Index page

Introduction

Welcome to Eventique your ultimate destination for seamless event planning and unforgettable experiences! We specialize in bringing people together through well-organized, innovative, and engaging events tailored to meet diverse needs, whether you're planning a corporate conference, a community gathering, or a social celebration.Let's make every event extraordinary, together!

1.1 Purpose

The Purpose section explains the reason for creating the Event Management System (EMS) and defines the objectives this document seeks to achieve. It provides an overview of the EMS's goals and its role in streamlining event planning, registration, and management.

The main goals could include:

- Improved Event Organization: Automating the scheduling, attendee registration, ticketing, and other processes involved in organizing an event.
- Enhanced User Experience: Providing an easy-to-use platform for event organizers and attendees.
- Real-time Notifications: Ensuring that event updates and important alerts are promptly communicated to all users.
- Data Insights: Offering analytics to assess the success of events and guide future decisions.

The purpose section sets the foundation for the rest of the document, explaining what the software will do and why it is important.

1.2 Scope

The Scope section defines the boundaries of the Event Management System. It specifies the functional areas the system will cover and what it will not cover. This ensures clarity regarding the system's capabilities, which could include:

- Inclusions:
  - Event creation and management: Allows organizers to set up event details such as schedules, locations, and speakers.
  - Attendee registration: Facilitates attendee sign-ups for events with secure payment options.
  - Ticketing: Provides attendees with digital or physical tickets after registration.
  - Notifications and Alerts: Sends reminders and updates to attendees and organizers.
  - Reporting and analytics: Tracks event performance, attendance rates, and other key metrics.
- Exclusions:
  - Physical venue management (like venue booking, catering).
  - Non-event-related services (such as product sales unrelated to the event).

The Scope sets the limitations and focuses on what the system will achieve in the context of event management.

1.3 Definitions and Acronyms

The Definitions and Acronyms section provides a glossary of terms, abbreviations, and technical jargon used in the SRS document to ensure consistency and clarity. This helps avoid confusion for various stakeholders (developers, project managers, users, etc.). Here are some examples:

- EMS: Event Management System – The software platform that facilitates the creation, management, and analysis of events.
- API: Application Programming Interface – A set of protocols that allows different software applications to communicate with each other.
- UI: User Interface – The visual elements through which users interact with the EMS.
- GDPR: General Data Protection Regulation – A legal framework that governs the processing of personal data within the European Union.
- Admin: Administrator – A user with full access to the system, responsible for managing event settings, users, and reports.
- Attendee: A user who registers to participate in an event.
- Organizer: A user responsible for creating, managing, and overseeing an event.

This section ensures that readers understand the technical terminology and acronyms used throughout the document.

Project Overview

- The EMS is a web-based application designed to simplify event management by integrating with services like payment gateways and social media platforms.
- Key features include event creation, attendee registration, notifications, and analytics.
- The system supports different user roles: event organizers, attendees, and administrators.
- It operates on a server with an ASP.NET runtime, using SQL Server for the database, and is accessible via modern web browsers.
- The system ensures compliance with GDPR, supports third-party integrations, and offers comprehensive user documentation.

🔑 Key Objectives:

- **Streamline Event Management**
- **Provide an intuitive platform that simplifies the creation, management, and organization of events for both small and large-scale activities.**
- **Enhance User Experience**
- **Ensure seamless interactions for organizers and attendees through user-friendly interfaces and efficient workflows.**
- **Enable Secure Transactions**
- **Integrate secure payment gateways for ticket sales, ensuring data encryption and compliance with privacy laws.**
- **Automate Notifications and Updates**
- **Provide real-time communication with attendees via email and SMS for event updates and reminders.**
- **Generate Actionable Insights**
- **Offer detailed analytics and reporting to organizers, enabling them to assess event performance and make informed decisions for future events.**
- **Ensure Scalability and Reliability**
- **Design the system to handle high user traffic and ensure consistent availability with robust backend architecture.**
- **Integrate Seamlessly with Third-Party Services**
- **Enable smooth integration with payment processors, social media platforms, and marketing tools to extend functionality and improve user outreach.**

💡Overall Description

2.1 Product Perspective

The EMS is a standalone web-based application designed for seamless integration with third-party services like payment gateways and social media platforms.

2.2 Product Features

- **Event creation and customization.**
- **Real-time notifications and updates.**
- **Attendee registration and ticketing management.**
- **Post-event analytics and reporting.**

2.3 User Classes and Characteristics

- **Event Organizers: Manage event details and track attendee data.**
- **Attendees: Register and participate in events.**
- **Administrators: Oversee operations and ensure system reliability.**

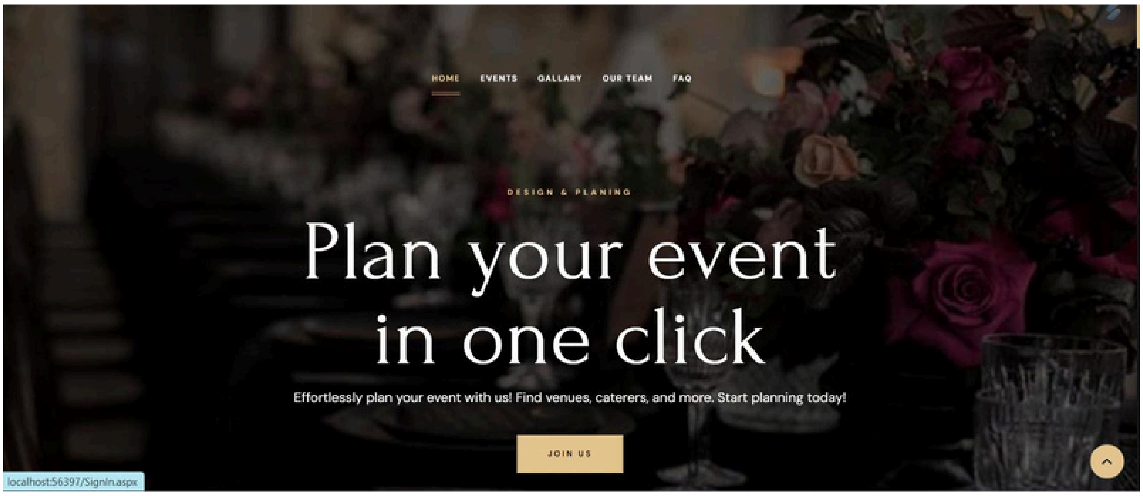
2.4 Operating Environment

- **Server: Windows Server or Linux with ASP.NET runtime.**
- **Database: SQL Server.**

- 2.5 Design and Implementation Constraints
  - Compliance with GDPR for user data privacy.
  - Integration with third-party payment services like Stripe and PayPal.
- 2.6 User Documentation
  Comprehensive user manuals, FAQs, and online help will be provided.
  2.7 Assumptions and Dependencies
  - Stable internet connection for all users.
  - Compatible with modern browsers and devices.4 Operating Environment
  - Server: Windows Server or Linux with ASP.NET runtime.
  - Database: SQL Server.
  - Client: Modern web browsers (Chrome, Firefox, Edge).

