

Group Task – 01 Assessment Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Only for Course Teacher** | | | | | | |
|  | | **Needs Improvement** | **Developing** | **Sufficient** | **Above Average** | **Total Mark** |
| **Allocate mark & Percentage** | | **25%** | **50%** | **75%** | **100%** | **10** |
| **Functionality** | **04** |  |  |  |  |  |
| **Code Quality** | **03** |  |  |  |  |  |
| **Collaboration & Presentation** | **03** |  |  |  |  |  |
| **Total obtained mark** | | | | | |  |
| **Comments** |  | | | | | |

**Semester: Summer 2025 Group No:- 01**

**Name: Mahim Ahmad ID: 242-35-705**

**Name: Tamim Munshi ID: 242-35-858**

**Name: Sahed Alam ID: 242-35-590**

**Batch:** 43 **Section:** J1 **Course Code**: SE 133

**Course Name:** Software Development Capstone Project

**Course Teacher Name:** Md. Sakib Ali Mazumder **Designation:** Lecturer

**Submission Date: 04/07/2025**

****

**Group Name:** **Eventronix**

**ABSTRACT**

The Event Management System (EMS) is a structured and user-centric software solution developed in the C programming language, designed to automate and simplify the end-to-end process of organizing, managing, and participating in events. By integrating core modules such as User Management, Event Scheduling, Venue Booking, Registration, Simulated Payment Processing, Service Requests, and Equipment Rentals, the system delivers a seamless experience for all types of users — from administrators and organizers to attendees and service providers.

At its foundation, the EMS offers a basic Registration and Login system with file-based credential checking that supports role-based access control, ensuring that each user — whether an administrator, organizer, or attendee — is granted appropriate privileges and interfaces tailored to their responsibilities. This role separation enhances operational clarity and system security.

Event organizers benefit from a comprehensive Event Management interface, allowing them to create, update, search, and delete events with attributes like schedules, venues, ticket pricing, and status. The system also supports simulated real-time Venue Management, enabling the booking of event locations based on availability, capacity, and facilities.

For attendees, EMS simplifies the participation process through easy registration, ticket issuance via text-based confirmation codes, simulated payment tracking, and a dedicated section to view and manage their registered events. It also includes a complaint submission feature, enabling feedback collection for continuous improvement.

To support logistics, the system integrates a Service Management module, allowing event organizers to request additional services such as catering, decoration, or photography, as well as an Equipment Rental system for procuring necessary items like projectors or sound systems.

Behind the scenes, EMS leverages efficient data structures — including arrays and structs — alongside robust file I/O operations for data persistence (.txt/.csv files). It includes well-structured input validation and error handling, ensuring data integrity and system reliability during execution.

Overall, the Event Management System represents a complete, scalable, and efficient software solution that transforms traditional event planning into a digital experience. It not only enhances management productivity but also enriches user satisfaction by offering transparency, accessibility, and simulated real-time engagement across all stages of event execution.

Contents

[**Chapter -- 1** 4](#_Toc202564605)

[INTRODUCTION 4](#_Toc202564606)

[1.1 About the System 4](#_Toc202564607)

[1.2 Problem Statement 4](#_Toc202564608)

[1.3 Purpose of the System 4](#_Toc202564609)

[1.4 Objectives of the System 5](#_Toc202564610)

[1.5 Why an Event Management System Is Necessary 5](#_Toc202564611)

[**Chapter -- 2** 6](#_Toc202564612)

[Literature Review 6](#_Toc202564613)

[2.1 Related Work 6](#_Toc202564614)

[2.2 Gap Analysis 7](#_Toc202564615)

[**Chapter -- 3** 7](#_Toc202564616)

[Methodology 7](#_Toc202564617)

[3.1 Approach 7](#_Toc202564618)

[3.2 Tools and Techniques: 8](#_Toc202564619)

[3.3 Project Phases: 8](#_Toc202564620)

[3.4 Error Handling 8](#_Toc202564621)

[**Chapter -- 4** 9](#_Toc202564622)

[4.1 Stakeholder: 9](#_Toc202564623)

[4.2 User Profiling 9](#_Toc202564624)

[4.3 Elicitation Techniques 11](#_Toc202564625)

[4.4 Requirement Prioritization 11](#_Toc202564626)

[4.5 Functional Requirement 12](#_Toc202564627)

[4.6 Non-Functional Requirement 16](#_Toc202564628)

[**Chapter -- 5** 17](#_Toc202564629)

[5.1 Use Case Diagram 17](#_Toc202564630)

[5.2 Use Case Description 21](#_Toc202564631)

[5.3 Activity Diagram 50](#_Toc202564632)

[5.4 Sequence Diagram 58](#_Toc202564633)

[**Chapter -- 6** 59](#_Toc202564634)

[6.1 Requirements Traceability Matrix – RTM 59](#_Toc202564635)

[**Chapter – 7** 61](#_Toc202564636)

[7.1 Prototyping: 61](#_Toc202564637)

[**Chapter -- 8** 62](#_Toc202564638)

[8.1 Gantt Chart: 62](#_Toc202564639)

[**Chapter – 9** 63](#_Toc202564640)

[9.1 Conclusion 63](#_Toc202564641)

# **Chapter -- 1**

## ****INTRODUCTION****

### **1.1 About the System**

The **Event Management System (EMS)** is a C-based software application designed to streamline and automate all processes related to planning, organizing, and managing events. The system serves as a centralized platform where users including administrators, event organizers, attendees, and vendors can interact efficiently. EMS supports a wide range of functionalities such as user registration and authentication, event creation and scheduling, venue booking, attendee registration, service request handling, and equipment rental management. Built with robust data structures and file-handling techniques, EMS offers a lightweight yet powerful solution for managing both small-scale and large-scale events

### **1.2 Problem Statement**

In many organizations and academic institutions, event management is still handled manually or through disconnected digital tools such as spreadsheets and email chains. This results in various problems, including:

* Inefficient communication between organizers, attendees, and vendors.
* Increased chances of double bookings, missed deadlines, and human errors.
* Lack of centralized data, making tracking and reporting difficult.
* Difficulty in handling last-minute changes and service requests.

These inefficiencies ultimately reduce the quality of the event experience for all stakeholders and increase operational overhead for organizers.

### **1.3 Purpose of the System**

The primary purpose of the Event Management System is to:

* Eliminate manual, paper-based event coordination.
* Provide a digital platform where different stakeholders can interact with event-related operations seamlessly.
* Ensure that event logistics, scheduling, registration, and communication are handled efficiently through automation.
* Offer a modular and scalable system that improves accuracy, reduces workload, and enhances user experience.

### **1.4 Objectives of the System**

The main objectives of developing the Event Management System are:

1. To implement input validation and basic error handling to ensure the system behaves reliably under incorrect input or failed file access.
2. **To develop a user-friendly platform** that allows users to log in based on roles and access customized features.
3. **To automate event operations** such as creation, update, registration, and scheduling with data persistence.
4. **To implement secure and efficient file handling**, ensuring all data such as users, events, bookings, and complaints are stored reliably.
5. **To provide modules** for venue management, service provider integration, and equipment rental in one unified interface.
6. **To ensure transparency and accountability** through reporting tools like attendance tracking and revenue summaries.
7. **To reduce administrative burden** by centralizing all tasks in a single, accessible application.

### **1.5** **Why an Event Management System Is Necessary**

**An Event Management System (EMS) is essential for organizing, streamlining, and scaling the process of booking, managing, and delivering events. Here's why your system is necessary:**

**1. Centralized Booking and Management**

**Without a system, handling event bookings is manual, error-prone, and time-consuming. Your system provides:**

* **A single platform for customers to book events like weddings, birthdays, seminars, etc.**
* **A centralized dashboard for both users and admins to manage everything easily.**

**2. Automation Saves Time and Effort**

* **Automates tasks like payment tracking, availability checking, discount application, and confirmation messaging.**
* **Reduces manual errors and repetitive administrative work.**

**3. Real-Time Availability and Scheduling**

* **Shows users up-to-date availability of venues, services, and equipment.**
* **Prevents double-bookings and ensures smooth coordination across event types.**

**4. Improved Customer Experience**

* **Users can browse, select, and book events online 24/7.**
* **Clear workflows, confirmation messages, and dashboards give users confidence and control.**

**5. Increased Transparency and Accountability**

* **Users accept terms and conditions before confirming.**
* **Booking data, payment records, and feedback are all stored — creating traceability and reducing disputes.**

**6. Revenue Growth Through Better Management**

* **Discounts and offers can be easily managed and promoted.**
* **Admin can monitor which event types perform best and adjust services accordingly.**

**7. Data-Driven Decisions**

* **Feedback history, booking trends, and payment status reports help improve service quality and business strategy.**

**8. Supports Scaling**

**As your business grows, managing 10–100 bookings per week manually becomes impossible. Your system:**

* **Allows unlimited users, event types, and automation.**
* **Supports future expansion (e.g., mobile app, online chat support, analytics, etc.).**

# **Chapter -- 2**

## **Literature Review**

### **2.1 Related Work**

Several event management systems have been developed to handle tasks like registration, ticketing, and scheduling. Platforms such as Eventbrite and Whova offer comprehensive features but are mostly web-based and require internet access and databases. Student-level projects in languages like Java or Python focus on basic operations such as login and event listing, but they often lack full integration and offline capability.

