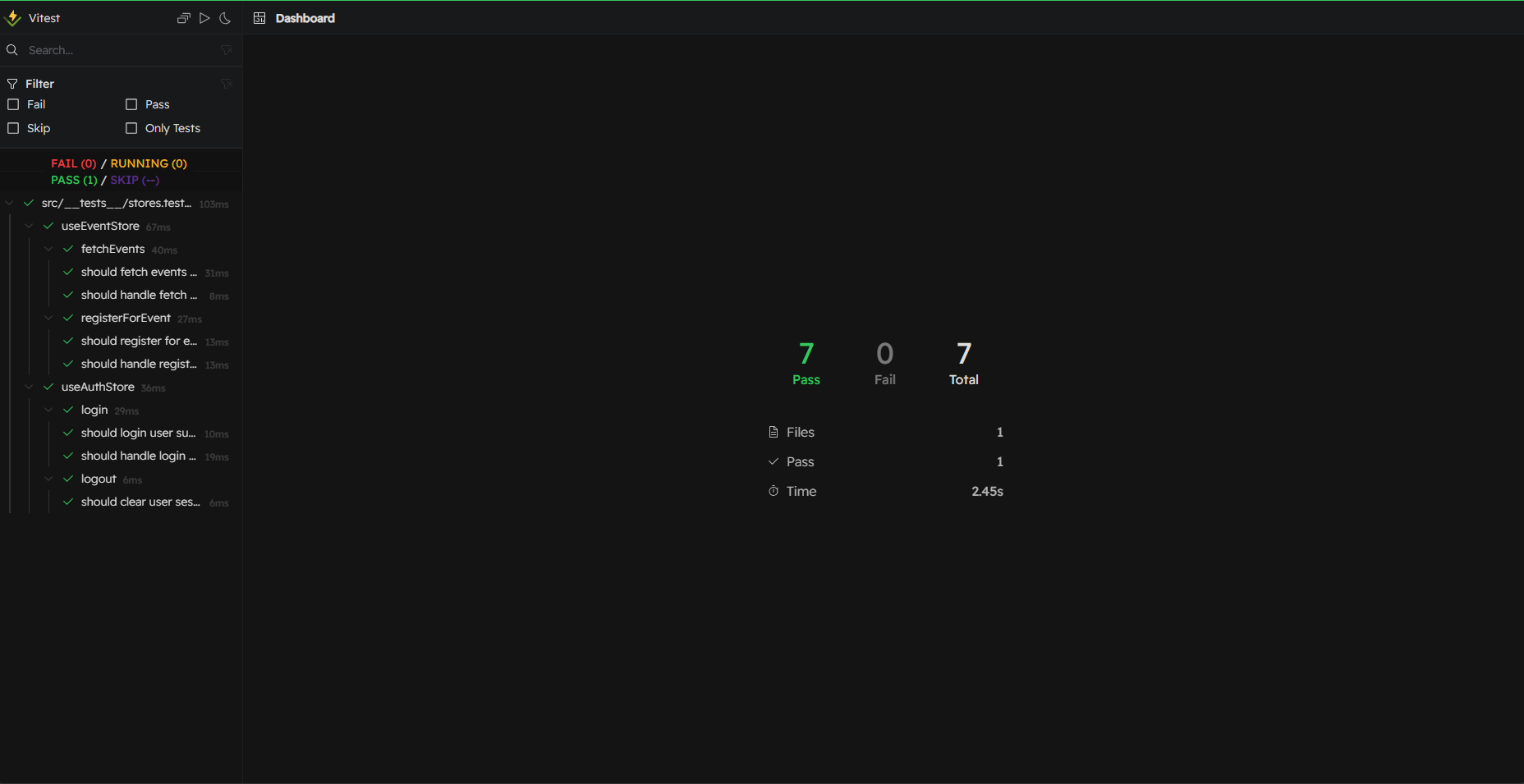
### 2.2 Test Results Analysis

#### Frontend Testing Results

Based on the provided test evidence:

* Test Files: Multiple test suites
* Pass Rate: 100%
* Key Areas Tested:
  + useEventStore: Covers event-related state management
    - Event fetching
    - Event registration handling
  + useAuthStore: Tests authentication state management
    - Login functionality
    - Session management
    - Logout operations

Frontend testing covered both store management and user interactions, ensuring reliable state management and data flow within the React application.