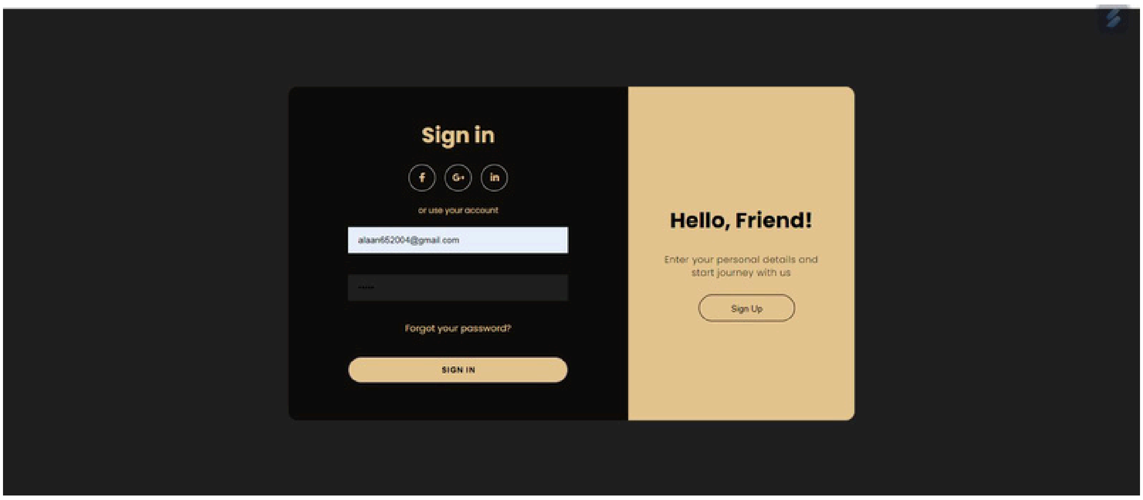
- 2.5 Design and Implementation Constraints
  - Compliance with GDPR for user data privacy.
  - Integration with third-party payment services like Stripe and PayPal.
- 2.6 User Documentation
  Comprehensive user manuals, FAQs, and online help will be provided.
- 2.7 Assumptions and Dependencies
  - Stable internet connection for all users.
  - Compatible with modern browsers and devices.

2.3 User Classes and Characteristics



Plan your event in one click

Effortlessly plan your event with us! Find venues, caterers, and more. Start planning today!

