**Centre for Development of Advanced Computing, Bangalore, India**

---

# Design Document

## (As a part of ACTS Management System)

# GET YOUR ARTIST


**Project Guide:**

Mr. Arpit Dubey

**Prepared By:**

Abhishek Parashar (PRN- 240350120006)

Akanksha Bharne (PRN- 240350120011)

Aditi Sali (PRN- 240350120130)

Vaibhav Kakad (PRN- 240350120169)

# Contents:

# 1.0 Introduction:

## 1.1 Purpose:

The GetYourArtist program aims to provide a complete web platform to facilitate booking artists for both private and public events. By making it easier for users to book artists interactively, the website seeks to help participants. It makes it easier for event planners to identify and select the right talent for their event and allows them to manage the issues. With feature rich solutions for creating, editing, managing and canceling event bookings, the platform improves event administration and assures that event bookings will be efficiently managed to meet the requirements of individuals and artists organizing by providing customized options for booking needs. The main goal of GetYourArtist is to enhance the user experience by providing a simple, intuitive platform that makes artist and event management easy for every user.

## 1.2 Scope:

Artist Booking System aims to develop a comprehensive platform that simplifies the process of booking artists for various events. System will address current industry challenges by providing automated and centralized solutions for managing inquiries and bookings. Basically System will create and maintain detailed artist profiles, including bio, performance history, availability, and contact information. Allow event organizers to search for artists based on various criteria such as genre, location, availability, and budget. Facilitate the booking process from initial inquiry to final confirmation, including negotiation, contract generation, and digital signing. Manage artists' calendars to prevent double bookings and ensure availability is accurately represented. Implement role-based access control to ensure users have appropriate permissions based on their roles (artists,agents, organizers, admin).

## 1.3 Intended Audience:

- **Event organizers:** the people and teams responsible for organizing and organizing the open event; Artists will be booked and the platform will be used to manage the technical aspects of the event. Particular attention should be given to the aspects describing the features of the event management system, the booking process, and the user interface.
- **Private Event Planners:** People looking to hire actors for personal events, weddings, or other events. They should focus on the platform’s user manual, hire artists and manage bookings.
- **Artists:** Entrepreneurs looking to document events by creating and maintaining accounts on the platform. They should go through sections on setting up and maintaining profiles and tips on how to effectively implement prospect exposure and registration.

## 1.4 References


- Reference: https://www.academia.edu/38112670/SRS_Event_Management_System research paper from: hitesh Shrivastava.
- Reference: https://www.academia.edu/38112670/SRS_Event_Management_System research paper from: hitesh Shrivastava.
- Project on DJ Booking Management System: https://www.studocu.com/in/document/srm-institute-of-science-and-technology/bechlor-of-science/odjms-project-report-hiiii/53485120.
- Review Paper on an Event Management System. Amir Saleem1, Davood Ahmed Bhat2, Mr. Omar Farooq Khan3, Students of SSM College of Engineering and Technology, University of Kashmir India. Assistant Professor of SSM College of Engineering and Technology, University of Kashmir, India.
- Project report and research on Online Event Management System Deepanshu Goyala , Arbab Alib , Md Nafis Haiderc.

## 2.0 Acronyms:

| Sr.No | Acronyms | Stands For |
|---|---|---|
| 1 | Admin | Administrators,Administrative |
| 2 | APIs | Application Programming Interface |
| 3 | Auth | Authentication |
| 4 | CCPA | Central Consumer Protection Authority |
| 5 | DB | Database |
| 6 | DBMS | Database Management System |
| 7 | ERD | Entity-Relational Diagram |
| 8 | FR | Functional Requirement |
| 9 | FRS | Functional Requirement Specifications |
| 10 | GOVT | Government |
| 11 | GDPR | General Data Protection Regulation |
| 12 | HTTPs | Hypertext Transfer Protocol Secure |
| 13 | PAN | Permanent Account Number |
| 14 | SRS | Software Requirements Specification |
| 15 | SQL | Structured Query Language |

# 3.0 Assumptions and constraints:

## Assumptions:

**User Access and Internet Connectivity:**

- Users will have a stable internet connection to access the platform.Users will access the platform using the latest versions of web browsers (like Chrome, Firefox, Safari).
- Users will have a stable internet connection to access the platform.
- Users will access the platform using modern web browsers (e.g., Chrome, Firefox, or Safari).