### **2.2 Gap Analysis**

Existing systems usually address only part of the event management process. There is a lack of a simple, file-based system written in C that integrates all core features — including role-based access, event and venue management, service handling, and equipment rental. This project fills that gap by offering a complete and lightweight solution suitable for offline use and academic learning.

# **Chapter -- 3**

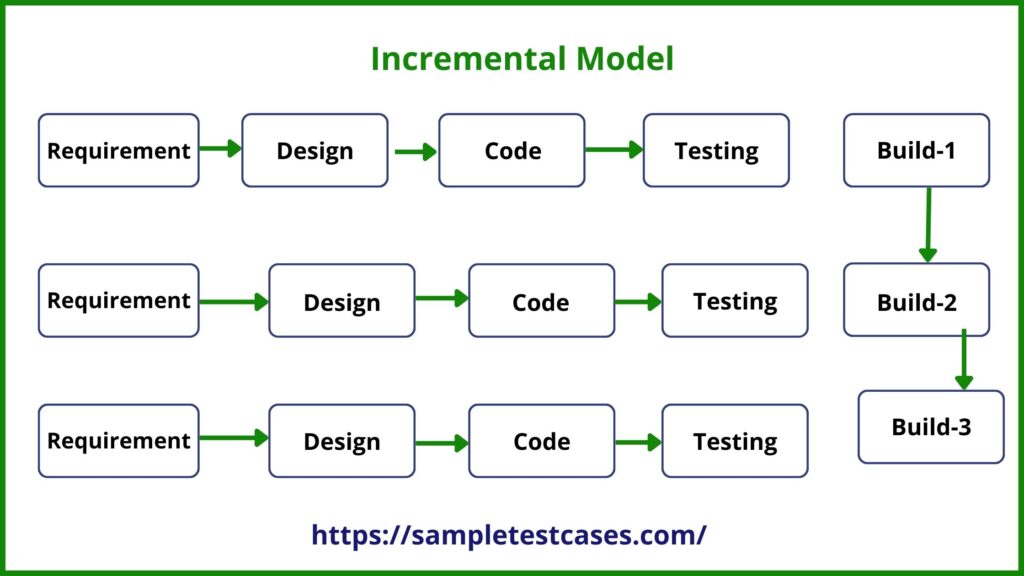
## ****Methodology****

### **3.1 Approach**

The development of the Event Management System follows a **modular and incremental approach** using the **Procedural Programming paradigm** in the C language. Each module (e.g., User Management, Event Handling, Registration, etc.) was developed independently and then integrated to ensure flexibility and maintainability.

The project uses a **bottom-up design strategy**, where lower-level functions such as file I/O operations and data validation were implemented first, followed by higher-level features like service integration and reporting. This ensured early testing and debugging of core operations before expanding to complex features

Error Handling: During the development, basic error handling techniques were used, such as input validation for empty or invalid fields, and file I/O error checks. The system provides user-friendly messages instead of crashing when an error occurs.



### **3.2 Tools and Techniques:**

|  |  |  |
| --- | --- | --- |
|  | Category | Tools / Techniques Used |
| **1.** | **Programming Language** | C (ANSI C – GCC Compiler) |
| **2.** | **Development Environment** | Code::Blocks / Visual Studio Code |
| **3.** | **Data Handling** | File I/O (.txt, .csv) for persistent storage |
| **4.** | **Data Structures** | Arrays, struct, file-based linked data |

### **3.3 Project Phases:**

|  |  |  |
| --- | --- | --- |
|  | Phase | Description |
| **1.** | **Requirement Analysis** | Identified core system features and stakeholder roles. |
| **2.** | **Design** | Structured the system into modules and defined data structures (struct). |
| **3.** | **Implementation** | Developed modules in C: login, event creation, registration, venue management, etc. |
| **4.** | **Testing and debugging** | Conducted functional testing for each module using sample data files |
| **5.** | **Integration** | Connected all modules and ensured data flow between components (e.g., user to event). |
| **6.** | **Finalization** | Added features like reporting, file persistence, and error messages. |

### 3.4 Error Handling

The Event Management System (EMS) uses simple and effective error handling to prevent crashes and guide users properly. Key strategies include:

* **Input Checks**: All user inputs (name, email, date, etc.) are checked. If invalid, the system shows an error message and asks for correction.
* **File Handling Errors**: If required .txt or .csv files are missing or can't be opened, a message is shown — "File not found or cannot be opened."
* **Booking Conflicts**: If a venue or time slot is already booked, the system informs the user and suggests other available options.
* **Payment Failures** : If payment input is wrong or incomplete, the system cancels the process and shows — "Payment failed. Please try again."
* **Unexpected Errors**: For unknown issues, the system prevents crashing and shows — "An unexpected error occurred. Returning to main menu."

These simple techniques help the EMS run smoothly and provide a better user experience.

# **Chapter -- 4**

### 4.1 Stakeholder:

* **Manager**
* **User/customer**
* **Admin**
* **Developer**

### 4.2 User Profiling

**User Profile (Manager)**

|  |  |  |
| --- | --- | --- |
| **User Class** | **Notes on Characteristic** | **Requirement Implied** |
| Type of User | Event Manager | Interface |
| Age Range | 25-55 |  |
| Frequency of Use | Daily | Performance, Maintainability |
| Mandatory | Yes |  |
| Computer Experience | Intermediate to Advanced | Documentation |
| Education | Graduate or Equivalent |  |
| Goals | Oversee event planning, booking, and coordination | Performance, Quality, Safety |
| Language Skills | Bangla, English |  |
| Number of Users | 5-10 (organization-based) |  |
| Training | Must provide | Interface, Documentation |
| Other Systems Used | Sometimes CRM or email platforms | Integration |
| Ways of Working | Works in office or on-site at events | portability |

**User Profile (Customer/User)**

|  |  |  |
| --- | --- | --- |
| **User Class** | **Notes on Characteristic** | **Requirement Implied** |
| Type of User | General Customer | Interface, Usability |
| Age Range | 18-60 |  |
| Frequency of Use | Occasionally (event-based) | Responsiveness, Simplicity |
| Mandatory | Yes (to book events) |  |
| Computer Experience | Basic to Intermediate | User-friendly Interface |
| Education | Any |  |
| Goals | Book/manage events, make payments, submit feedback | Accessibility, Usability |
| Language Skills | Bangla, English |  |
| Number of Users | Many (100s–1000s) | Scalability |
| Training | Not needed |  |
| Other Systems Used | No |  |
| Ways of Working | Use from home/mobile/office |  |

**User Profile (Admin)**

|  |  |  |
| --- | --- | --- |
| **User Class** | **Notes on Characteristic** | **Requirement Implied** |
| Type of User | System Administrator | Access Control, Interface |
| Age Range | 25-50 |  |
| Frequency of Use | Daily | Security, Auditability |
| Mandatory | Yes |  |
| Computer Experience | Advanced | Access Control, Interface |
| Education | Graduate in IT or related |  |
| Goals | Manage system settings, users, events, and content | Security, Maintainability |
| Language Skills | English (main), Bangla (preferred) |  |
| Number of Users | 1-2 per system instance |  |
| Training | Internal system training required | Interface, Configuration Documentation |
| Other Systems Used | Server/Database management tools | Integration, Maintenance |
| Ways of Working | Office or remotely via secured access | Secure Access, Portability |

### 4.3 Elicitation Techniques

* **Focus Group**

To use the focus group technique, we bring together a selected group of stakeholders such as users, event organizers, customer service agents, and admin staff. A facilitator guides the session by asking open-ended questions related to event registration, booking processes, payment methods, feedback collection, and more. Participants discuss their experiences, expectations, and frustrations with current systems or manual processes.

This method helps gather collective insights, uncover pain points, and prioritize features like venue selection, event categories, or approval workflows.

* **Documentation Analysis**

With documentation analysis, we systematically review existing materials such as event booking forms, feedback reports, policy documents, and operational checklists. The goal is to extract business rules, standard procedures, and hidden requirements already embedded in current workflows.

