Event Management System

SRS Document

Purpose of Project :

This document contains the complete software requirements for Event Management System and describe the design decisions, architectural design needed to implement the system. It provides the visibility in design and provide
information needed for software support. It will provide the clients of the firm with an easy-to-use platform for communicating with the organizers and also reducing the firm's labor load (*taking orders will be digitized*). Our purpose is to override the problems prevailing in the practicing manual system and provide OUR clients with a system which will link them to their account, store and categorize their orders (*according to dates/ most time required for planning/level of urgency etc.*) so they can function efficiently and provide their best service.

Our Management System is not only digitalizing the booking of orders providing ease of access to the customers, but will also digitalize the firm's functionalities such as adding/removing employees, vendors etc. In short, our management system will digitalize the whole firm along with its jobs.

Scope of Project :

The scope of our EMS is to automate the existing manual system by the help of computerized equipment and a software that is fulfilling their requirements so their information can be stored for a longer period along with ease of access to that information. It manages the details of events, organizers, attendees, payments etc. The EMS is built at administrative access, hence only the manager is guaranteed the access to information mentioned above whereas the customer has access to only his/her information (limited access). Similarly, customers would also be benefitted as he/she can access this software

from anywhere, as to access it the only requirement is having an internet connection.

Definitions :

EMS-Event Management System
SRS-Software Requirement Specification
Stack Holder-The person who will participate in the system.
(Admin, Employee ,Customer)

Overview :

This EMS Established the link between all the firms of the system which are part of that specific event and to assign the works/tasks as well as provide all the information of their work to all Employees and getting work updates from them to record the progress of the event.

An admin has all permission to check all the event details as well as assign the manager and employees to specific events and maintain or manage all events easily from anywhere.

And provide the all information and progress of the event to the customer with the user-friendly interface and booking of event required services according to his/her choice.

Overall Description :

The EMS application allows the customer to book the required Services according to there requirements and after that admin assign the group of employees according to the event and book media, catering etc. for the events according to the customer booking manager manage and assign task to the employees and update the progress report and notify all the progress updates to admin and customer the whole application is online so that customer can see his/her event progress from anywhere anytime.

The main goal is to provide all the event related facilities to the customer in one application.

Functional Requirement :

1. Add/Delete Client Orders:
   The system provides this facility to the manager and customer both. Customer can add his/her order whereas manager can add, view and delete customers' orders.

2. Create/Edit/Delete Client Account:
   Customer can create and edit his/her account details, whereas the manager can delete customers' account, along with editing and viewing it.

3. View Services:
   Customer has an option to view the list of services offered by the firm.

4. Cancel Bookings:
   Manager can cancel any customer's booking.

5. Data Entry to Database:
   All of the data inserted into the system is stored into a database in the backend. The database that we used for this project is "MySQL".

6. Customized Catering Menu:
   Customer is able to create a custom menu depending on his/her preferences.

7. Customized Media Services:
   Customers are able to customize their media requirements.

8. Generate Event ID:
   A unique id is allocated to each event in order to access it easily.

9. Generate Customer ID:
   Each customer will be assigned a unique id associated with his/her email address.

10. Generate Employee ID:
    Each employee will be assigned a unique id associated with his/her email address.

11. Generate Catering ID:
    A unique id is associated with each caterer (vendor) to access their information from the database.

12. Generate Media ID:
    A unique id is associated with each media requirement in the database for ease of access.

13. Generate Menu ID:
    A unique id is associated with each menu in the database for ease of access.

14. Generate Studio ID:
    A unique id is associated with each studio in the database for ease of access.

15. Generate Venue ID:
    A unique id is associated with each venue in the database for ease of access.

16. Generate Total Bill:
    In the end a total bill will be computed from all the choices selected by the customer inclusive of taxes and this bill will be displayed on the screen which the user can print.

17. Enter Payment Details:
    Customer will be asked to enter payment details after selecting payment option.

18. Order Confirmation:
    This screen will be displayed showing that the order has been confirmed and awaiting approval.