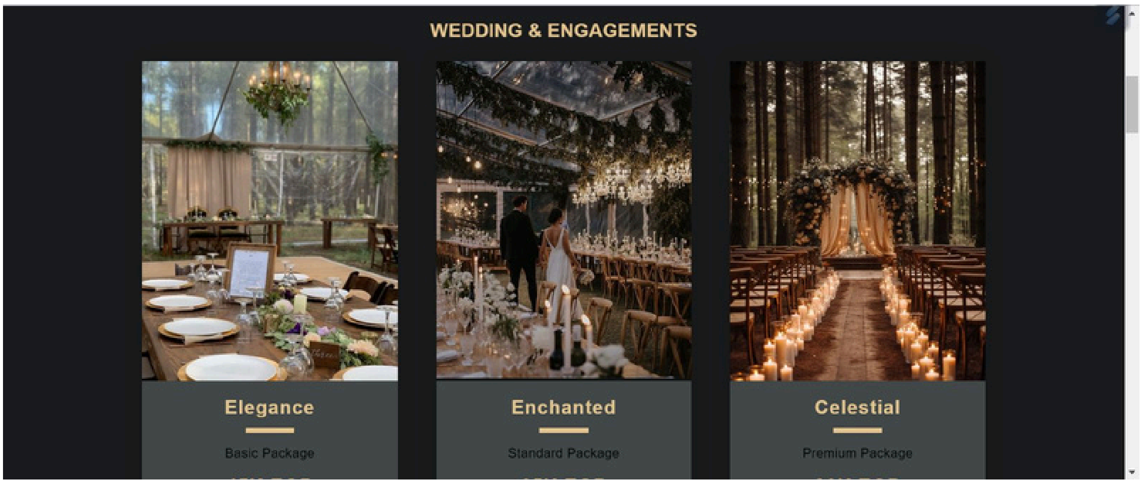


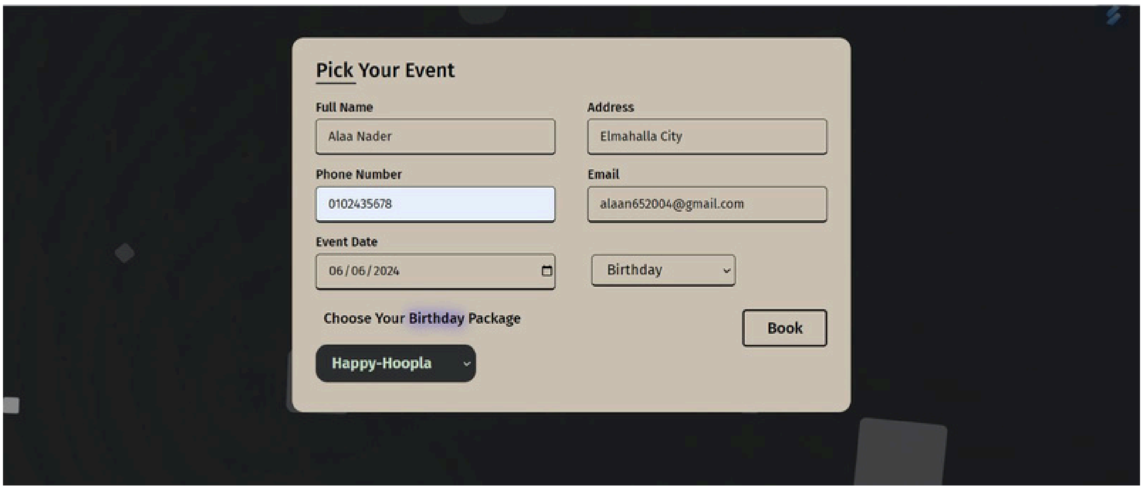
alaan652004@gmail.com

•••••

Enter your personal details and start journey with us

2.2 Product Features




Pick Your Event

Full Name: Alaa Nader

Address: Elmahalla City

Phone Number: 0102435678

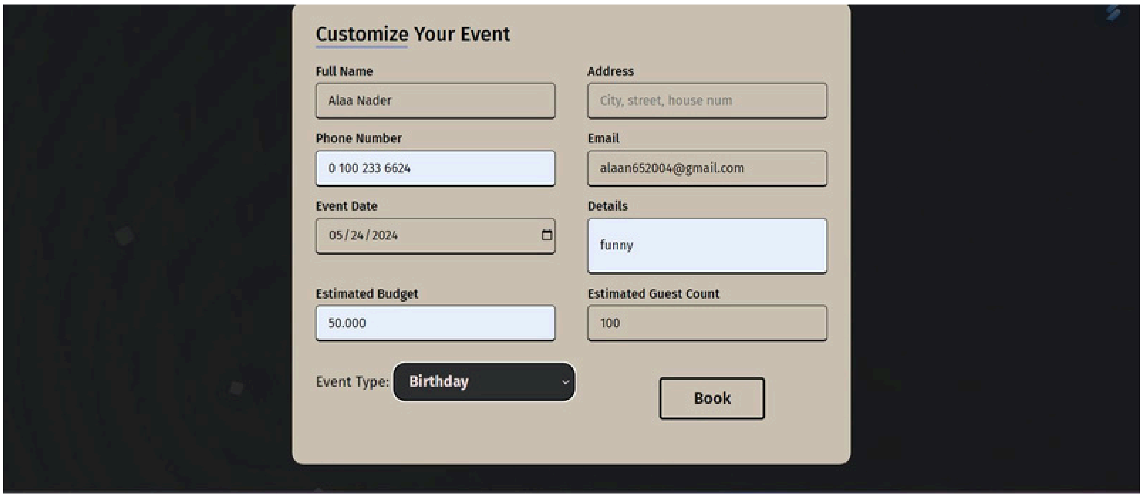
Email: alaan652004@gmail.com

Event Date: 06/06/2024

Birthday

Choose Your Birthday Package

Happy-Hoopla


Customize Your Event

Full Name: Alaa Nader

Address: City, street, house num

Phone Number: 0 100 233 6624

Email: alaan652004@gmail.com

Event Date: 05/24/2024

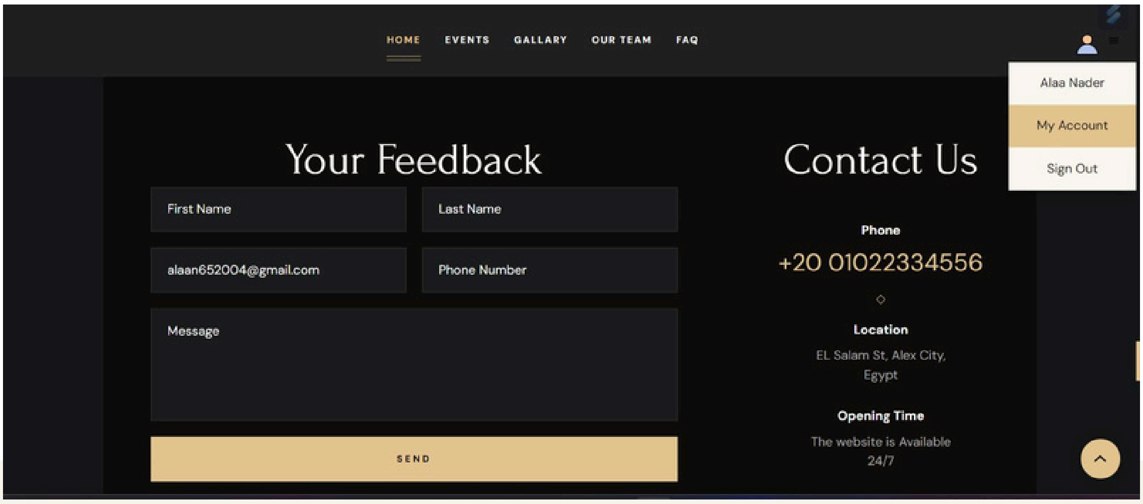
Details: funny

Estimated Budget: 50.000

Estimated Guest Count: 100

Event Type: Birthday

Design and Implementation Constraints



Your Feedback

First Name

Last Name

alaan652004@gmail.com

Phone Number

Message

Contact Us

Phone

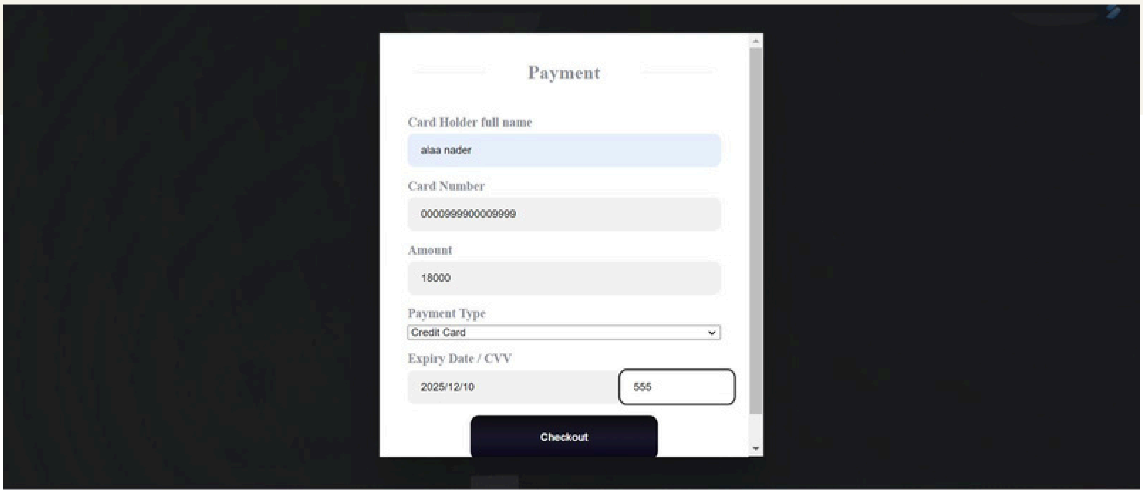
+20 01022334556

Location

EL Salam St, Alex City, Egypt

Opening Time

The website is Available 24/7



Payment

Card Holder full name: alaa nader

Card Number: 0000999900009999

Amount: 18000

Payment Type: Credit Card

Expiry Date / CVV: 2025/12/10 555

➡ Requirements Gathering

- **1.1 User Management - User Registration and Login:**

  - Users should be able to register and log in with email, password, or social media.

  - Roles and Permissions: Support different user roles, such as Organizer, Attendee, Sponsor, Volunteer, etc.

  - Profile Management: Users can edit their profiles with personal information and preferences.

- **1.2 Event Creation and Management - Event Setup:**

  - Organizers can create events with details like title, description, date, time, location, and type (in-person/online).

  - Agenda Planning: Allows adding agenda items, speakers, and session details.

  - Registration Management: Supports attendee registration, waitlists, and ticketing options.

  - Venue Management: For in-person events, manage venue details, including seating arrangements, capacity, and floor plans.

  - User can choose a pre-made package with its details like price, features, photos, decoration, and themes.

- **1.3 Ticketing and Payment**

  - Ticketing: Enables ticket creation with options for different types (e.g., General Admission, VIP, etc.), pricing, and limits.

  - Payment Integration: Integrates payment gateways (e.g., Stripe, PayPal) for secure online transactions.

  - Discounts and Promotions: Allows adding discount codes, early-bird pricing, and group discounts.

- **1.4 Communication and Notifications**

  - Email and SMS Notifications: Automated alerts for registration confirmation, reminders, and updates.

  - Push Notifications: For mobile apps, send reminders and announcements.

  - Messaging: Enables in-app or email messaging between attendees and organizers.

- **1.5 Check-in and Attendance**

  - QR Code Generation: Generate QR codes for attendees for quick check-in.

  - Attendance Tracking: Track attendance through check-in and check-out features.

  - Badge Printing: Supports printing of attendee badges with names

1.6 Reports and Exports

- Data Export: Export attendee lists, financial data, and event details in formats like CSV or PDF.