This approach ensures that the system aligns with established practices and uncovers gaps or inefficiencies that automation can improve.

### 4.4 Requirement Prioritization

* **Technique Used: MoSCoW Method**

The MoSCoW technique is used to categorize system requirements based on their importance and urgency. It helps developers and stakeholders agree on what features are essential for the initial version and which can be delayed or optional.

* **Must have**: Core functionality
* **Should have**: Key improvements
* **Could have**: Nice-to-have features
* **Won’t have**: Low-priority items

|  |  |  |  |
| --- | --- | --- | --- |
| **Must have** | **Should have** | **Could have** | **Won’t have** |
| Registration | Contact Us | Update Event Items | Feedback |
| Login | Customer Dashboard | Service Request | Feedback History |
| Registration Confirm message | View Offer | Equipment Rental | Event Roles and Regulation |
| Login Confirm message | Event Confirm message | Manage Discount and Offer |  |
| Admin Dashboard | Select Vanu | Password Change |  |
| Event selection | Find event Details |  |  |
| Event Information | View All Details |  |  |
| Payment | Delete Details |  |  |
| Get Payment Message |  |  |  |
| Event Approval |  |  |  |
| Payment processing |  |  |  |
| Payment method |  |  |  |
| Logout |  |  |  |

### 4.5 Functional Requirement

|  |  |
| --- | --- |
| **FR-01** | **Registration** |
| Description | user must have to registrar himself first. |
| Stakeholder | Customer, Manager, Admin. |

|  |  |
| --- | --- |
| **FR-02** | **Registration Confirmation message** |
| Description | After successful registration, the user sees a confirmation message. |
| Stakeholder | Customer, Manager,Admin. |

|  |  |
| --- | --- |
| **FR-03** | **Log-In** |
| Description | Registered users must log in before accessing the main system. Only authorized users are allowed access. |
| Stakeholder | Customer, Admin, Manager. |

|  |  |
| --- | --- |
| **FR-04** | **Login Confirmation message** |
| Description | After successful login, the user sees a confirmation message. |
| Stakeholder | Customer, Admin, Manager. |

|  |  |
| --- | --- |
| **FR-05** | **Password Change** |
| Description | Users can update or change their passwords through the dashboard. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-06** | **Customer Dashboard** |
| Description | Upon login, customers can access their personal dashboard to manage their events, view bookings, and request services. |
| Stakeholder | Customer. |

|  |  |
| --- | --- |
| **FR-07** | **Event Selection** |
| Description | Customers can select an event type from the available categories before proceeding to the next steps. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-08** | **Marriage Event Booking** |
| Description | Customers can book a marriage event and select related services. |
| Stakeholder | Customer. |

|  |  |
| --- | --- |
| **FR-09** | **Cultural Event Booking** |
| Description | Customers can book cultural events and select related services. |
| Stakeholder | Customer. |

|  |  |
| --- | --- |
| **FR-10** | **Institution Event Booking** |
| Description | Customers can book institutional events like convocation or annual programs. |
| Stakeholder | Customer. |

|  |  |
| --- | --- |
| **FR-11** | **Birthday Event Booking** |
| Description | Customers can book birthday events and manage related items. |
| Stakeholder | Customer. |

|  |  |
| --- | --- |
| **FR-12** | **Seminar Event Booking** |
| Description | Customers can arrange and book seminars. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-13** | **View Offers** |
| Description | Customers can view ongoing offers and discounts on events. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-14** | **Event Confirmation message** |
| Description | Once an event is booked, a confirmation SMS is sent to the customer. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-15** | **View Event Information** |
| Description | Detailed information about each event type is shown upon selection. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-16** | **Select Venue** |
| Description | Users can choose the venue from a list of available locations for their event. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-17** | **Payment Processing** |
| Description | Customers must complete payment to confirm their event booking. |
| Stakeholder | Customer. |

|  |  |
| --- | --- |
| **FR-18** | **Payment method** |
| Description | Customers must complete payment different methods. |
| Stakeholder | Customer. |

|  |  |
| --- | --- |
| **FR-19** | **Payment Confirmation Message** |
| Description | A message is sent to confirm the payment has been received. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-20** | **Service Request** |
| Description | Users can request additional services such as decoration, food, lighting, etc. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-21** | **Equipment Rental** |
| Description | Customers can rent equipment like microphones, speakers, chairs, etc. as part of their event. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-22** | **Submit Feedback** |
| Description | Customers can submit feedback after the event is completed. |
| Stakeholder | Customer |

|  |  |
| --- | --- |
| **FR-23** | **Admin Dashboard** |
| Description | Admins have access to a dashboard to manage all users, events, and services. |
| Stakeholder | Admin |

|  |  |
| --- | --- |
| **FR-24** | **Find Event Details** |
| Description | Admins can search and view detailed information about any event. |
| Stakeholder | Admin, Manager |

|  |  |
| --- | --- |
| **FR-25** | **Delete Event Details** |
| Description | Admin can delete unnecessary or incorrect event records. |
| Stakeholder | Admin |

|  |  |
| --- | --- |
| **FR-26** | **View All Details** |
| Description | Admin can see all user registrations, events, and transactions in one place. |
| Stakeholder | Manager, Admin |

|  |  |
| --- | --- |
| **FR-27** | **Event Approval** |
| Description | Admin must approve event requests before they are finalized. |
| Stakeholder | Admin, Manager. |

|  |  |
| --- | --- |
| **FR-28** | **Manage Discounts and Offers** |
| Description | Admin can create, edit, or remove discount offers. |
| Stakeholder | Admin, Manager. |

|  |  |
| --- | --- |
| **FR-29** | **Update Event Items** |
| Description | Admin can update services and items associated with each event. |
| Stakeholder | Manager, Admin. |

|  |  |
| --- | --- |
| **FR-30** | **View Feedback History** |
| Description | All customer feedback is saved and viewable by the admin. |
| Stakeholder | Admin, Manager. |

|  |  |
| --- | --- |
| **FR-31** | **Contact Us** |
| Description | A dashboard where users can see manager and support team phone number and email and etc. |
| Stakeholder | Customer. |

|  |  |
| --- | --- |
| **FR-32** | **Rules and regulation** |
| Description | Users can see event rules and conditions. |
| Stakeholder | Customer. |

|  |  |
| --- | --- |
| **FR-33** | **Logout** |
| Description | Users can securely log out from the system. |
| Stakeholder | Customer, Admin , Manager |

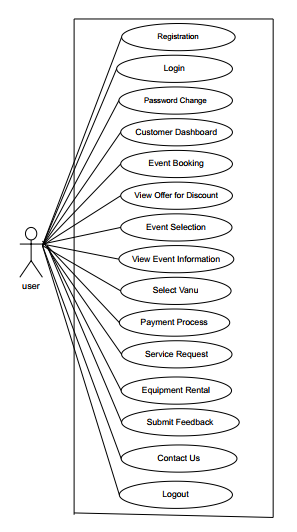
### 4.6 Non-Functional Requirement

* **Performance**:  
  The system should load dashboards and pages within 3 seconds and complete payments within 10 seconds.
* **Reliability**:  
  The system must operate with at least 99% uptime and handle errors smoothly It also includes input validation and error handling to prevent unexpected failures.
* **Security**:  
  User data must be protected with login authentication and role-based access.
* **Usability**:  
  The interface should be simple and user-friendly for all types of users.
* **Availability**:   
  The system should be accessible 24/7 for booking and event management.
* **Maintainability**:  
  Code and modules must be well-organized and easy to update.
* **Scalability**:  
  The system uses role-based access control and login authentication to protect data. User actions are restricted by roles, and sensitive files are accessed only through the system.
* **Compatibility**

Should work on standard C compilers with .txt/.csv files.