19. Book Event:
    Customers can book events as per his/her requirements.

20. Add/Remove Employee:
    Manager will have the facility to add or remove an employee from the database.

21. Approve/Deny Order:
    All the confirmed orders will need to be approved by the manager.

22. Add/Remove Vendors:
    Manager has the facility to add or remove a vendor (caterer or studio).

23. View Orders:
    Customer is able to view his order. Moreover, manager can view a particular order and he/she can view all orders too.

24. Client Sign In:
    When a customer signs in, his/her credentials are verified from the database and then allowed access into the system.

25. Error Alarms:
    Whenever there is an error, a notification beep will be played to gain the user's attention followed by an error popup.

26. Sign Out:
    When a user clicks “Sign Out” button, they are logged out of the system and the flow returns back to the main menu screen

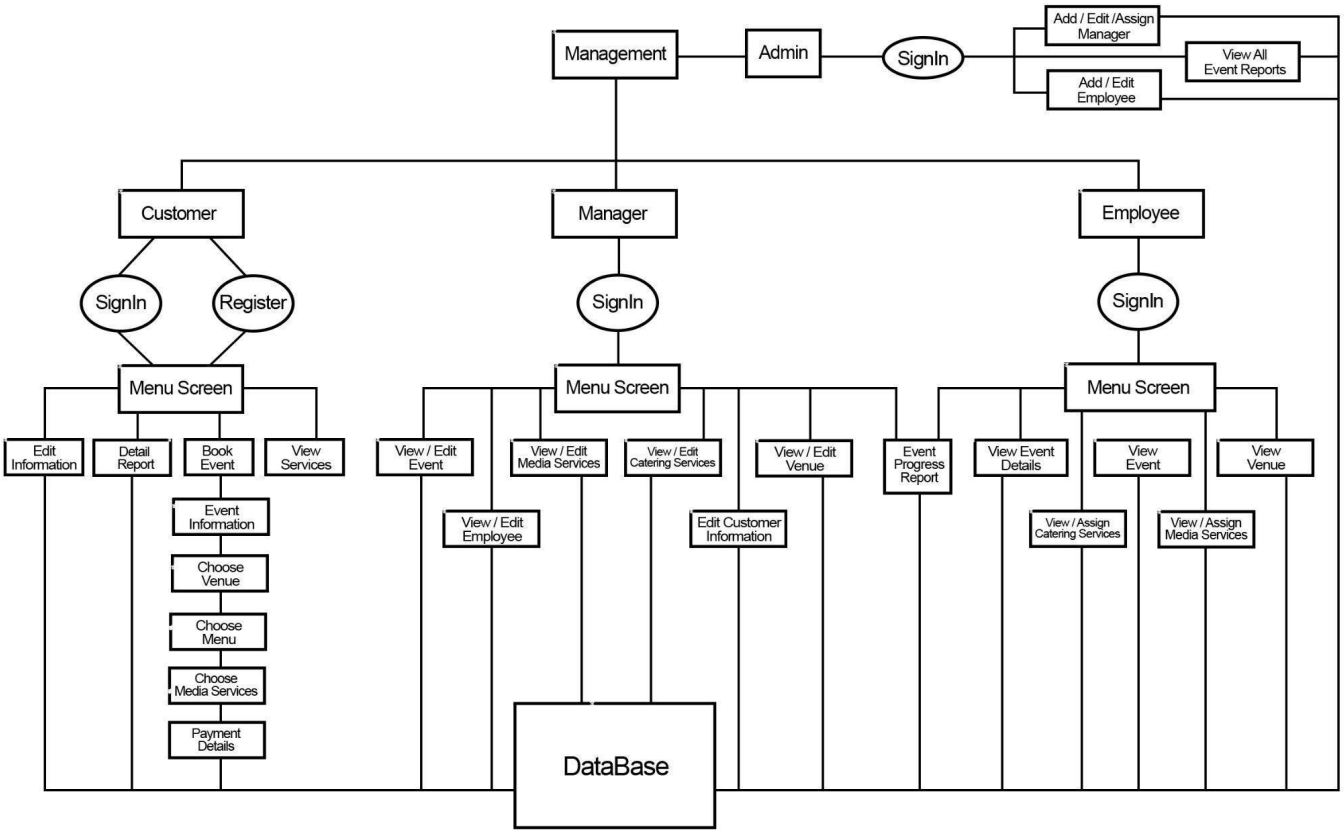
Non-Functional Requirement :

