SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Event Management System**

**Prepared by:-**

*Dheeranraj S*

*Bhuvaneswaran M*

*Rajarajan P*

# Introduction

## Purpose

The main objective of this document is to illustrate the requirements of the project Event Management system. The document gives the detailed description of the both functional and non-functional requirements proposed by the client.The purpose of this project is to provide When choosing or developing an event management system, it's essential to prioritize features based on the specific needs of your events and attendees. This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

## Document Conventions

* + - Entire document should be justified.
    - Convention for Main title

Font face: Times New Roman Font style: Bold

Font Size: 14

* + - Convention for Sub title

Font face: Times New Roman Font style: Bold

Font Size: 12

* + - Convention for body

Font face: Times New Roman Font Size: 12

## Scope of Development Project

The objective of this application is to develop a system that effectively manages all the data related to the various events that take place in an organization. The purpose is to maintain a centralized database of all event related information. The goal is to support various functions and processes necessary to manage the data efficiently

## Definitions, Acronyms and Abbreviations

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

## References

* + - Books

 A Professional & Development Approach; Ashutosh Chaturvedi

 Software Requirements (Microsoft) Second EditionBy Karl E. Wiegers

Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman

* Corporate Event Project Management: William O'Toole, Phyllis Mikolaitis
  + - Websites

[**http://www.slideshare.net/**](http://www.slideshare.net/)

[**http://ebookily.net/doc/srs-library-management-system**](http://ebookily.net/doc/srs-library-management-system)

# Overall Descriptions

## Product Perspective

Use Case Diagram of Library Management System

*searches*

1

1 *requests*

1

1

1..\*

\*

search\_book



1..\*

check\_limit

check\_availability

User 1

issue\_book

*request\_renew*

<<include>>

*monitors\_request*

1

*monitors\_renew* 1

1

*performs*

*give\_book*

<<include>>

0..\*

1..\*

renew\_book

verify\_member

<<include>>

*take\_book*

1

1 Librarian

Student

0..\*

1..\*

*adds\_new\_book*

*perform\_transaction\_updation*

Staff

\*

\*

return\_book

View\_logs

<<extend>>

add\_book

\*

calculate\_fine

update\_record

This is a broad level diagram of the project showing a basic overview. The users can be either staff or student.. This System will provide a search functionality to facilitate the search of resources. This search will be based on various categories viz. book name or the ISBN. Further the library staff personnel can add/update the resources and the resource users from the

system.The users of the system can request issue/renew/return of books for which they would have to follow certain criteria.

## Product Function

Entity Relationship Diagram of Library Management System



The Online Library System provides online real time information about the books available in the Library and the user information. The main purpose of this project is to reduce the manual work. This software is capable of managing Book Issues, Returns, Calculating/Managing Fine, Generating various Reports for Record-Keeping according to end user requirements. The Librarian will act as the administrator to control members and manage books. The member’s status of issue/return is maintained in the library database. The member’s details can be fetched by the librarian from the database as and when required. The valid members are also allowed to view their account information.

## User Classes and Characteristics

The system provides different types of services based on the type of users [Client/Management].

* Event Organizers (Admins)

Role: Admins have privileged access to the system. They manage events, user accounts, and overall

system functionality.

* Event Participants (Clients):

Role: Clients interact with the system to register for events and access event-related information.

* Event Crews:

Role: Crew members assist during events (e.g., event staff, technical support)

* System Users (General):

Role: General users who interact with the system (e.g., browsing events, exploring options).

* Event Types:

Role: Represents different categories of events (e.g., conferences, concerts, workshops).

* Payment Gateway:

Role: Handles financial transactions related to event registrations.

* Reports Generator:

Role: Generates event-related reports (e.g., attendance, revenue).

* Pricing Module:

Role: Determines event ticket prices.

## Operating Environment

The product will be operating in windows environment. The Event Management System is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer,Google Chrome,and Mozilla Firefox.Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

## Assumptions and Dependencies

The assumptions are:-

* + - The coding should be error free
    - The system should be user-friendly so that it is easy to use for the users
    - The information of all clients and events must be stored in a database that is accessible by the website
    - The system should have more storage capacity and provide fast access to the database
    - The system should support quick transactions
    - The Event System is running 24 hours a day
    - Users may access from any computer that has Internet browsing capabilities and an

Internet connection

* + - Users must have their correct usernames and passwords to enter into their account

The dependencies are:-

* + - The specific hardware and software due to which the product will be run
    - On the basis of listing requirements and specification the project will be developed and run
    - The end users (admin) should have proper understanding of the product
    - The system should have the general report stored
    - The information of all the users must be stored in a database that is accessible by the Event Management System

## Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows 98, Windows XP Language: Java Runtime Environment, Net beans 7.0.1 (front end) Database: MS SQL Server (back end)

Hardware Configuration:- Processor: Pentium(R)Dual-core CPU Hard Disk: 40GB

RAM: 256 MB or more

## Data Requirement

The inputs consist of the query to the database and the output consists of the solutions for the query. The output also includes the user receiving the details of their accounts. In this project the inputs will be the queries as fired by the users like create an account and putting into account. Now the output will be visible when the user requests the server to get details of their account in the form of time, date and current status of the event.