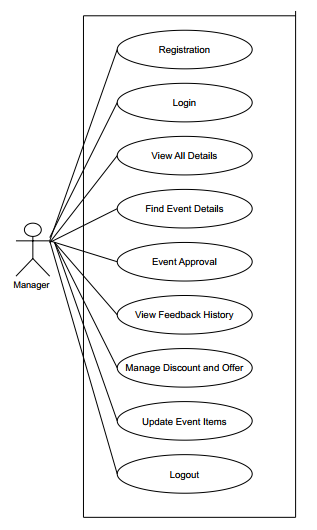
# **Chapter -- 5**

### 5.1 Use Case Diagram

**Use cases for user:**

* Registration
* Login
* Password Change
* Customer Dashboard
* Marriage Event Booking
* Cultural Event Booking
* Institution Event Booking
* Birthday Event Booking
* Seminar Event Booking
* View Offer for Discount
* Event Selection
* View Event Information
* Select Vanu
* Payment Process
* Payment Metho
* Payment Confirmation Message
* Service Request
* Equipment Rental
* Submit Feedback
* Contact Us
* Logout

Figure-: Use Case Diagram Event Management System for user

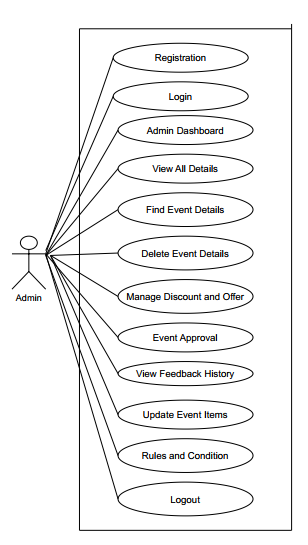
****

**Use cases for Manager:**

* Registration
* Login
* View All Details
* Find Event Details
* Event Approval
* View Feedback History
* Manage Discount and Offer
* Update Event Items
* Logout

Figure-: Use Case Diagram Event Management System for Manager

**Use cases for Admin:**

****

* Admin Dashboard
* View All Details
* Find Event Details
* Delete Event Details
* Manage Discount and Offer
* Event Approval
* View Feedback History
* Update Event Items
* Rules and Condition
* Logout

Figure-: Use Case Diagram Event Management System for Admin

**Diagrams (Use Case for All)**

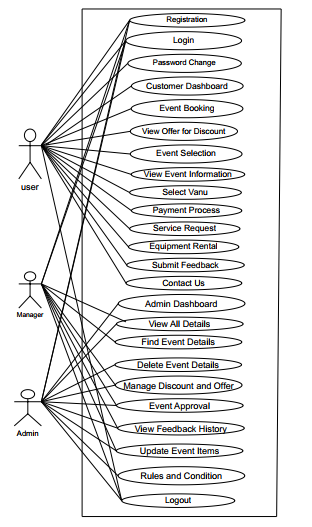
****

Figure-: Use Case Diagram Event Management System

### 5.2 Use Case Description

**Use case: Registration**

|  |  |
| --- | --- |
| **Use Case** | **Registration** |
| **Goal** | **A user register for a new account by providing required personal information.** |
| **Precondition** | **User is not already registered.** |
| **Success Condition** | **Message User is successfully registered and can log in.** |
| **Failed Condition** | **Message Registration fails due to missing or invalid information.** |
| **Primary Actors**  **Secondary Actors** | **Customer (user), Admin,** **Manager**  **System database** |
| **Trigger** | **User select on the “Register” option.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User select the registration page.** | | **2** | **User enters name, email, password, and other required details.** | | **3** | **System validates input.** | | **4** | **System stores the user data.** | | **5** | **System confirms successful registration.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Missing or invalid input** | |  | **3.a.i** | **System displays error and asks user to re-enter details.** | | **4** | **a** | **User already registered** | |  | **4.a.i** | **System prevents registration and suggests login.** | |
| **Quality Requirement** | **The user should complete this process within 15-20 minutes.** |

**Use case: Registration Confirm Message**

|  |  |
| --- | --- |
| **Use Case** | **Registration Confirm Message** |
| **Goal** | **System confirms that user has successfully registered.** |
| **Precondition** | **Registration process is completed.** |
| **Success Condition** | **User sees confirmation on screen.** |
| **Failed Condition** | **No message shown due to system failure.** |
| **Primary Actors**  **Secondary Actors** | **System** |
| **Trigger** | **Registration success.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **System detects successful registration.** | | **2** | **System displays “Registration Successful” message.** | | **3** | **System sends confirmation message.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Message delivery fails** | |  | **3.a.i** | **System logs failure and may retry sending.** | |
| **Quality Requirement** | **System must display confirmation message within 2 seconds after successful Registration processing.** |

**Use case: Login**

|  |  |
| --- | --- |
| **Use Case** | **Login** |
| **Goal** | **Registered user logs into the system.** |
| **Precondition** | **User has a valid registered account.** |
| **Success Condition** | **Message:” Successfully Login”** |
| **Failed Condition** | **Message” incorrect login”.**  **Message” Try again later”.** |
| **Primary Actors**  **Secondary Actors** | **Customer (user), Admin. Manager** |
| **Trigger** | **User submit information and details.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User enters username and password and other details.** | | **2** | **System checks information.** | | **3** | **User is authenticated and redirected.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **2** | **a** | **Information is incorrect** | |  | **3.a.i** | **System shows “Invalid login” error.** | |
| **Quality Requirement** | **Login should complete in <5 seconds.** |

**Use case: Login Confirm Message**

|  |  |
| --- | --- |
| **Use Case** | **Login Confirm Message** |
| **Goal** | **Notify user of successful login.** |
| **Precondition** | **Valid login information provided.** |
| **Success Condition** | **Confirmation shown and user continues.** |
| **Failed Condition** | **Confirmation not displayed due to system error.** |
| **Primary Actors**  **Secondary Actors** | **System** |
| **Trigger** | **Successful authentication.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **System authenticates user.** | | **2** | **Confirmation message “Login Successful” is shown.** | | **3** | **User go to dashboard.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **2** | **a** | **Confirmation message fails to load** | |  | **3.a.i** | **System retries message or skips directly to dashboard.** | |
| **Quality Requirement** | **System must display confirmation message within 2 seconds after successful login processing** |

**Use case: Password Change**

|  |  |
| --- | --- |
| **Use Case** | **Password Change** |
| **Goal** | **User can change password for their account.** |
| **Precondition** | **User is logged in.** |
| **Success Condition** | **Password is successfully updated.** |
| **Failed Condition** | **Password change fails.** |
| **Primary Actors**  **Secondary Actors** | **Customer (user),Admin**  **System database** |
| **Trigger** | **User selects "Change Password" option.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User select to change password option.** | | **2** | **User enters current and new passwords.** | | **3** | **System verifies current password.** | | **4** | **System updates password in database.** | | **5** | **System confirms update success.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Current password is incorrect** | |  | **3.a.i** | **Show error message and block update.** | | **4** | **a** | **Database update fails.** | |  | **4.a.i** | **Show error and allow retry.** | |
| **Quality Requirement** | **Response must appear within 2 minute.** |

**Use Case: Customer Dashboard**

|  |  |
| --- | --- |
| **Use Case** | **Customer Dashboard** |
| **Goal** | **User views and manages their personal profile, bookings, and payment history.** |
| **Precondition** | **User is logged into the system** |
| **Success Condition** | **Dashboard is displayed with relevant information.** |
| **Failed Condition** | **Dashboard fails to load.** |
| **Primary Actors**  **Secondary Actors** | **Customer (user)**  **Database** |
| **Trigger** | **Must log in.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User accesses the dashboard page** | | **2** | **User see system fetches user profile, past bookings, payment status etc.** | | **3** | **System fetches user profile, past bookings, payment status.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **2** | **a** | **Database unavaiiable** | |  | **2.a.i** | **System shows error and retries.** | | **3** | **a** | **No booking history found.** | |  | **3.a.i** | **Show “No records available” message.** | |
| **Quality Requirement** | **Dashboard should load within 3 seconds.** |

**Use Case: Event Selection**

|  |  |
| --- | --- |
| **Use Case** | **Event Selection** |
| **Goal** | **Customer selects a specific type of event to book (e.g., wedding, seminar) etc.** |
| **Precondition** | **User is logged in and viewing event options** |
| **Success Condition** | **Event booking successfully.** |
| **Failed Condition** | **Failed event booking.** |
| **Primary Actors**  **Secondary Actors** | **Customer(user)** |
| **Trigger** | **Go to the “Book Event” section.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User views available event types.** | | **2** | **User selects a specific event category.** | | **3** | **Input all event information.** | | **4** | **System confirm message booking successfully.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **4** | **a** | **Selected event type unavailable** | |  | **4.a.i** | **Invalid booking.** | |
| **Quality Requirement** | **Response within 3 seconds of click.** |

**Use case: Marriage Event Booking**

|  |  |
| --- | --- |
| **Use Case** | **Marriage Event Booking** |
| **Goal** | **Customer books a marriage event by selecting date, venue, and services.** |
| **Precondition** | **User is logged in** |
| **Success Condition** | **Event is successfully booked and confirmed.** |
| **Failed Condition** | **Booking not completed.** |
| **Primary Actors**  **Secondary Actors** | **Customer**  **Admin** |
| **Trigger** | **User select “Book Marriage Event.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User Goes to dashboard** | | **2** | **User selects marriage event type.** | | **3** | **User chooses date, time, and venue etc.** | | **4** | **User selects services (decoration, catering, etc.).** | | **5** | **Adman check and confirms booking.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Selected date/venue unavailable** | |  | **3.a.i** | **Show alternate options.** | | **4** | **a** | **Invalid service selection.** | |  | **4.a.i** | **Try again.** | |
| **Quality Requirement** | **Booking confirmation must be provided within 5 minutes.** |

