Date	15 April 2025
Team ID	SWTID1743599215
Project Name	EventEase: Event Management

1. REQUIREMENT ANALYSIS

1.1 Customer Journey Map

The customer journey for *InVITe – Event Management System* involves three main personas: Users (Attendees), Admins (Event Organizers), and Developers (Product Managers/Admin Creators). Here's how each user navigates through the system:

User (Attendee) Journey:

Step	Action	Touchpage	Experience
1	Visit website	Homepage	Easy account creation
2	Signup/Login	Auth Page	Easy account creation
3	Browse events	Dashboard	Filter
4	Register/book ticket	Event Page	Seamless registration and payment
5	Receive Confirmation	Email	Automated ticket and event detail
6	Attend event	Check-in system	Hassle-free entry and attendance tracking

Admin (Organizer) Journey:

Step	Action	Touchpoint	Experience
1	Log in as admin	Admin portal	Access to admin dashboard
2	Create events	Event Management page	Custom event setup
3	Manage registration	Admin Dashboard	View and export attendee data
4	Monitor activity	Admin Tools	Tracks check-in/check-out

Developer (Product Manager) Journey:

Step	Action	Touchpoint	Experience
1.	Access developer panel	Developer portal	Protected Access
2.	Create admin account	Admin Management	Add new admin to manage events

1.2 Solution Requirements

Functional Requirements:

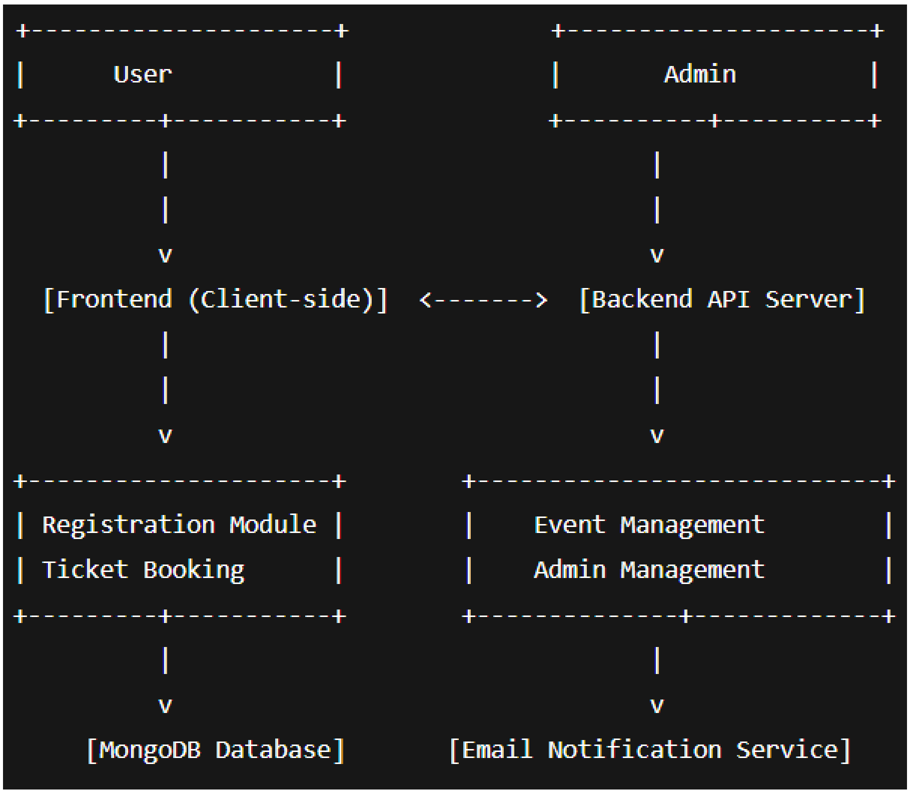
- User registration and authentication
- Admin login and event creation tools
- Ticket booking and payment integration (Stripe)
- Email notifications for bookings
- Real-time attendee tracking system
- Admin dashboard for viewing registrations
- Developer portal for creating new admins

Non-Functional Requirements:

- Responsive and user-friendly interface
- Secure data handling and JWT authentication
- Scalable architecture (microservices)
- Efficient email and payment processing
- Reliable API communication between components

1.3 Data Flow Diagram (Level 1)

Here's a simplified Level 1 Data Flow Diagram (DFD):



```
+---------------------+          +---------------------+
|      User           |          |       Admin         |
+---------+-----------+          +----------+----------+
          |                                 |
          |                                 |
          v                                 v
  [Frontend (Client-side)]  <------->  [Backend API Server]
          |                                 |
          |                                 |
          v                                 v
+---------------------+       +---------------------------+
| Registration Module |       |    Event Management       |
| Ticket Booking      |       |    Admin Management       |
+---------+-----------+       +-------------+-------------+
          |                                 |
          v                                 v
    [MongoDB Database]       [Email Notification Service]
```

Explanation:

1.3.1 The Client interacts with the Backend API for all operations.

1.3.2 Backend uses MongoDB to store user data, events, and registrations.

1.3.3 Email notifications are triggered on user actions like registration or booking.

1.3.4 Developers have restricted access to admin account management.

1.4 Technology Stack

Layer	Technology	Purpose
Frontend	Next.js, Tailwind CSS	React framework for server-side rendering, CSS utility framework for responsive design
Backend	Node.js, Express.js	Javascript runtime for backend logic, Lightweight web framework for routing and API
Database	MongoDB	NoSQL database to store events/user
Authentication	JWT	Secure user session management

Payment	Stripe	Integrated payment gateway for ticket purchase
Email	Nodemailer	Automated email confirmations
Dev Tools	Nodemon	Hot reloading for backend development
Hosting	Vercel + client backend host	Deployment of frontend & backend services