# External Interface Requirement

## GUI

* User Interface Requirements:

The user interface (UI) is the bridge between the software and its users. It encompasses the visual design, layout, and interaction elements. Some examples of UI requirements include:

Screenshots or screen layouts: Provide visual representations of each separate part (e.g., pages of a website) of the software.

User input forms: Specify how users will input data (e.g., event details, attendee information).

Navigation menus: Define how users can navigate through different sections of the event management system.

* Hardware Interfaces Requirements:

These requirements describe the logical and physical characteristics of the interfaces between the software and the hardware components. Consider the following aspects:

Required hardware: Specify any hardware components needed for the system to function. For instance, if your event management system uses thumb identification for attendance, mention the necessary thumb recognition hardware.

Supported device types: Indicate the operating systems or platforms the software will run on (e.g., Windows 32-bit, 64-bit, Android, iOS).

Data and control interactions: Describe how the software communicates with the hardware. For example, an ATM machine software identifies users based on fingerprint or login details provided to the thumb recognition hardware within the machine.

* Software Interfaces Requirements:

These requirements address the connections between your software and other software systems. Consider the following:

Operating system compatibility: Specify which operating systems your software will support (e.g., Android, iOS, Windows 7, 8, 10).

Library dependencies: If your event management system involves face detection, mention relevant image-processing Python libraries (e.g., OpenCV, NumPy, SciPy).

Connections with other tools or plugins: If your system integrates with third-party tools (e.g., WordPress plugins), document those connections.

* Communications Interfaces Requirements:

These interfaces relate to communication functions required by the software. Examples include:

Web browsers: Specify compatibility with different browser versions (e.g., Opera V1, V2) or any required browser extensions (e.g., for flash effects).

Communication standards and protocols: Address network server communications (e.g., HTTP, TCP/IP).

# System Features

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

* User authentication and validation of members using their unique member ID
* Proper monitoring by the administrator which includes event creation, scheduling, and promotion. Attendee registration and ticket sales. Ticket scanning and Check-In.
* Proper accountability which includes not allowing a member to see other member’s account. Only administrator will see and manage all member accounts

# Other Non-functional Requirements

## Performance Requirement

Reliability: The system must be reliable, ensuring smooth operation even under heavy loads.

Response Time: Web pages should load within a few seconds to provide a seamless user experience.

## Safety Requirements:

Data Integrity: The system must maintain accurate and consistent data.

User Authentication: Proper authentication mechanisms should be in place to ensure authorized access.

Backup: Regular database backups are essential to prevent data loss.

## Security Requirements:

Access Control: Define user roles and permissions to restrict unauthorized access.

Encryption: Sensitive data (such as payment information) should be encrypted.

Protection Against Attacks: Implement measures to prevent security breaches (e.g., SQL injection, cross-site scripting).

## Maintainability:

The system should be designed for easy maintenance and updates.

Clear code documentation and modular architecture contribute to maintainability.

Portability:

Consider compatibility with different platforms (e.g., Windows, Android, iOS).

Ensure the system can be deployed across various environments.

## Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data.This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

## User Requirement

* Ease of Use:

The user interface (UI) should be intuitive and user-friendly.

Participants, organizers, and administrators should find it easy to navigate, register for events, and manage their profiles.

* Efficiency:

The system should streamline event-related tasks.

Quick registration processes, efficient check-in procedures, and seamless communication enhance efficiency.

* Connectivity:

Event planners, team members, and attendees need to stay connected.

Real-time updates, notifications, and collaboration tools are crucial.

* Mobility:

Users expect access from various devices (desktops, tablets, smartphones).

A responsive design ensures usability across platforms.

* Compatibility:

The system should work well with different browsers (e.g., Mozilla Firefox, Google Chrome, Internet Explorer).

Compatibility with operating systems (Windows, macOS, Android, iOS) is essential.

# Other Requirements

## Data and Category Requirement

Depending upon the category of user the access rights are decided.It means if the user is an administrator then he can be able to modify the data,delete, append etc. All other users except the Adminstrator only have the rights to retrieve the information about database. The categories and the data related to each category should be coded in the particular format.

## Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance,Perspective,Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

## Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

* + - Administrator: A login id representing a user with user administration privileges to the software
    - User: A general login id assigned to most users
    - Client: Intended users for the software
    - SQL: Structured Query Language; used to retrieve information from a database
    - SQL Server: A server used to store data in an organized format
    - Layer: Represents a section of the project
    - User Interface Layer: The section of the assignment referring to what the user interacts with directly
    - Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
    - Data Storage Layer: The section of the assignment referring to where all data is recorded
    - Use Case: A broad level diagram of the project showing a basic overview
    - Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system’s cases, their attributes, and the relationships between the classes
    - Interface: Something used to communicate across different mediums
    - Unique Key: Used to differentiate entries in a database

## Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes’ structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes

which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here ‘Librarian’, ‘Member’ and ‘Books’ are the most important classes which are related to other classes.