**Use Case: Cultural Event Booking**

|  |  |
| --- | --- |
| **Use Case** | **Cultural Event Booking** |
| **Goal** | **Customer books a cultural event.** |
| **Precondition** | **User is logged in.** |
| **Success Condition** | **Cultural event is booked and stored.** |
| **Failed Condition** | **Booking attempt fails.** |
| **Primary Actors**  **Secondary Actors** | **Customer**  **Admin** |
| **Trigger** | **User selects “Cultural Event” booking option.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User goes to dashboard.** | | **2** | **User selects cultural event option.** | | **3** | **User enters event details, number of guests, and selects date etc.** | | **4** | **System checks service and venue availability.** | | **5** | **Adman check and confirms booking.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Selected date/venue unavailable.** | |  | **3.a.i** | **Show alternate options.** | | **4** | **a** | **Invalid service selection.** | |  | **4.a.i** | **Try again.** | |
| **Quality Requirement** | **Booking confirmation must be provided within 5 minutes.** |

**Use Case: Institution Event Booking**

|  |  |
| --- | --- |
| **Use Case** | **Institution Event Booking** |
| **Goal** | **User books an institutional event (e.g., seminar, workshop).** |
| **Precondition** | **Logged-in user.** |
| **Success Condition** | **Institutional event is confirmed.** |
| **Failed Condition** | **Booking attempt fails.** |
| **Primary Actors**  **Secondary Actors** | **Customer(user)**  **Admin** |
| **Trigger** | **“Institution Booking” option selected.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User goes to dashboard.** | | **2** | **User chooses event type (seminar, workshop).** | | **3** | **User inputs event description, date, capacity etc.** | | **4** | **System checks service and venue availability.** | | **5** | **Adman check and confirms booking.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Selected date/venue unavailable** | |  | **3.a.i** | **Show alternate options.** | | **4** | **a** | **Invalid service selection.** | |  | **4.a.i** | **Try again.** | |
| **Quality Requirement** | **Booking confirmation must be provided within 5 minutes.** |

**Use Case: Birthday Event Booking**

|  |  |
| --- | --- |
| **Use Case** | **Birthday Event Booking** |
| **Goal** | **Customer books a birthday celebration event.** |
| **Precondition** | **User is logged in.** |
| **Success Condition** | **Birthday event successfully scheduled.** |
| **Failed Condition** | **Booking attempt fails.** |
| **Primary Actors**  **Secondary Actors** | **Customer(user)**  **Admin** |
| **Trigger** | **User chooses “Birthday Event” option.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User goes to dashboard.** | | **2** | **User selects birthday event.** | | **3** | **Inputs child/adult name, date, and guest number etc.** | | **4** | **Selects available themes or services.** | | **5** | **Adman check and confirms booking.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Selected date/venue unavailable** | |  | **3.a.i** | **Show alternate options.** | | **4** | **a** | **Invalid service selection.** | |  | **4.a.i** | **Try again.** | |
| **Quality Requirement** | **Booking confirmation must be provided within 5 minutes.** |

**Use Case: View Offer for Discount**

|  |  |
| --- | --- |
| **Use Case** | **View Offer for Discount** |
| **Goal** | **Customer views available discounts or promotional offers for events.** |
| **Precondition** | **Customer is logged in.** |
| **Success Condition** | **Discounts are displayed based on eligibility or event type.** |
| **Failed Condition** | **Offers are not shown due to system error or no offers available.** |
| **Primary Actors**  **Secondary Actors** | **Customer(user)** |
| **Trigger** | **Customer select “View Offers” or enters offer section.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Go to offer section.** | | **2** | **Customer accesses offer section.** | | **3** | **User view current discounts based on user or event type.** | | **4** | **System displays available offers.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **No offers found** | |  | **3.a.i** | **Display message “No discounts currently available.”** | | **3** | **b** | **Offer fetch fails** | |  | **3.b.i** | **Display retry option or error.** | |
| **Quality Requirement** | **Offers should load in under 3 seconds.** |

**Use Case: Event Confirm Message**

|  |  |
| --- | --- |
| **Use Case** | **Event Confirm Message** |
| **Goal** | **customer see that their event has been successfully booked.** |
| **Precondition** | **Event booking process is completed.** |
| **Success Condition** | **Message “ confirm booking”.** |
| **Failed Condition** | **Message “ reject booking” ,try again.** |
| **Primary Actors**  **Secondary Actors** | **customer**  **System database.** |
| **Trigger** | **Select conformation option.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Choice conformation option.** | | **2** | **Go to conformation** | | **3** | **View any massage sent by admin.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Admin doesn’t send any massage.** | |
| **Quality Requirement** | **Admin sent Booking confirmation message.** |

**Use Case: View Event Information**

|  |  |
| --- | --- |
| **Use Case** | **View Event Information** |
| **Goal** | **Customer views detailed information about events including date, venue, services etc.** |
| **Precondition** | **Events are listed in the system** |
| **Success Condition** | **Event details are shown accurately.** |
| **Failed Condition** | **Event details are not shown accurately.** |
| **Primary Actors**  **Secondary Actors** | **Customer(user)** |
| **Trigger** | **User selects an event to view details.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Go to view event information section.** | | **2** | **User clicks on a listed event.** | | **3** | **System retrieves event data.** | | **4** | **System displays full event information.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Event data missing** | |  | **3.a.i** | **Show error** | |  | **3.a.ii** | **System displays full event information.** | |
| **Quality Requirement** | **Event data must display within 3 seconds.** |

**Use Case: Select Venue**

|  |  |
| --- | --- |
| **Use Case** | **Select Venue** |
| **Goal** | **Customer chooses a venue for their selected event.** |
| **Precondition** | **Event is selected and venue data is available.** |
| **Success Condition** | **Venue is selected and reserved** |
| **Failed Condition** | **Venue is not selected or unavailable** |
| **Primary Actors**  **Secondary Actors** | **Customer**  **Admin** |
| **Trigger** | **User accesses venue selection form** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User select venue selection option** | | **2** | **User fill-up venue selection form** | | **3** | **System shows available venues based on date and capacity** | | **4** | **User selects a venue** | | **5** | **System confirm reserves the venue.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **No venues available** | |  | **3.a.i** | **select alternative dates or venues** | | **5** | **a** | **Venue reservation fails** | |  | **5.a.i** | **Try again.** | |
| **Quality Requirement** | **Venue availability check must complete in 4 seconds.** |

**Use Case: Payment Process**

|  |  |
| --- | --- |
| **Use Case** | **Payment Process** |
| **Goal** | **User initiates payment to complete event booking.** |
| **Precondition** | **Event and venue are selected; total amount is calculated.** |
| **Success Condition** | **Payment is processed successfully** |
| **Failed Condition** | **Payment fails due to technical or user input error.** |
| **Primary Actors**  **Secondary Actors** | **Customer(user)**  **system** |
| **Trigger** | **User clicks “Proceed to Payment.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Go to payment section.** | | **2** | **User reviews booking summary.** | | **3** | **Clicks on payment.** | | **4** | **User enters payment information and confirms** | | **5** | **Payment is processed and user is redirected to confirmation screen.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **4** | **a** | **Payment declined.** | |  | **4.a.i** | **Error try again.** | | **5** | **a** | **System does not work.** | |  | **5.a.i** | **Try again.** | |
| **Quality Requirement** | **Payment process should not exceed 10 seconds.** |

**Use Case: Payment Method**

|  |  |
| --- | --- |
| **Use Case** | **Payment Method** |
| **Goal** | **Customer selects a method of payment (e.g., card, mobile banking, cash).** |
| **Precondition** | **User has begun the checkout process.** |
| **Success Condition** | **Selected method is passed to payment gateway.** |
| **Failed Condition** | **No method selected.** |
| **Primary Actors**  **Secondary Actors** | **Customer(user)**  **System** |
| **Trigger** | **User select “Choose Payment Method.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Go to payment option** | | **2** | **Select a method.** | | **3** | **Confirm payment** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **2** | **a** | **Method unavailable.** | |  | **2.a.i** | **Try again.** | |
| **Quality Requirement** | **Payment method page should load in under 3 seconds.** |