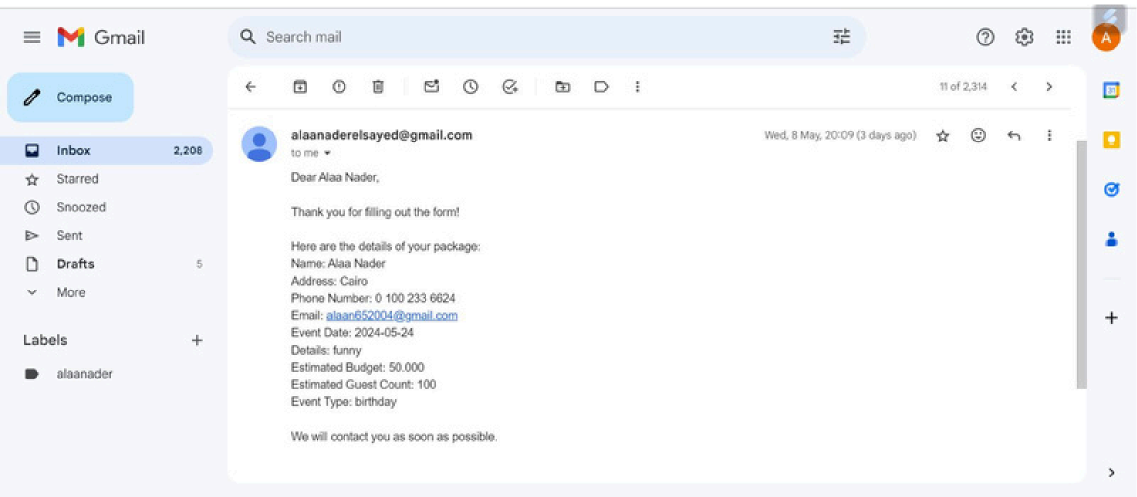
- Report Generation: Create reports on event performance, financials, and attendance.

1.7 Admin Capabilities

- Can add other admin using a form.

.- Can show all users in the system and do CRUD operations on them.

- Can show all pre-made packages and do CRUD operations on them.



alaanaderelsayed@gmail.com
to me
Wed, 8 May, 20:09 (3 days ago)

Dear Alaa Nader,

Thank you for filling out the form!

Here are the details of your package:
Name: Alaa Nader
Address: Cairo
Phone Number: 0 100 233 6624
Email: alaan652004@gmail.com
Event Date: 2024-05-24
Details: funny
Estimated Budget: 50.000
Estimated Guest Count: 100
Event Type: birthday

We will contact you as soon as possible.

. External Interface Requirements

4.1 User Interfaces

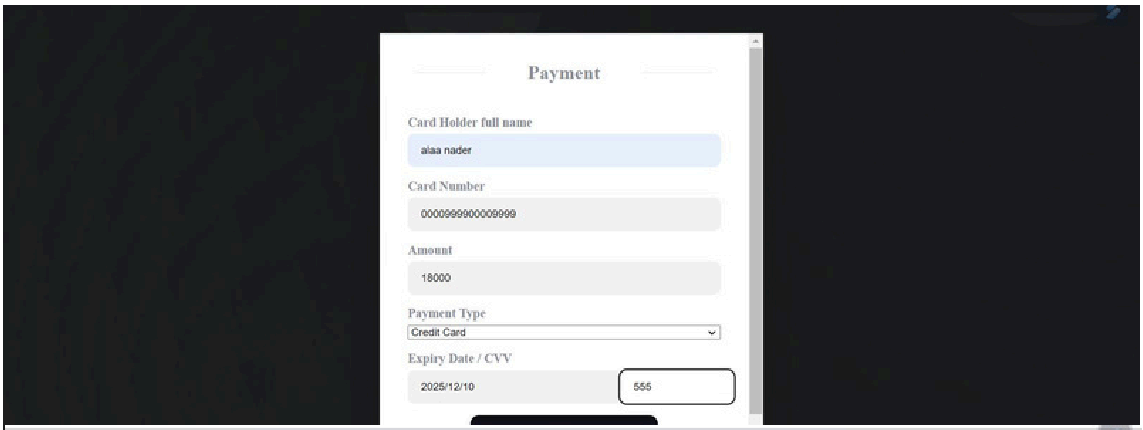
- Admin Dashboard: A web-based interface for managing events, attendees, and reports.
- Attendee Portal: A registration and ticketing interface for event participants.

4.2 Hardware Interfaces

- The EMS will be hosted on a cloud server, with no specific hardware requirements for end-users.

4.3 Software Interfaces

- Payment Gateway: Integration with third-party payment services like Stripe or PayPal.
- Email Server: Integration with SMTP or third-party email services (e.g., SendGrid) for notifications.



Payment

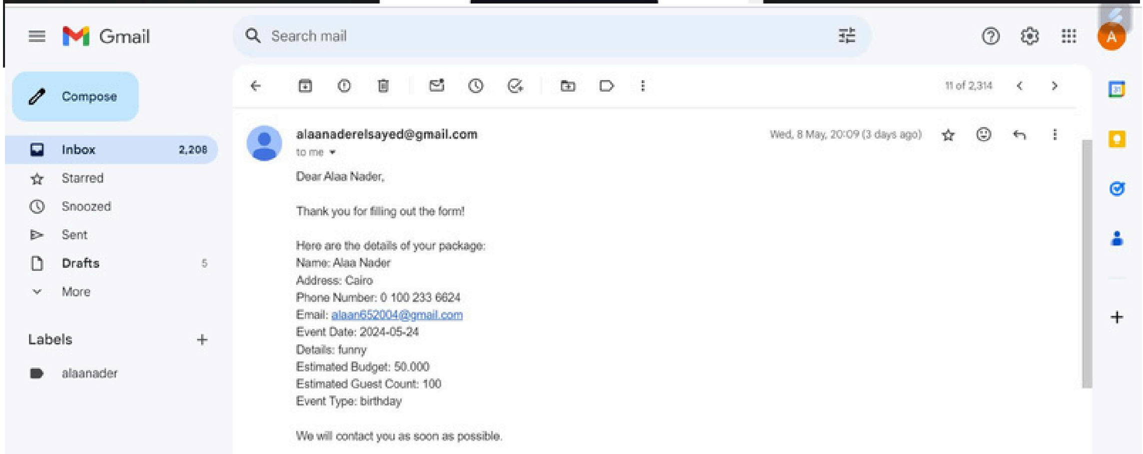
Card Holder full name: alaa nader

Card Number: 0000999900009999

Amount: 18000

Payment Type: Credit Card

Expiry Date / CVV: 2025/12/10 555



alaanaderelsayed@gmail.com
to me
Wed, 8 May, 20:09 (3 days ago)

Dear Alaa Nader,

Thank you for filling out the form!

Here are the details of your package:
Name: Alaa Nader
Address: Cairo
Phone Number: 0 100 233 6624
Email: alaan652004@gmail.com
Event Date: 2024-05-24
Details: funny
Estimated Budget: 50.000
Estimated Guest Count: 100
Event Type: birthday

We will contact you as soon as possible.

Design

- **Objective:**The objective of the design phase in the Event Management System (EMS) is to create a structured and scalable architecture that ensures seamless functionality and a user-friendly experience. The design should emphasize
  - Efficiency: Optimize the flow of data and user interactions to provide quick and reliable access to core features like event creation, registration, and analytics.
  - Scalability: Develop a system capable of handling increased loads as user demand grows, ensuring high performance even during peak usage.
  - Integration Readiness: Allow for seamless integration with external services such as payment gateways, social media platforms, and notification APIs.
  - Security: Incorporate robust security measures, such as encrypted communication and secure data storage, to protect sensitive user and transaction data.
  - Accessibility: Design interfaces that are responsive and adhere to accessibility standards, ensuring inclusivity for all users.
- **System Design and Architecture**

5.1 Architecture

- **Frontend: Built with HTML, CSS, JavaScript for dynamic interactions.**
- **Backend: Built using ASP.NET Web Forms (as per your existing development stack), with SQL Server for the database.**
- **APIs: RESTful APIs for payment gateway and email integration.**

5.2 Database Design

- **Event: Event details (ID, name, description, date, time).**
- **Attendee: Personal information (ID, name, email, ticket type).**
- **Ticket: Ticket details (ID, event ID, attendee ID, status).**

Database Management System (DBMS)

- **Relational Database**:
  - MySQL or SQLSERVER for structured data (user registrations, event details, ticket information).
  - SQLite (for smaller setups or demo sites).'
  - NoSQL Database (Optional): .MongoDB for handling unstructured data, if required for dynamic content.