**Technology Stack:**

- The chosen technology stack will meet all the requirements of the core functionalities.
- The seamless operation of other third-party services regarding booking and registration will be at hand.

**Security Measures:**

- Normal security protocols such (like encrypted connections (HTTPS) and secure password) as password storing, and OAuth2 for authentication.
- The passwords should be long and non-authentic, what is worse.

**Regulatory Compliance:**

- The platform is compliant with the data protection laws of the countries that it operates. The company's policy to data usage goes only with the recorded consents of the users and the partnership to the artists is through consent as well.
- The compliance of data to the GDPR and CCPA allows our company to operate reachable with the protection of data.

## Constraints:

**Data Privacy:**

- Strict protection measures are needed when storing and transmitting sensitive data, like the AADHAAR and PAN card info.

**User Authentication and Authorization:**

- The security of user authentication and authorization must be ensured through secure methods alone.
- ARABIC is one mechanism that should be put into place to restrict the access of some features.

**Maintenance and Updates:**

- You cannot evade the fact that security and performance would have to be overlooked if you do not practice regular maintenance exercises.
- Updates have to be properly managed so neither the user experience is destroyed nor the features be allowed to be used.

# 4.0 Basic Design Approach:

## Pages:

- **User Login - Sign In/Up:**

  Users can create new accounts or log in to their current ones using this page. Email addresses or usernames and passwords are included. Enter the necessary information, such as your password, email address, and name.

- **Home:**

  The platform's main landing page is the home page. It offers a summary of the features of the platform, together with the latest blog entries, upcoming events, and popular artist highlights. In order to make sure users can quickly obtain the information they require, it also offers navigation connections to other sections of the website.

- **About Us:**

  This joined page provides details about the GetYourArtist platform, including its goals, the people that created it, and their mission. Users can also utilize the contact form to send questions, feedback, or requests for help to the support staff. There may be FAQs and thorough instructions on how to use the platform on this page as well.

- **Get Your Artist:**

  The goal of this page is to assist users in finding and exploring artists. Artists can be found and filtered by users according to several factors, including availability, location, genre, and more. Every artist profile has comprehensive details, past performances, availability, and contact details.

- **Book Artist:**

  Users can proceed with reserving an artist for their event on this page. It leads consumers through every step of the reservation process, from choosing dates to verifying availability, settling conditions and completing the reservation.

- **Register:**

  Artists can sign up and set up their platform profiles on this page. It has sections for availability, performance history, professional background, and personal information. The registration page makes sure that artists fill out all the forms required in order to be hired and displayed on the platform.

- **Events Page:**

  A schedule of upcoming events with artists from the platform is shown on the events page. The date, venue, and artist lineup are among the details that users can view about the event. Users may also be able to attend the event or indicate their interest on the page.

- **Feedback:**

  Users can leave feedback on this page regarding their interactions with artists, the platform as a whole, and the booking procedure itself. It has a form where people may write comments and review their experience. Feedback guarantees a high degree of service quality and aids in platform improvement.