**Use Case: Payment Confirmation Message**

|  |  |
| --- | --- |
| **Use Case** | **Payment Confirmation Message** |
| **Goal** | **Notify user of successful payment.** |
| **Precondition** | **Payment is processed successfully.** |
| **Success Condition** | **Message “payment successfully”.** |
| **Failed Condition** | **Message “payment not successfully”.** |
| **Primary Actors**  **Secondary Actors** | **System** |
| **Trigger** | **Payment gateway confirms transaction.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Go to payment option.** | | **2** | **Select one method.** | | **3** | **Shows “Payment Successful” message.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Notification service unavailable** | |  | **3.a.i** | **Show message “Again payment”.** | |
| **Quality Requirement** | **System must display confirmation message within 2 seconds after successful payment processing** |

**Use Case: Service Request**

|  |  |
| --- | --- |
| **Use Case** | **Service Request** |
| **Goal** | **User submits a custom request for services (e.g., catering, decoration).** |
| **Precondition** | **User is authenticated** |
| **Success Condition** | **Request is submitted to service team.** |
| **Failed Condition** | **Submission fails.** |
| **Primary Actors**  **Secondary Actors** | **Customer(user)** |
| **Trigger** | **User select “Request a Service.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Go to service option.** | | **2** | **User fills in service request form.** | | **3** | **Submits the form.** | | **4** | **System stores request and notifies service team.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **2** | **a** | **Does not fill all information.** | |  | **2.a.i** | **Fill-up from again.** | |  | **2.a.ii** | **Invalid input** | |
| **Quality Requirement** | **Admin check services request and confirm message.** |

**Use Case: Equipment Rental**

|  |  |
| --- | --- |
| **Use Case** | **Equipment Rental** |
| **Goal** | **User rents items like lights, sound systems, etc.** |
| **Precondition** | **User is logged in.** |
| **Success Condition** | **Equipment request successfully.** |
| **Failed Condition** | **Request not submitted.** |
| **Primary Actors**  **Secondary Actors** | **Customer(user)** |
| **Trigger** | **User select “Equipment Rental” section.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Go to Equipment Rental.** | | **2** | **User selects required items and duration.** | | **3** | **Confirms rental request** | | **4** | **System checks availability and confirms booking.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **4** | **a** | **Equipment unavailable.** | |  | **3.a.i** | **Try another date.** | |  | **3.a.ii** | **Try again later.** | |
| **Quality Requirement** | **Admin check request and confirm message.** |

**Use Case: Submit Feedback**

|  |  |
| --- | --- |
| **Use Case** | **Submit Feedback** |
| **Goal** | **Customer shares their opinion on service or experience.** |
| **Precondition** | **User completed an event booking.** |
| **Success Condition** | **Feedback is stored and available to admin** |
| **Failed Condition** | **Feedback submission fails.** |
| **Primary Actors**  **Secondary Actors** | **Customer** |
| **Trigger** | **User select “Submit Feedback.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User go to feedback option.** | | **2** | **User writes message and submits.** | | **3** | **System validates and stores feedback.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Feedback does not submit.** | |  | **3.a.i** | **Try again.** | |
| **Quality Requirement** | **Confirmation message within 2 seconds.** |

**Use Case: Admin Dashboard**

|  |  |
| --- | --- |
| **Use Case** | **Admin Dashboard** |
| **Goal** | **Admin monitors bookings, users, feedback, and services etc.** |
| **Precondition** | **Admin is logged in.** |
| **Success Condition** | **Dashboard shows data overview** |
| **Failed Condition** | **Admin panel fails to load.** |
| **Primary Actors**  **Secondary Actors** | **Admin**  **System database** |
| **Trigger** | **Admin select “Dashboard.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Admin go to dashboard.** | | **2** | **Enter password ad user name.** | | **3** | **Login successfully.** | | **4** | **System show event details.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Invalid user name or password.** | |  | **3.a.i** | **Try again.** | | **4** | **a** | **System doesn’t show event details.** | |
| **Quality Requirement** | **Load time <3 seconds.** |

**Use Case : Find Event Details**

|  |  |
| --- | --- |
| **Use Case** | **Find Event Details** |
| **Goal** | **Admin view specific event details** |
| **Precondition** | **Admin is logged in and User add events.** |
| **Success Condition** | **Event data is shown.** |
| **Failed Condition** | **Event not found.** |
| **Primary Actors**  **Secondary Actors** | **Admin , Manager**  **Event database.** |
| **Trigger** | **Admin go to event details.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Admin go to event details section.** | | **2** | **Admin enters event ID or event name.** | | **3** | **system displays the event.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **No event found** | |  | **3.a.i** | **Show “No results” message.** | |
| **Quality Requirement** | **Confirmation within 20 seconds.** |

**Use Case: Delete Event Details**

|  |  |
| --- | --- |
| **Use Case** | **Delete Event Details** |
| **Goal** | **Admin deletes any event .** |
| **Precondition** | **Admin is logged in and User add event.** |
| **Success Condition** | **Event is removed from system** |
| **Failed Condition** | **Deletion fails.** |
| **Primary Actors**  **Secondary Actors** | **Admin, Manager** |
| **Trigger** | **Admin select “Delete” on an event.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Admin Go to event delete section.** | | **2** | **Admin selects an event.** | | **3** | **Confirms deletion** | | **4** | **System removes data** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Deletion fails.** | |
| **Quality Requirement** | **Confirmation within 2 seconds.** |

**Use Case: View all Event Details**

|  |  |
| --- | --- |
| **Use Case** | **View All Details** |
| **Goal** | **Admin reviews all booking and user details** |
| **Precondition** | **Admin is logged in.** |
| **Success Condition** | **All event list show on screen.** |
| **Failed Condition** | **All event list doesn’t show on screen.** |
| **Primary Actors**  **Secondary Actors** | **Admin, Manager**  **System** |
| **Trigger** | **Admin selects “View All.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Go to event details.** | | **2** | **Select view all event.** | | **3** | **System shows all events.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **All event doesn’t show.** | |  | **3.a.i** | **Please try again.** | |
| **Quality Requirement** | **Display complete data in <5 seconds.** |

**Use Case: Event Approval**

|  |  |
| --- | --- |
| **Use Case** | **Event Approval** |
| **Goal** | **Admin approves or rejects pending event bookings.** |
| **Precondition** | **Event Pending bookings exist.** |
| **Success Condition** | **Event is either approved or rejected.** |
| **Failed Condition** | **Action not saved.** |
| **Primary Actors**  **Secondary Actors** | **Admin. Manager** |
| **Trigger** | **Admin selects “Pending Events.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Admin go to approval option.** | | **2** | **Admin opens pending list.** | | **3** | **Select approve or reject** | | **4** | **System updates status.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **4** | **a** | **Update fails** | |  | **4.a.i** | **Show error, allow retry** | |
| **Quality Requirement** | **1. Status updated in 1 minute.** |

**Use Case: Manage Discount and Offer**

|  |  |
| --- | --- |
| **Use Case** | **Manage Discount and Offer** |
| **Goal** | **Admin creates, updates, or removes discounts** |
| **Precondition** |  |
| **Success Condition** | **Discount changes are saved.** |
| **Failed Condition** | **Save fails.** |
| **Primary Actors**  **Secondary Actors** | **Admin, Manager** |
| **Trigger** | **Admin select on “Discounts & Offers.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Admin go to Discounts & Offers section** | | **2** | **Admin creates or edits a discount.** | | **3** | **Saves changes.** | | **4** | **Show update confirm massage.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Save fails** | |  | **3.a.i** | **Retry again.** | | **4** | **a** | **Does not show any massage.** | |
| **Quality Requirement** | **Update confirmation <2 seconds** |

**Use Case: View Feedback History**

|  |  |
| --- | --- |
| **Use Case** | **View Feedback History** |
| **Goal** | **Admin views all user feedback submitted** |
| **Precondition** | **Feedback has been submitted by user.** |
| **Success Condition** | **All feedback entries are listed.** |
| **Failed Condition** | **Feedback list fails to load.** |
| **Primary Actors**  **Secondary Actors** | **Admin, Manager** |
| **Trigger** | **Admin accesses “Feedback History.”** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **Admin go to feedback option.** | | **2** | **Admin opens feedback file.** | | **3** | **System display all feedback.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **System doesn’t work** | |  | **3.a.i** | **System doesn’t show any feedback.** | |
| **Quality Requirement** | **List loads in <3 seconds.** |

**Use Case: Contact Us**

|  |  |
| --- | --- |
| **Use Case** | **Contact Us** |
| **Goal** | **User contact to admin/support.** |
| **Precondition** | **User must login.** |
| **Success Condition** | **User see contact number or email.** |
| **Failed Condition** | **User can not see contact number or email.** |
| **Primary Actors**  **Secondary Actors** | **Customer(User)** |
| **Trigger** | **User select “Contact Us” option.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User go to contact option** | | **2** | **Then user see contact number and email.** | | **3** | **User use this number to contact support team or admin.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **User can not see any contact number or email** | |
| **Quality Requirement** | **Admin set contact number and email.** |