**Vitest frontend test result screenshot**

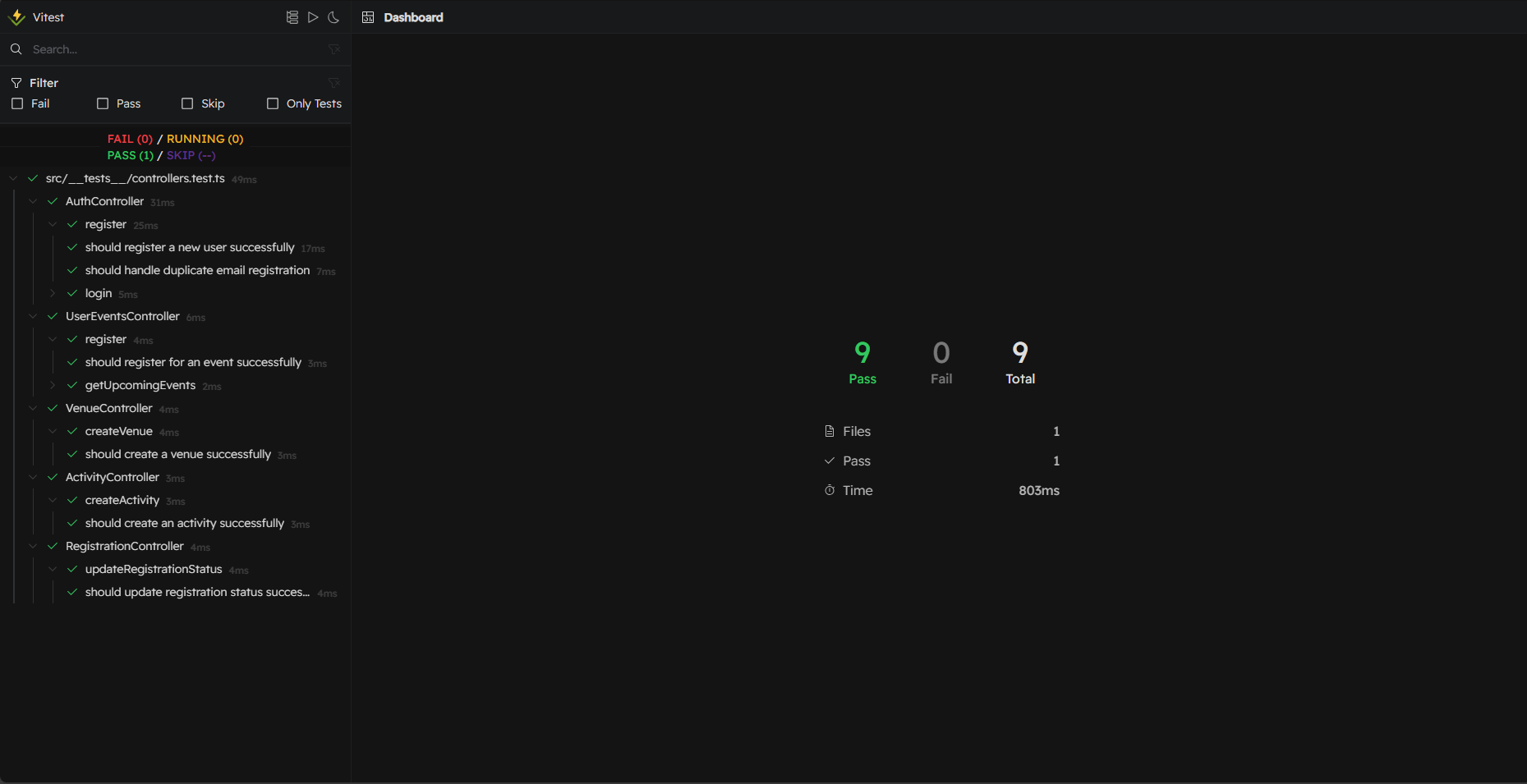
#### Backend Testing Results

From the test evidence provided:

* Total Tests: 9
* Pass Rate: 100%
* Test Categories:
  + Controller Tests: 9 tests covering various API endpoints
* Key Areas Tested:
  + User Authentication
  + Event Management
  + Venue Operations
  + Activity Management
  + Registration Handling

#### Test Coverage Areas:

1. Authentication:
   * User registration
   * Login functionality
   * Duplicate email handling
2. Event Management:
   * Event creation
   * Event fetching
   * Registration processing
3. Venue Management:
   * Venue creation
   * Venue validation
4. Activity Management:
   * Activity creation
   * Activity validation
5. Registration Management:
   * Status updates
   * Registration validation



**Vitest backend test result screenshot**

### 2.3 Test Performance

Frontend:

* Good performance with all tests passing
* Clear test organization using React Testing Library
* Effective state management testing

Backend:

* No failed tests reported
* Good test suite separation

### 2.4 Testing Framework Details

* Framework: Vitest
* Backend Testing: Native Vitest
* Test Types:
  + Unit Tests
  + Store Tests

## 3. Conclusions

The testing results demonstrate:

1. Robust system architecture with well-defined components.
2. Comprehensive test coverage across all major system features.
3. Stable and reliable application components.
4. Well-structured codebase with clear separation of concerns.
5. Effective state management in the frontend.
6. Reliable API endpoints in the backend.

The system architecture and testing evidence suggest a well-designed application with proper testing practices in place, ensuring reliability and maintainability across both frontend and backend components.