## Modules:

1. **Login and Profile Management:**
   - **Description:** Handles user authentication, including sign-in, sign-up, and user profile management.
   - **Features:**
     - User Sign-In:
       1. Email ID
       2. Password
     - New User Sign-Up:
       1. First Name
       2. Last Name
       3. Email
       4. Password
2. **Artist Booking and Registration:**
   - **Description:** Manages the process of registering artists on the platform and allows users to book these artists for events.
   - **Features:**
     - Artist Registration Form
     - Artist Profile Management
     - Artist Booking Form
     - Artist Booking Status Tracking
3. **Events Management:**
   - **Description:** Allows users to add their events and display those events where artists are booked, including upcoming and future events.
   - **Features:**
     - Event Creation and Management
     - Viewing Upcoming and Future Events
4. **Blogs:**
   - **Description:** Facilitates the creation, management, and viewing of blog content by users and admin.
   - **Features:**
     - Blog Creation and Viewing
     - Blog Approval and Moderation by Admin
5. **Admin:**
   - **Description:** Provides an interface for the admin to manage users, artists, bookings, and other platform activities.

- **Features:**
  - User Management
  - Artist Management
  - Booking Management
  - Event Management
  - Blog Approval/Reject

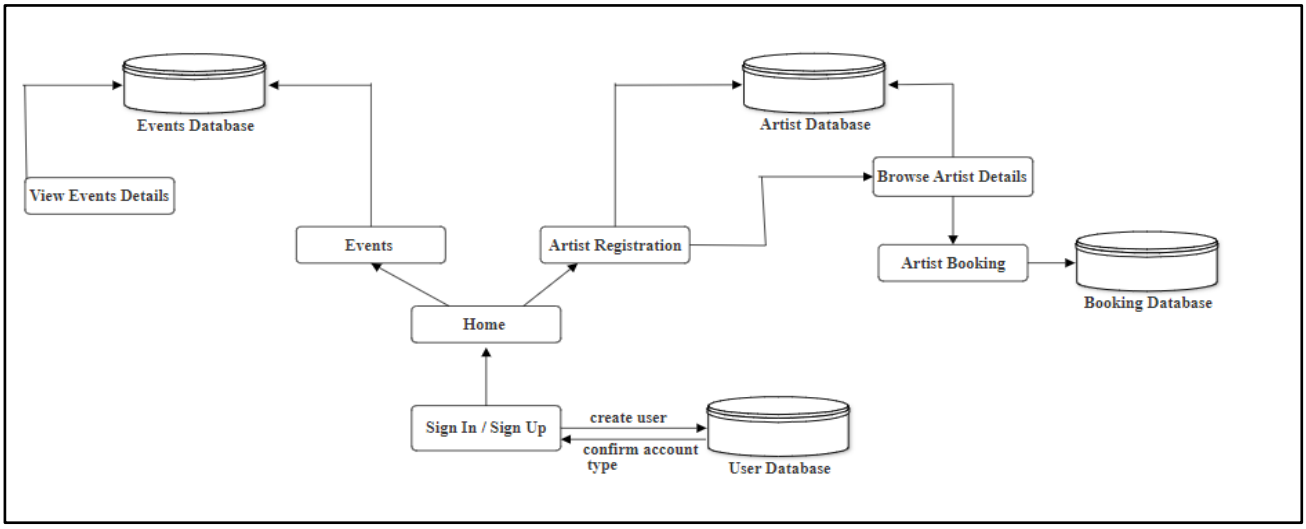


*Figure 4.1:System Flow Chart*

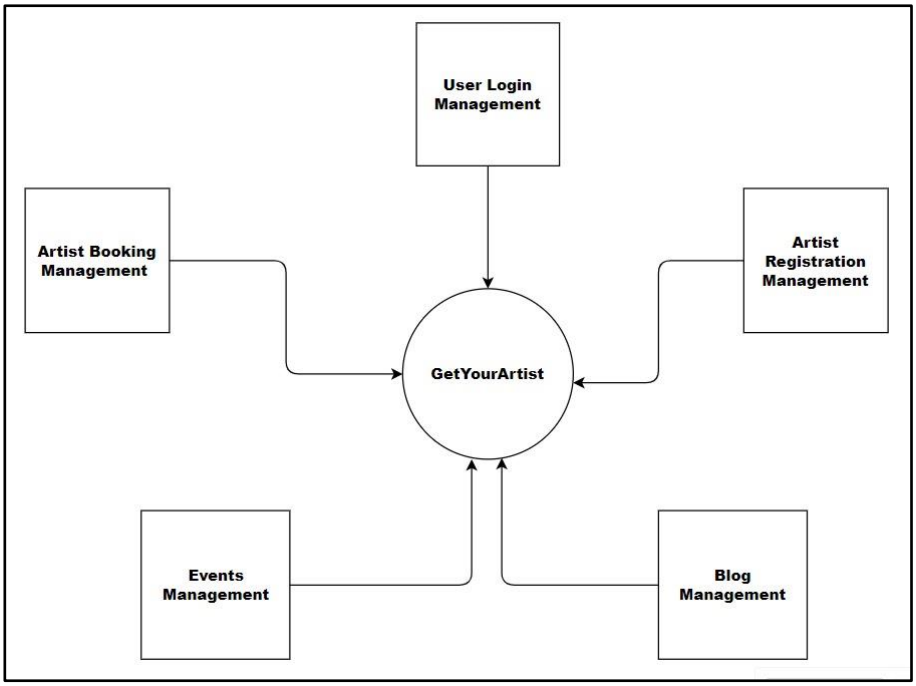


*Figure 4.2: GetYourArtist Data Flow Diagram*

# 5.0 Risks:

- **Complex State Management:**
  - As the system deals with artist registrations, bookings, approvals, and feedback, maintaining and managing the application state, especially in a large-scale platform, can become complex.
- **User Authentication and Authorization:**
  - Implementing a secure and efficient user authentication and authorization system to ensure that only authorized personnel can register, book, or manage artists.
- **Real-time Updates and Notifications:**
  - Achieving real-time functionality for artist booking updates, approvals, and feedback notifications can be challenging, especially when dealing with multiple users concurrently interacting with the system.
- **Ensuring Cross-Browser Compatibility:**
  - Making sure that the web application functions seamlessly across various browsers and devices.
- **Error Handling and Logging:**
  - Handling errors gracefully and implementing effective logging to diagnose issues during development and in a live environment
- **Maintaining Codebase Scalability:**
  - Ensuring that the codebase remains scalable as the application grows, adding new features or accommodating changes in business requirements.