**Use Case: Rules and Conditions**

|  |  |
| --- | --- |
| **Use Case** | **Rules and Conditions** |
| **Goal** | **Display the rules, terms, and conditions associated with event booking and service usage to the user.** |
| **Precondition** | **Log in system** |
| **Success Condition** | **Rules and conditions are successfully displayed to the user.** |
| **Failed Condition** | **Rules and conditions are doesn’t show displayed to the user.** |
| **Primary Actors**  **Secondary Actors** | **User (Customer)** |
| **Trigger** | **User clicks on “Rules and Conditions” or is prompted before confirming a booking.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User navigates to the “Rules and Conditions” section, or is shown it during booking.** | | **2** | **System loads the latest rules and displays them.** | | **3** | **User reads and acknowledges the terms before proceeding.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **2** | **a** | **Rules content fails to load** | |  | **2.a.i** | **Show error message and provide a retry option.** | | **3** | **a** | **User does not accept rules** | |  | **3.a.i** | **System prevents further action (e.g., booking cannot be submitted).** | |
| **Quality Requirement** | **Rules should be fully loaded in less than 2 seconds.** |

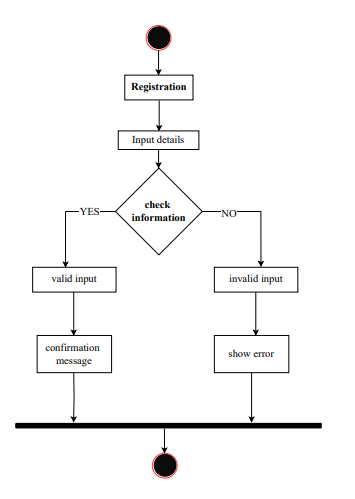
**Use Case: Logout**

|  |  |
| --- | --- |
| **Use Case** | **Logout** |
| **Goal** | **User must be log out system.** |
| **Precondition** | **User is must log in.** |
| **Success Condition** | **Message ”log out successfully”** |
| **Failed Condition** | **Message “Logout process fails try again”** |
| **Primary Actors**  **Secondary Actors** | **Customer(user),Admin. Manager** |
| **Trigger** | **User select “Logout” Option.** |
| **Description/Main Success Scenario** | |  |  | | --- | --- | | **1** | **User go to logout option.** | | **2** | **User select logout option.** | | **3** | **User log out homepage.** | |
| **Alternative Flows** | |  |  |  | | --- | --- | --- | | **3** | **a** | **Log out fail try again.** | |
| **Quality Requirement** | **Logout system in 2 seconds.** |

### 5.3 Activity Diagram

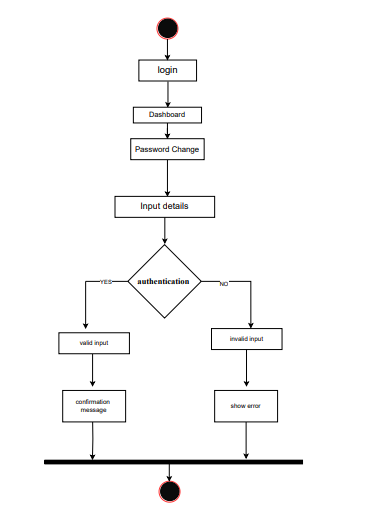
**Activity Diagram: Registration**

The activity diagram below illustrates the process of Registration:

****

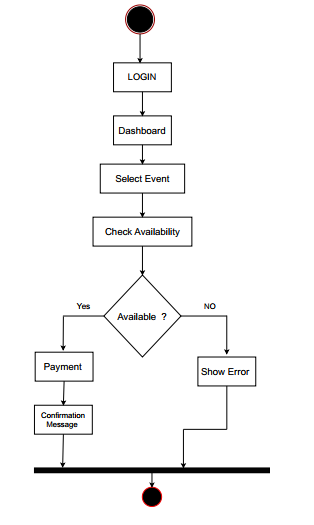
**Activity Diagram: Login**

The activity diagram below illustrates the process of Login:

****

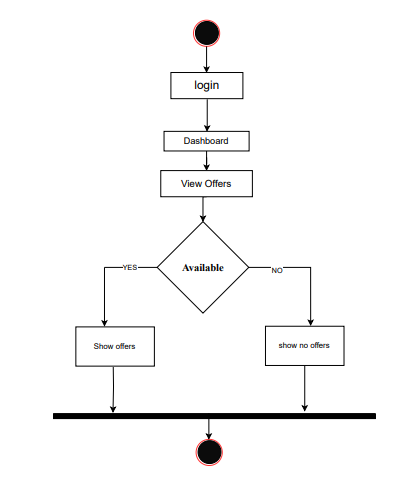
**Activity Diagram: Select Event**

The activity diagram below illustrates the process of a customer select a event :

****

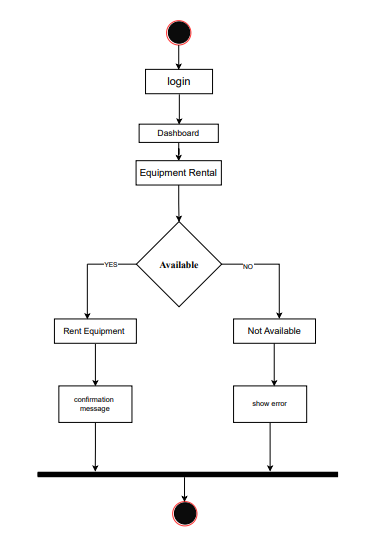
**Activity Diagram: Event offers**

The activity diagram below illustrates the process of event offers:



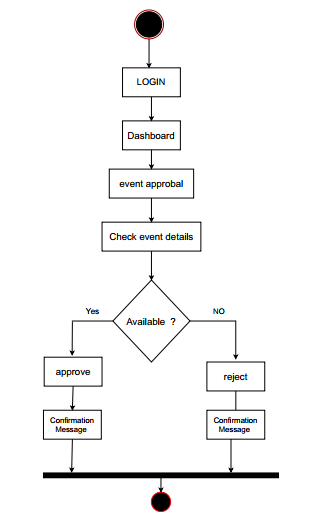
**Activity Diagram: Equipment Rental**

The activity diagram below illustrates the process of a customer Equipment Rental:



**Activity Diagram: Check event**

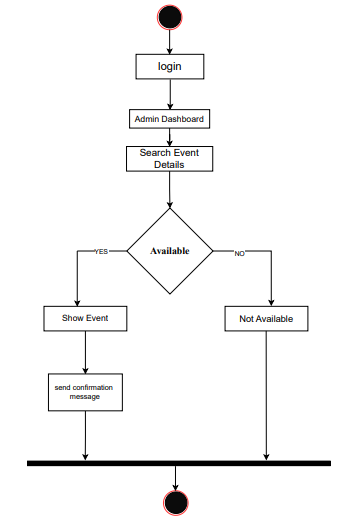
The activity diagram below illustrates the process of a customer check event :

****

**Event Approval**

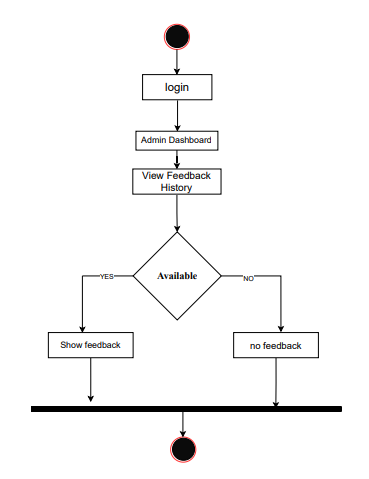
**Activity Diagram: Search Event details**

The activity diagram below illustrates the process of Search Event details :

****

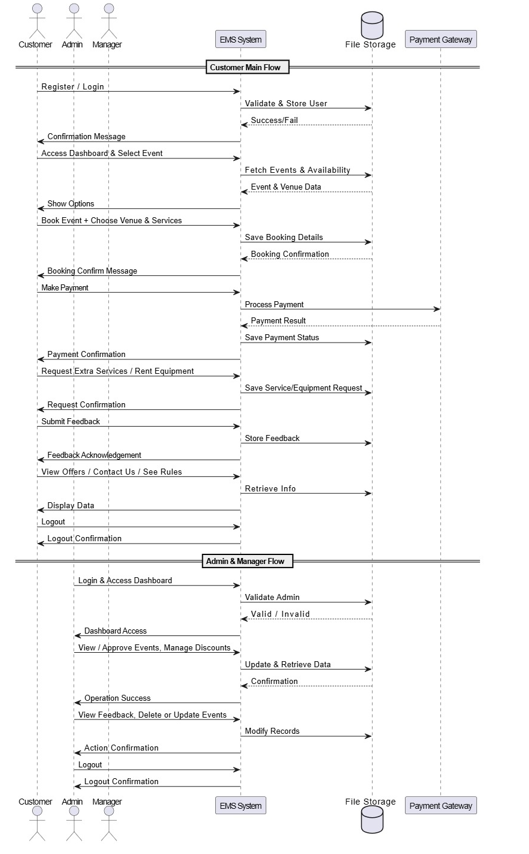
**Activity Diagram: View feedback history**