- Minimal Interface:
  Interface should be simple and easy for users to adapt to.
- Speed:
  System should be fast and responsive, with as minimal lag as possible.
- Portability:
  There should be abstraction between the application logic and the system interface – system should be modular
- Reliability:
  System should be stable, not prone to crashes. User should expect to have a consistent experience every time they use it.
- Scalability:
  System should be extendable if required, and adapt to greater number of users. It should be open for more functionalities to be added if required.
- Security:
  User data and/or other private details must be kept confidential and only visible to those authorized to view them.
- Data Integrity:
  Data stored should not be compromised, and should be accurate and reliable.

List of Features :

Employee Management
Account Management
Events booking management
Media Managements
Vendors Management
Payment management
Event Progress report

Flow Diagram:



ER listing :

CUSTOMER-
- Customer ID
- Full Name
- Aadhar Number
- DOB
- Contact Number
- Email
- Priority Status

CUSTOMERPASS -
- Customer ID
- Password

EMPLOYEE-
- Employee ID
- Employee Name
- DOB
- Email
- Contact Number
- Aadhar Number
- Account Number
- Salary
- Position
- Points
- Manager ID

EMPLOYEEPASS -
- Employee ID
- Password

EMPLOYEETODO -
- ToDo List
- Status
- Employee ID

EVENT-
- Event ID
- Event Name
- Event Type
- Event Date
- Guests Count
- Total Cost

Starting Time
Ending Time
Customer ID
Venue ID
Studio ID
Menu ID
Catering ID
Media ID
Status
Progress

MEDIA_REQUIREMENTS -
Media ID
Photography
Videography
Album
Drone
Crane

CATERING -
Catering ID
Caterers Name
Contact Number
Specialty
Days
Charges

MENU -
Product ID
Category
Subcategory
Menu Name
Price

BOOKED CATERING MENU -
Menu ID
Product Id

STUDIO -
Studio ID
Studio Name
Contact Info
Cost

VENUE-

- Venue ID
- Venue Name
- Location
- Address
- Max Capacity
- Description
- Category
- Contact Info
- Cost

PAYMENTS -

- Payment ID
- Amount
- Date and Time of Payment
- Payment Status

Application Architecture :

Application = Logic + data
Logic =(UI Logic + Business Logic + DataAccess Logic)
Data =(Structured data , Non Structured data)

Online Application :

- Web based Application: deployed on web and accessed by users from anywhere.

List of tables:

CUSTOMER-
(cust_id, name,aadhar_no, dob, phone_no,
email,account_number, priority_status)

CUSTOMERPASS -
(cust_id, password)
cust_id references cust_id in CUSTOMER

EMPLOYEE-
(emp_id, name, dob, email, phone_no, aadhar_no,
account_number,salary,position, points, mgr_id)
• mgr_id references emp_id in EMPLOYEE

EMPLOYEEPASS -
(emp_id, password)
• emp_id references emp_id in EMPLOYEE

EMPLOYEETOD -
(todo, status, emp_id)
• emp_id references emp_id in EMPLOYEE

EVENT-
(event_id, name, type, event_date, guests, total_cost,
starting_time, ending_time, cust_id, venue_id, studio_id,
menu_id, catering_id, media_id, status, progress)
• media_id references media_id from MEDIA_REQUIREMENTS
• studio_id references studio_id from STUDIO
• catering_id references catering_id from CATERING
• venue_id references from venue_id from VENUE
• menu_id references menu_id from MENU
• cust_id references cust_id from CUSTOMER

MEDIA_REQUIREMENTS -
(media_id, photography, videography, album, drone, crane)

CATERING -
(catering_id, name, contact, specialty, days, charges)