# 6.0 System Overview:

GetYourArtist is a web-based event booking platform that allows users to conveniently book artists for any type of occasion. The portal covers the full spectrum that is from musicians to speakers to performers and makes the planning and handling of events almost seamless for the end users. This report gives an overview of the system design, components, and operations of the GetYourArtist project.

The GetYourArtist system consists of the following main features: user registration and management, artist registration and management, searching and booking, event management, reviews and ratings, notification system, data security, and a blog page service.

The user registration and management module is where the users can register themselves by creating an account. Upon registration, they can log in to the platform to enjoy its functionality. They have the facility to manage their profiles, on the one hand, by editing their personal information and on the other hand by adding the type of events they are interested in. The artist registration and management module lets artists join the site, and then they can log in to access the back end to update their profile and availability. Artists have an opportunity to introduce themselves and refresh the information displayed in their profiles, thus showing their work and availability.

In the search and booking feature, the users can browse the artists using different kinds of methods such as places, the availability of performers, type of performances, and the satisfaction of customers. They can input the respective details and the request for the service. The artists might either reply positively or negatively. In case of the latter, there will be notifications sent. In this way, the booking will be made, and the notifications will be received by both parties.

The event management module comprises the option for users to manage booking events by creating, changing, viewing, and canceling bookings. The use of the reviews and ratings option allows the users to rate the artists' performance and leave the reviews. The user notification system makes sure that the users get notifications for booking requests, confirmations as well as updates, and the artists get the notifications about the new bookings and changes.

GetYourArtist is a reliable platform that ensures the security of data. Login and authentication are very secure methods developed to ensure; nobody can break in and users have not only secure access to the login, but also the use of data encryption protects them from hackers and viruses. Blog page is the unique place where the viewers are allowed to either write about the passing of time or the expression of the creative side of their souls. The GetYourArtist goal is the overall provision of the latest and high-quality event organization tools and the connecting of artists to these events along with the super features and security measures to help users and artists feel safe and comfortable respectively.