3**. Backend Development Frameworks**

- Languages: C #: Popular with ASP.NET Web Forms frameworks for rapid development and ease of maintenance.
- **Frameworks:**
- 1.ASP.NET Web Forms
- Description: A classic web development framework in ASP.NET, ideal for event-driven, rapid development.
- Scalability and Maintenance: ASP.NET Web Forms allows for easy maintenance, especially when paired with SQL Server for data handling and IIS for hosting.

4. **Frontend Development Frameworks**

- **Languages:**
- C #: Popular with ASP.NET Web Forms frameworks for rapid development and ease of maintenance.
- **Frameworks:**

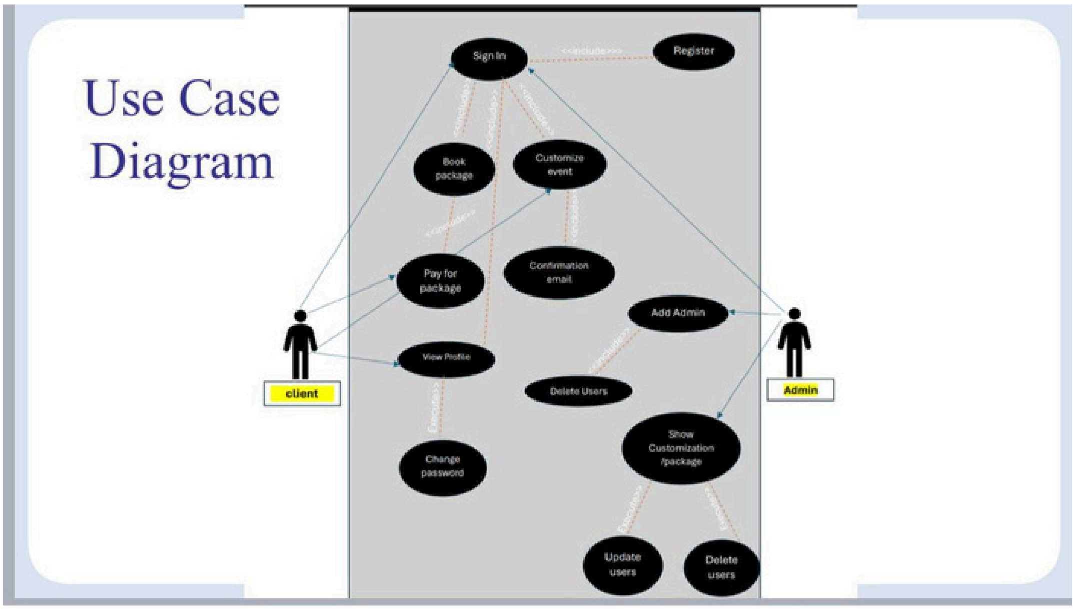
1.ASP.NET Web Forms

- Description: A classic web development framework in ASP.NET, ideal for event-driven, rapid development.

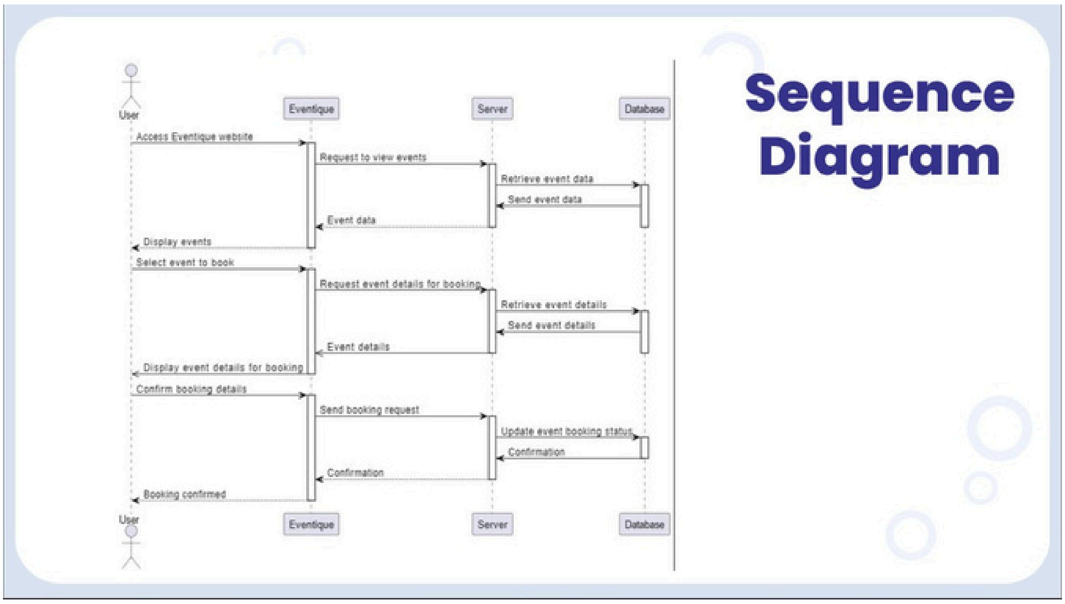
- Scalability and Maintenance: ASP.NET Web Forms allows for easy maintenance, especially when paired with SQL Server for data handling and IIS for hosting.

5.2 High-Level Architecture Diagram

A diagram showing the system components and how they interact (e.g., client-server architecture, components like the frontend, backend, and database).