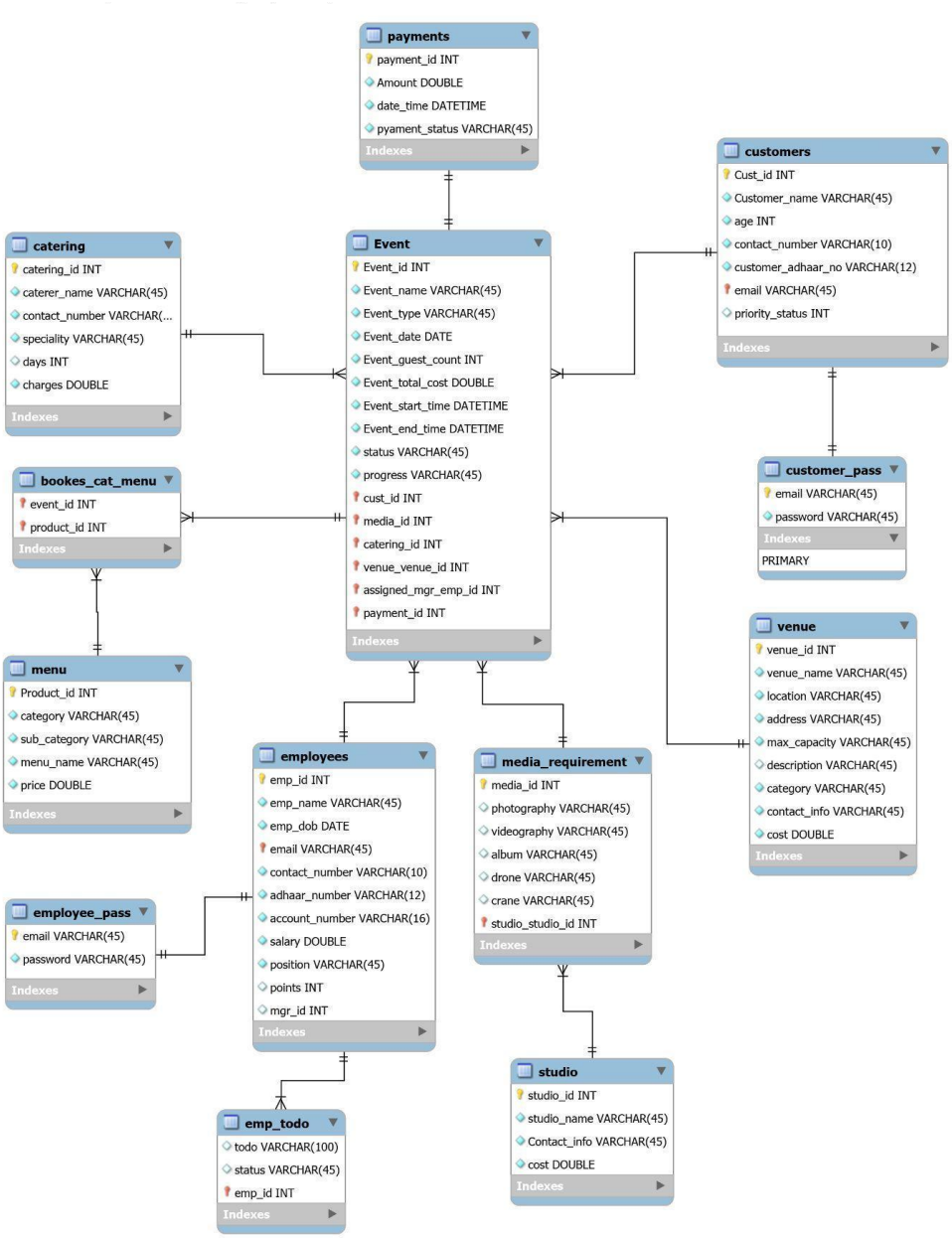
MENU -
(product_id, category, subcat, menu_name, price)

BOOKED CATERING MENU -
(menu_id,product_id)