The activity diagram below illustrates the process of View feedback history:



### 5.4 Sequence Diagram

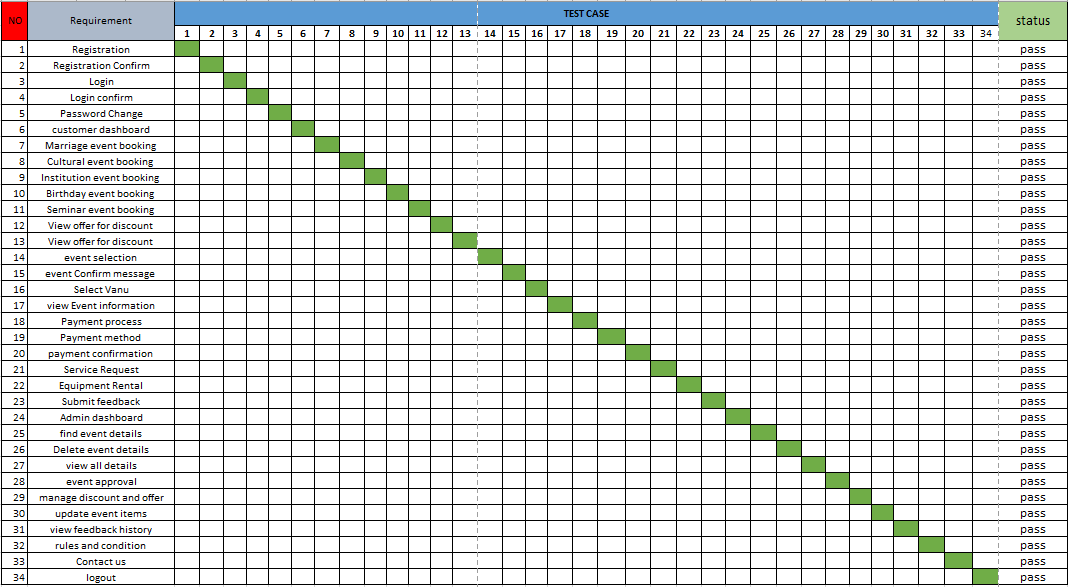
In this sequence diagram show all process:

****

# **Chapter -- 6**

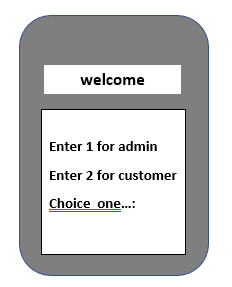
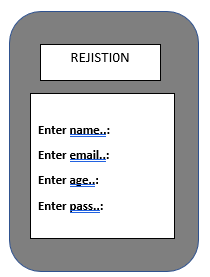
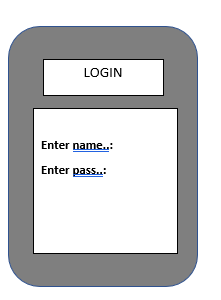
### 6.1 Requirements Traceability Matrix – RTM

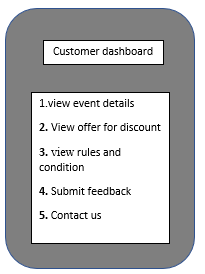
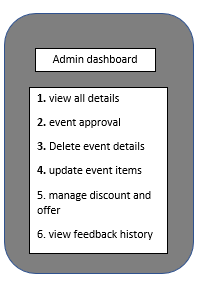
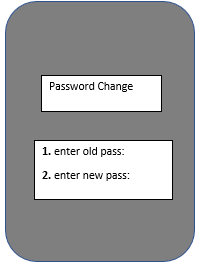
|  |  |
| --- | --- |
| **Test Case ID** | **Test Case Description** |
|  |  |
| TC-001 | Verify user can register with valid data |
| TC-002 | Confirm registration success message is shown |
| TC-003 | Verify user login with valid credentials |
| TC-004 | Confirm login success message is shown |
| TC-005 | Verify password can be changed successfully |
| TC-006 | Check user can access customer dashboard after login |
| TC-007 | Verify user can book marriage event |
| TC-008 | Verify user can book cultural event |
| TC-009 | Verify user can book institution event |
| TC-010 | Verify user can book birthday event |
| TC-011 | Verify user can book seminar event |
| TC-012 | Check user can view available discounts and offers |
| TC-013 | Confirm message appears after successful event booking |
| TC-014 | Check user can select event type from options |
| TC-015 | Verify user can view detailed info for selected event |
| TC-016 | Check user can select a venue for an event |
| TC-017 | Verify end-to-end payment process |
| TC-018 | Check different payment methods (Card, Mobile Banking,etc.) |
| TC-019 | Confirm payment success message is displayed |
| TC-020 | Verify user can request additional services |
| TC-021 | Check user can rent equipment for an event |
| TC-022 | Verify user can submit event feedback |
| TC-023 | Check admin can access the dashboard |
| TC-024 | Admin can search for specific event details |
| TC-025 | Admin can delete event details |
| TC-026 | Admin can view all event-related data |
| TC-027 | Admin can approve or reject submitted events |
| TC-028 | Admin can add, edit, or delete discount offers |
| TC-029 | Admin can update event-related equipment or services |
| TC-030 | Admin can view feedback submitted by users |
| TC-031 | Check user can view rules and conditions |
| TC-032 | Verify contact form submission works and message is sent |
| TC-033 | Verify user can successfully log out from the system |
| TC-034 | Verify user can see rules and regulations from the system. |

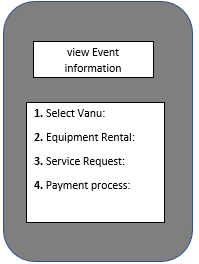
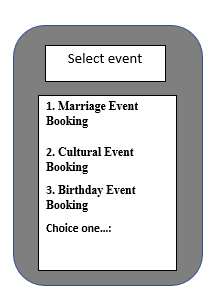
****

# **Chapter – 7**

### 7.1 Prototyping:

****

****

****

**Marriage Event Booking**

**1. Enter Place:**

**2. Date:**

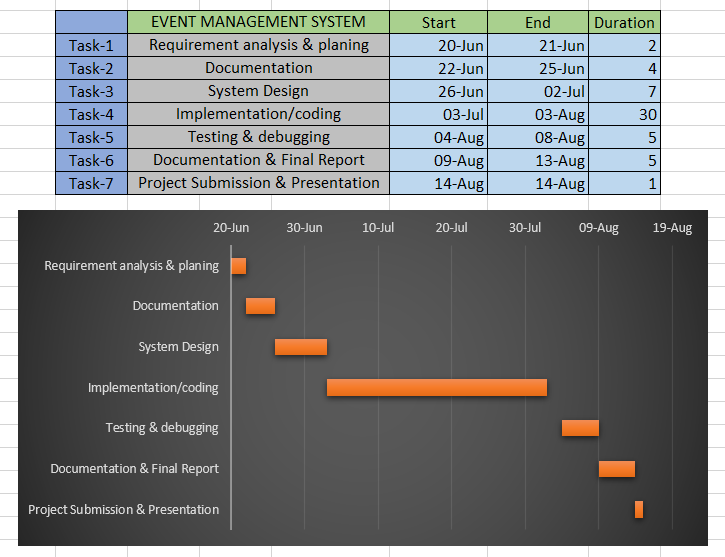
**3. Total Member**

**4. Budget:**

**Choice one…:**

# **Chapter -- 8**

### 8.1 Gantt Chart:

****

# **Chapter – 9**

### 9.1 Conclusion

The Event Management System is a robust, file-based software solution developed in C to automate and manage the full lifecycle of event organization, from registration to feedback. The system provides a modular, role-based interface for different users—customers, managers, and admins—ensuring a secure and personalized experience. By implementing both functional and non-functional requirements, it guarantees not only operational correctness but also performance, reliability, and usability.

With features like venue booking, service and equipment requests, discount management, and real-time confirmations, the system significantly improves efficiency over traditional manual processes. It is well-suited for educational, organizational, or small business use and lays the foundation for further development, such as web integration or mobile extension.