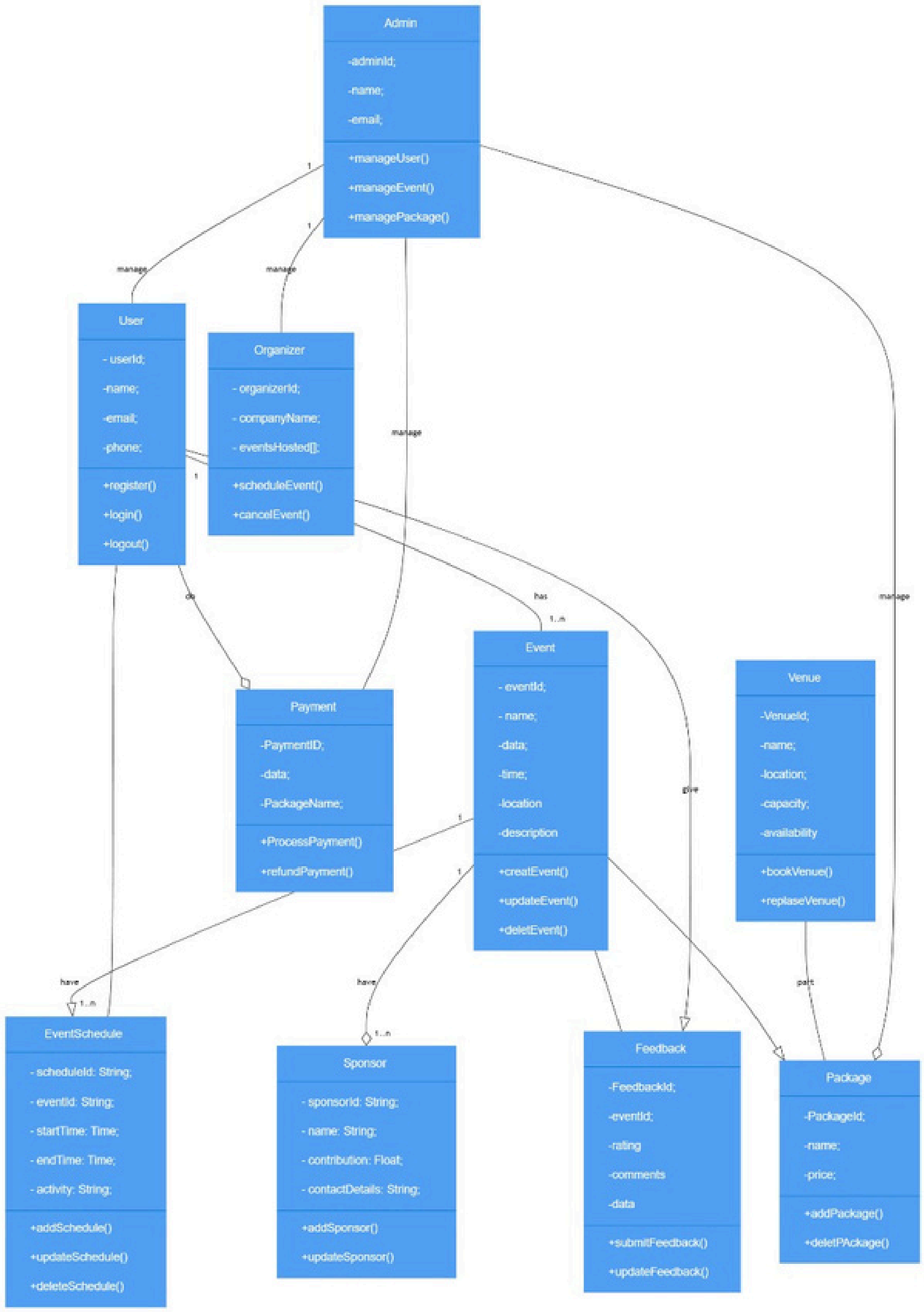


Use Case Diagram

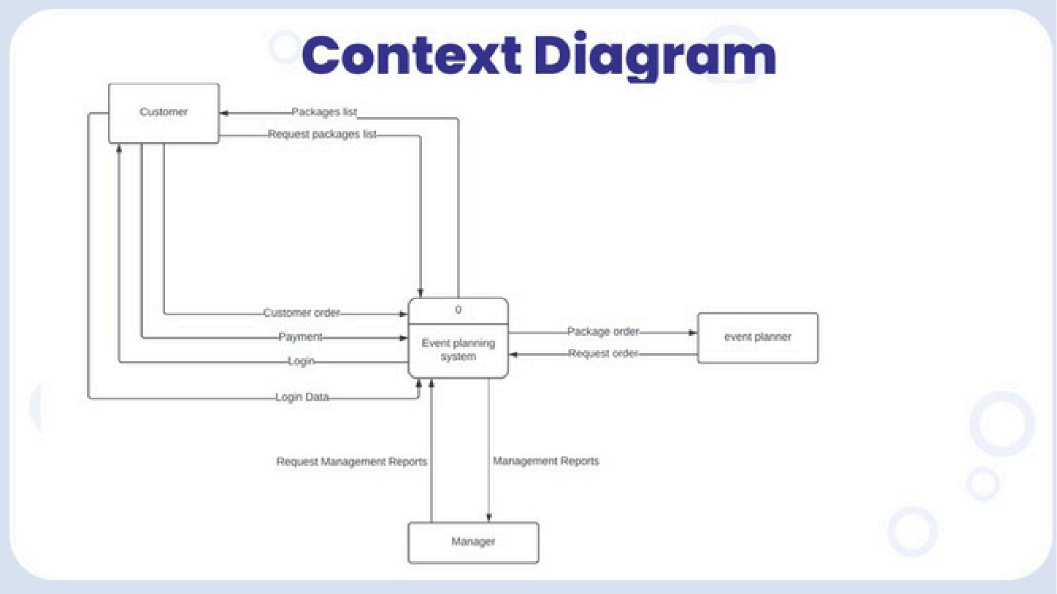


Sequence Diagram

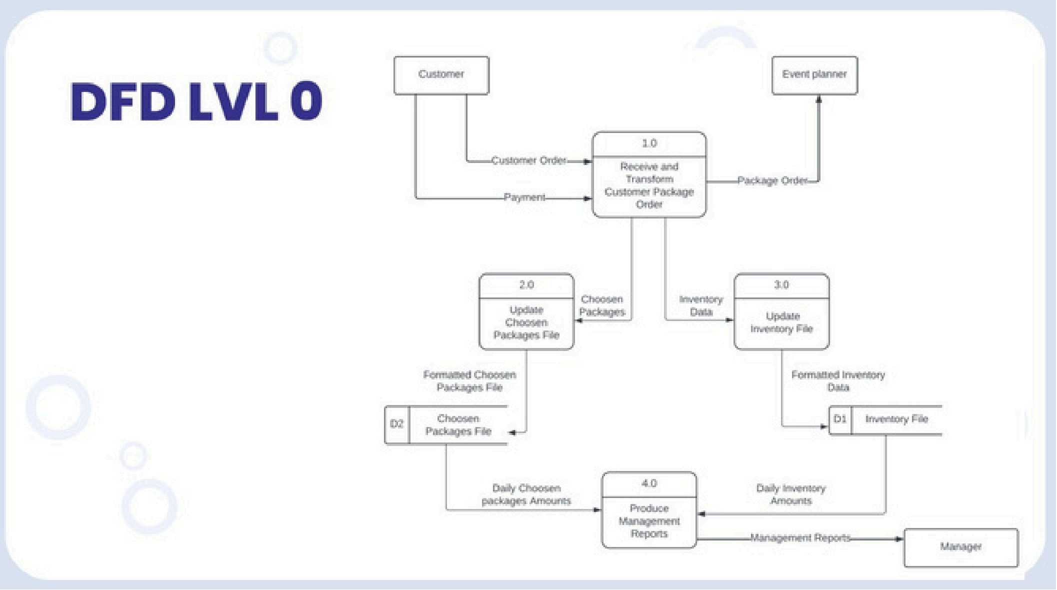
class Diagram



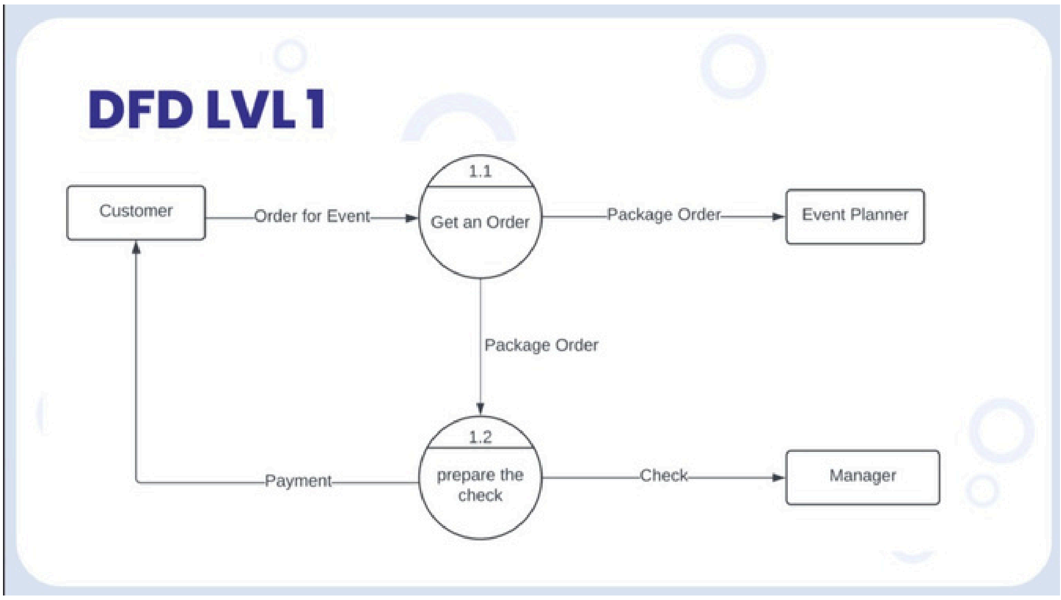
DFD Diagram



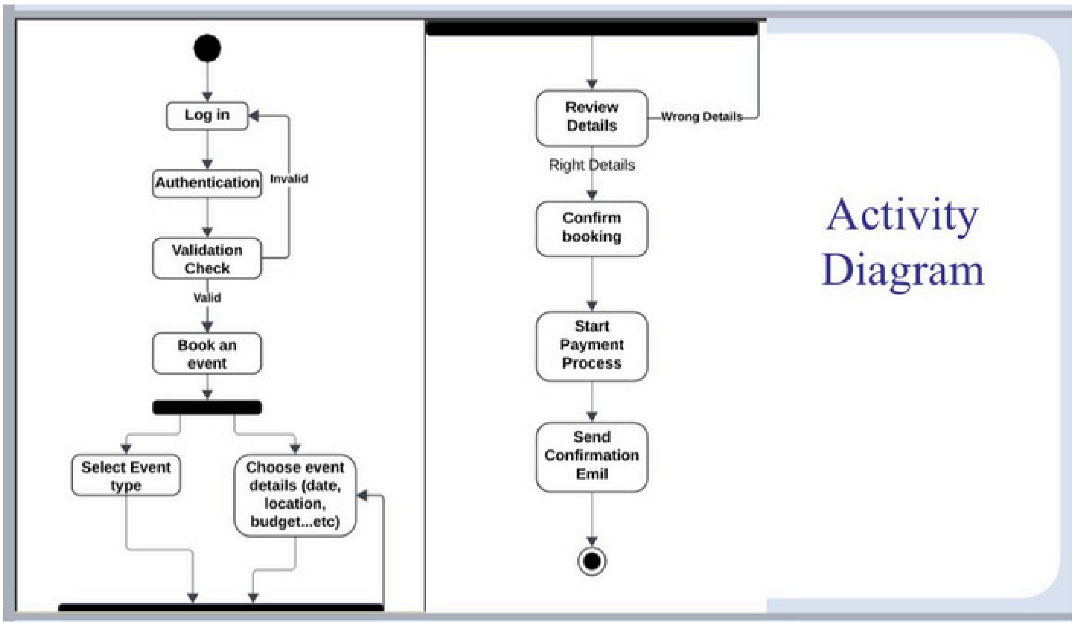
Context Diagram



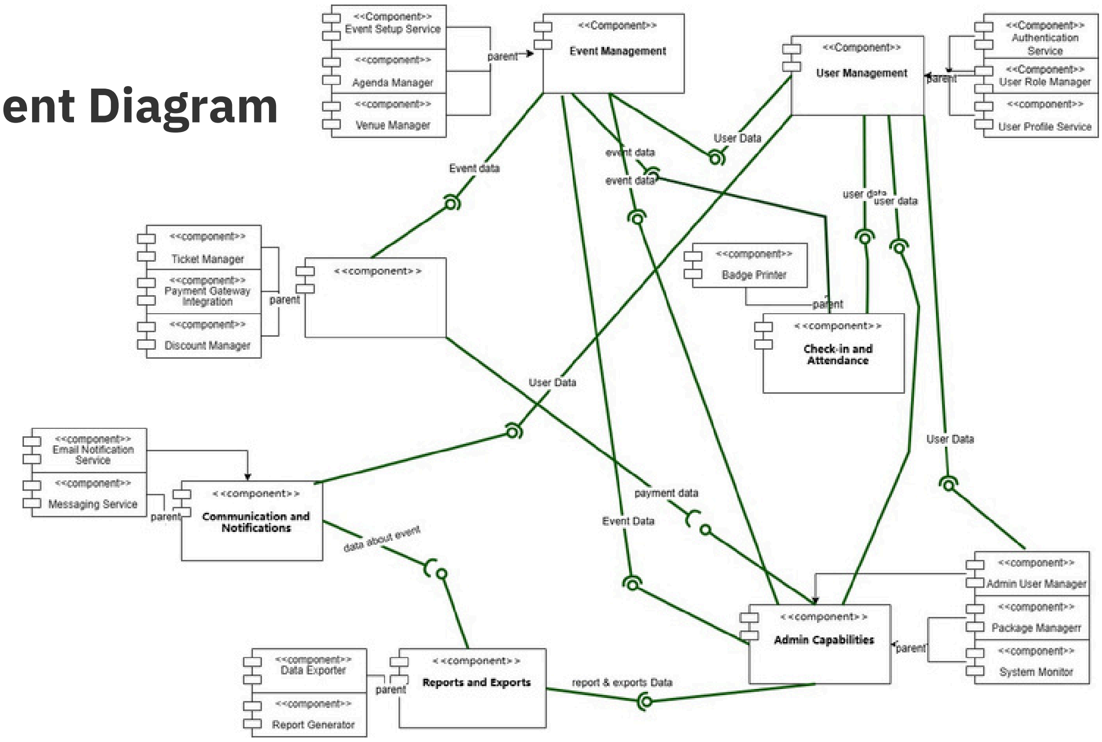
DFD LVL 0



DFD LVL 1



Component Diagram


<<Component>> Event Setup Service
<<component>> Agenda Manager
<<component>> Venue Manager
parent
<<Component>> Event Management
<<Component>> User Management
<<Component>> Authentication Service
<<Component>> User Role Manager
<<component>> User Profile Service
parent
Event data
User Data
event data
event data
user data
user data
<<component>> Ticket Manager
<<component>> Payment Gateway Integration
<<component>> Discount Manager
parent
<<component>>
<<component>> Badge Printer
parent
<<component>> Check-in and Attendance
User Data
<<component>> Email Notification Service
<<component>> Messaging Service
parent
<<component>> Communication and Notifications
User Data
payment data
Event Data
data about event
<<component>> Admin User Manager
<<component>> Package Managerr
<<component>> System Monitor
parent
<<component>> Admin Capabilities
<<component>> Data Exporter
<<component>> Report Generator
parent
<<component>> Reports and Exports
report & exports Data

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system should handle up to 10,000 concurrent users.

5.2 Safety Requirements

- Implement fail-safe mechanisms to prevent data loss.

5.3 **Security Requirements**

- Use encryption (SSL/TLS) for sensitive data.
- Multi-factor authentication for admin accounts.

5.4 **Software Quality Attributes**

- Reliability: 99.9% uptime.
- Usability: User-friendly interfaces for all user classes.

2.1 Performance

- Scalability: The system should handle large numbers of simultaneous users, especially during registration periods.
- Fast Loading Times: Pages should load within 2 seconds for optimal user experience.