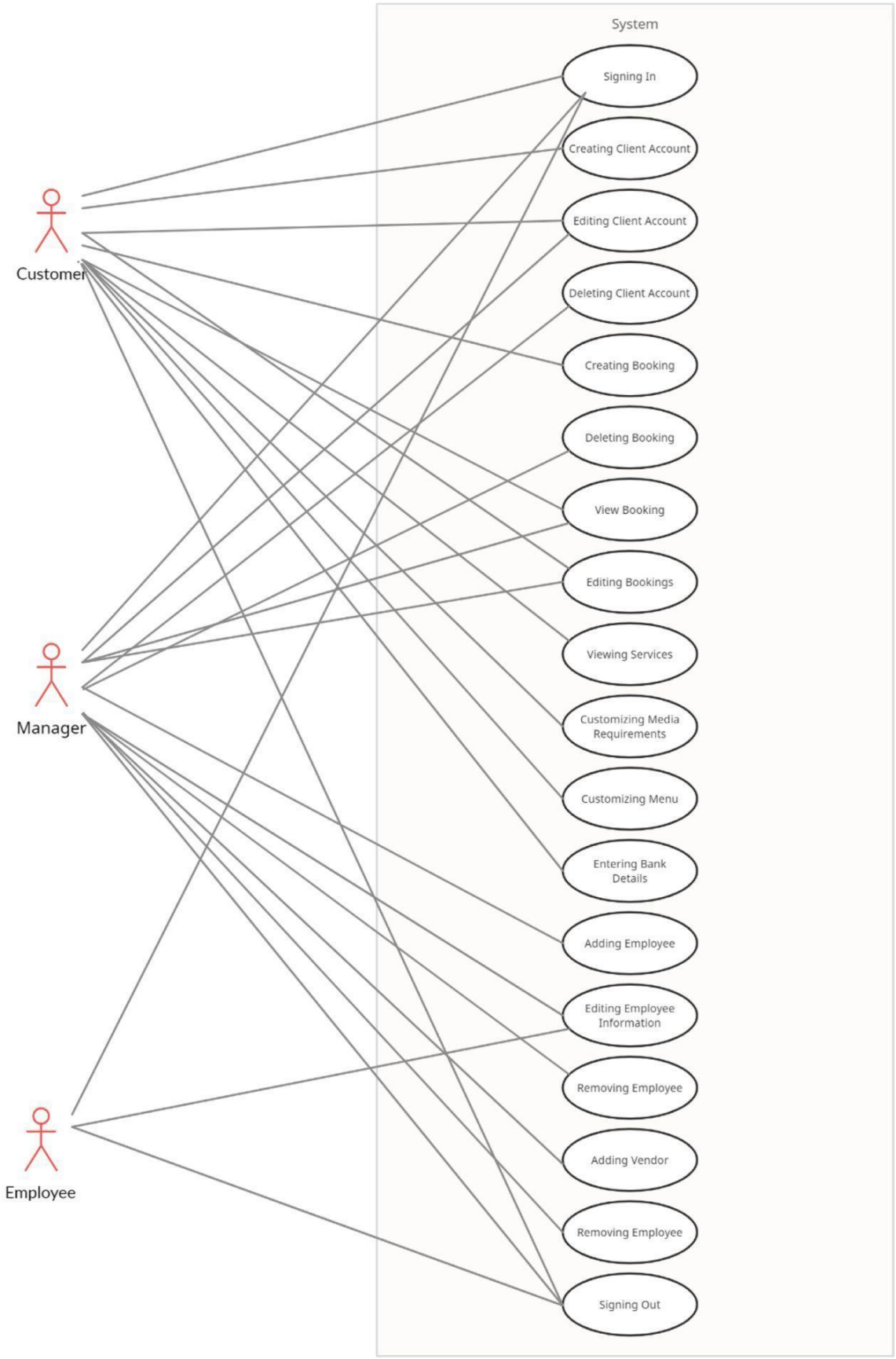
STUDIO -
(studio_id, name, contact_info, cost)
VENUE-
(venue_id, name, location, address, max_capacity,
description, category, contact_info, cost)

PAYMENTS -
(payment_id, amount, date_time, payment_status)

Entity Relationship Diagram:



Use Case Diagram :



Class Diagram :