2.2 Security

- Data Encryption: Secure sensitive data using encryption (e.g., HTTPS, SSL/TLS).
- Access Control: Ensure role-based access to different system features.
- Compliance: Adhere to data protection standards (e.g., GDPR).

2.3 Usability

- Responsive Design: The system should be accessible on desktops, tablets, and mobile devices. - User-Friendly Interface: Simple and intuitive UI/UX to facilitate easy navigation.
- Multilingual Support: Support for multiple languages if targeting international users.

2.4 Reliability and Availability

- 99.9% Uptime: The system should be available at all times with minimal downtime.
- Backup and Recovery: Regular data backups and a disaster recovery plan in case of failures.

2.5 Maintainability

- Modular Architecture: Adopt modular code to facilitate updates and maintenance.
- Documentation: Comprehensive documentation for developers, administrators, and end-users.

2.6 Compatibility

- Browser Compatibility: Supports popular browsers (Chrome, Firefox, Safari, Edge).
- Integration with External Tools: Integrate with CRM systems, marketing tools, and analytics platforms.

🔴 Testing

- **Objective:** Ensure the website functions smoothly, is user-friendly, and meets all technical and functional requirements.
- **Activities:** Execute unit tests to check individual components for errors. Carry out integration testing to ensure different parts of the system work together seamlessly. Conduct user acceptance testing with a select group of users to validate that the product meets their needs and expectations. Perform load testing to assess performance under high traffic conditions.

6.1 Usability Testing

Usability testing will be performed to ensure the system is user-friendly and intuitive for both attendees and admins. This will include user observation, task completion testing, and surveys.

6.2 Performance Testing

Performance testing will be conducted to verify that the system can handle 10,000 simultaneous users during peak event registration periods. Load and stress testing will be used to assess the system’s performance.

6.3 Functionality Testing

Functionality testing will verify that all core features of the system (event creation, registration, ticketing) work as expected. This includes unit and integration testing.

6.4 Compatibility Testing

Compatibility testing will ensure that the system is accessible on all modern browsers (Chrome, Firefox, Safari) and works across desktop and mobile devices.

6.5 Security Testing

Security testing will check for vulnerabilities such as SQL injection and cross-site scripting. Payment data and personal information will be encrypted, and role-based access will be enforced.

- **Deliverables:** Test reports, bug logs, and a revised version of the product addressing identified issues.

🤖 Launch and Maintenance:

- **Objective:** Successfully release the product to the market and ensure ongoing functionality and improvements.
- **Activities:** Deploy the website to a live environment, ensuring all systems are operational and secure. Monitor performance and user feedback closely to identify any issues. Regularly update the platform with improvements or new features based on user analysis and industry trends. Develop a maintenance plan to address future technical or content updates efficiently.
- **Deliverables:** Live website, maintenance schedule, and a roadmap for future enhancements.

Team Roles

1. **Project Manager:**
   - **Responsibilities:** Oversee the project from inception to completion. Coordinate between different teams, schedule meetings, and ensure that the project is progressing according to the timeline and budget. Conduct stakeholder interviews to gather requirements and feedback.
2. **Business Analyst:**
   - **Responsibilities:** Engage with stakeholders to elicit detailed requirements. Translate business needs into technical requirements and work closely with the development team to ensure these are implemented correctly.
3. **UI/UX Designer:**
   - **Responsibilities:** Design wireframes and prototypes based on user requirements. Develop a cohesive visual identity that aligns with the brand ethos. Ensure the platform is intuitive and userfriendly.
4. **Front-end Developer:**
   - **Responsibilities:** Develop the user-facing parts of the website using technologies like HTML, CSS, and JavaScript. Ensure the site is responsive and accessible on all devices.
5. **Back-end Developer:**
   - **Responsibilities:** Set up and maintain the server, databases, and application logic. Ensure data security and efficient data processing. Integrate third-party APIs and payment systems.
6. **Tester/Quality Assurance Specialist:**
   - **Responsibilities:** Execute comprehensive testing procedures to ensure the website is functional, user-friendly, and meets all specifications. Document any bugs or issues and work with developers to resolve them.
7. **Marketing and Communications Specialist:**
   - **Responsibilities:** Develop and execute a launch strategy for the website. Create promotional materials and manage communications with the target audience. Gather user feedback post-launch to guide future updates.

💀 Risk Management

- Identify potential risks
- Develop mitigation strategies
- Assign risk owners
- Server downtime
- Security vulnerabilities
- **Objective:** Proactively identify and mitigate risks to ensure project success and sustainability.
- **Activities:** Develop a risk management plan that identifies potential risks such as technological failures, data breaches, and market fluctuations. Assign a risk owner for each identified risk to ensure accountability. Implement monitoring tools to detect and log any risk occurrences. Regularly review and update the risk management plan as the project progresses.
- **Deliverables:** Risk management plan, risk logs, and mitigation strategies.

⏳ Timeline

1. **Kick-off Meeting:**

- **Date:** Week 1
- **Objective:** Define project goals, introduce team members, and outline the project plan.

2. **Completion of Requirements Gathering:**
   - **Date:** Week 3
   - **Objective:** Finalize user stories, requirement specifications, and the project scope document.
3. **Completion of Design Phase:**
   - **Date:** Week 6
   - **Objective:** Deliver finalized UI/UX designs and interactive prototypes.
4. **Completion of Development Phase:**
   - **Date:** Week 12
   - **Objective:** Have a fully functional front-end and back-end system, with integrated APIs ready for testing.
5. **Completion of Testing Phase:**
   - **Date:** Week 14
   - **Objective:** Finalize test reports, address bugs, and validate the product against requirements.
6. **Launch:**
   - **Date:** Week 16
   - **Objective:** Deploy the website to the live environment and begin post-launch marketing efforts.
7. **Post-launch Review and Maintenance Planning:**
   - **Date:** Continuous post-launch
   - **Objective:** Monitor platform performance, gather user feedback, and plan for future updates and maintenance.

Appendices

A. Glossary

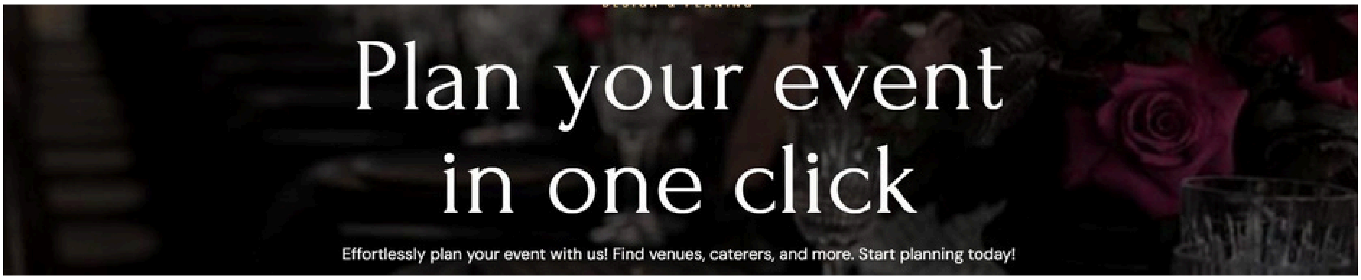
- **EMS: Event Management System**
- **GDPR: General Data Protection Regulation**

B. Analysis Models

- **To be provided during the design phase.**

C. Issues List

- **None at this stage.**


Plan your event
in one click
Effortlessly plan your event with us! Find venues, caterers, and more. Start planning today!