Your comments on the points mentioned are welcome or you may also suggest alternative methods of improving them. Develop brief reports on the topics listed above: events and artists that you have participated in, talk with artists and share your experiences with them, etc. The overall idea of GetYourArtist is to give everyone proper and efficient equipment along with easily going and secure features for the event organizers as well as the artists involve

# 7.0 Architecture Design

- **Frontend**
  - **Framework**: React.js
  - **Components**: User Login, Home, About Us, Artist Booking, Event Management, Blogs, Feedback
- **Backend**
  - **Framework**: Spring Boot
  - **Tools**: Postman for API testing
- **Database**
  - **Database**: MongoDB
  - **Collections**: Users, ArtistBookings, ArtistRegistrations, Feedbacks, Blog

## 7.1 Sequence Diagram

### 1. GetYourArtist Sequence Diagram

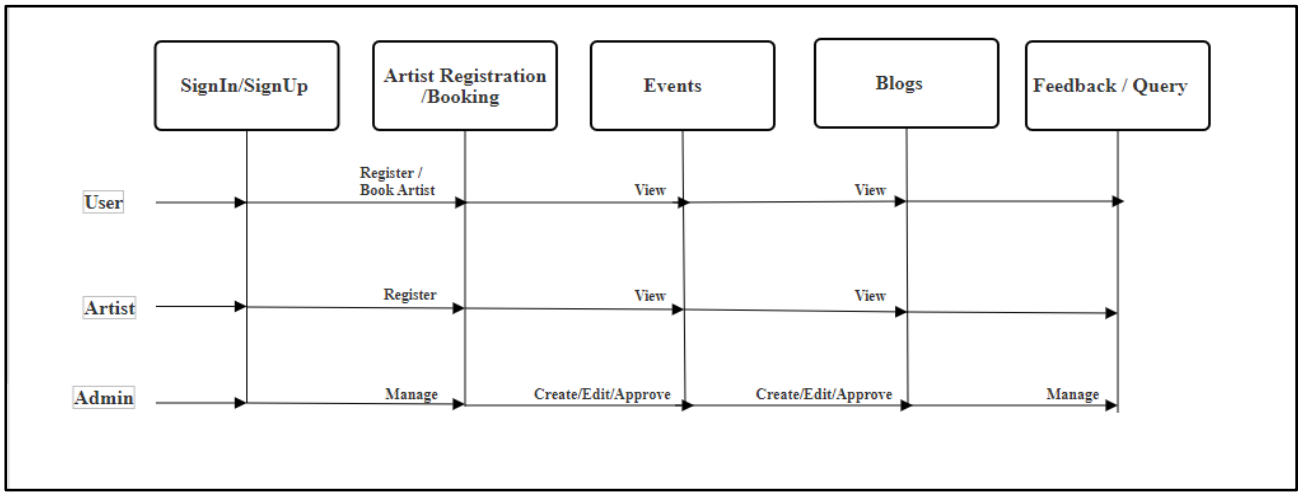


***Figure 7.1.1: GetYourArtist Sequence Diagram***

**2.Login Module**

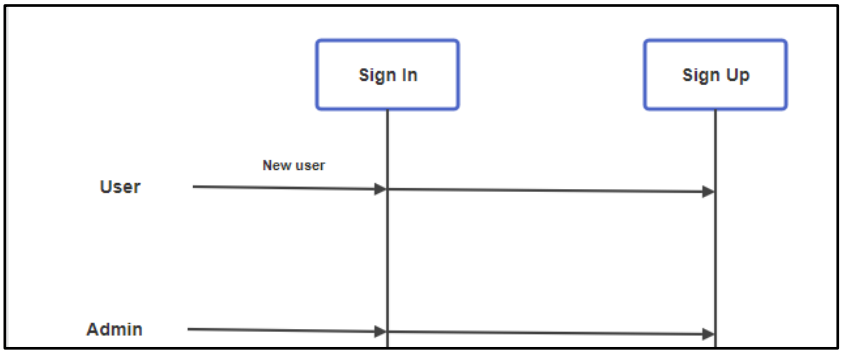


***Figure 7.1.2 : Login Module***

3. Artist Booking and Registration Module

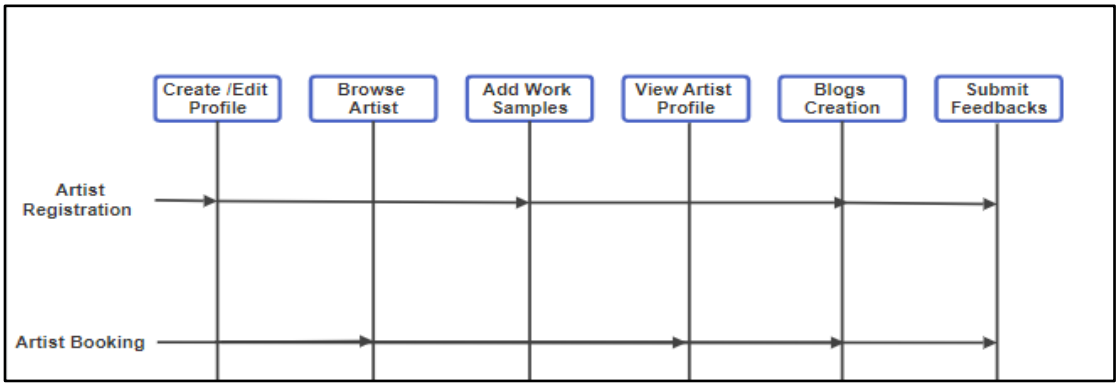


***Figure 7.1.3: Artist Booking and Registration Module***

4. Event Management Module

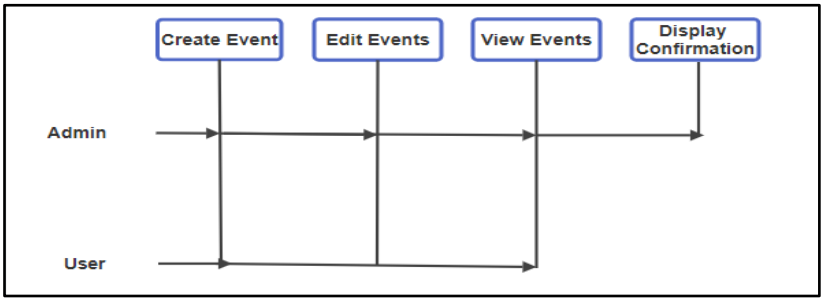


***Figure 7.1.4 : Artist Booking and Registration Module***

5. Admin Management Module

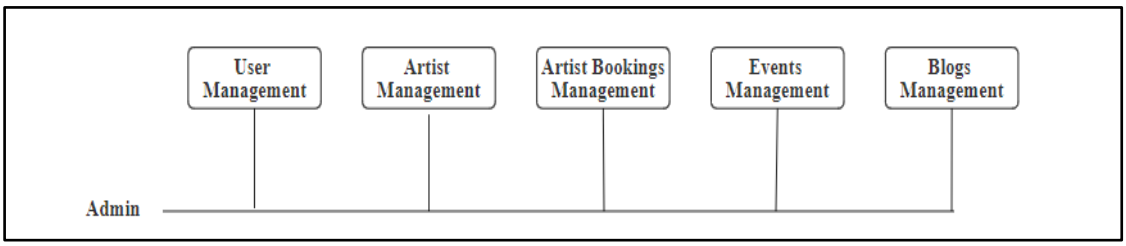


***Figure 7.1.5 : Artist Booking and Registration Module***

## 7.2 Use Case Diagrams

### 1. System Use Case Diagram

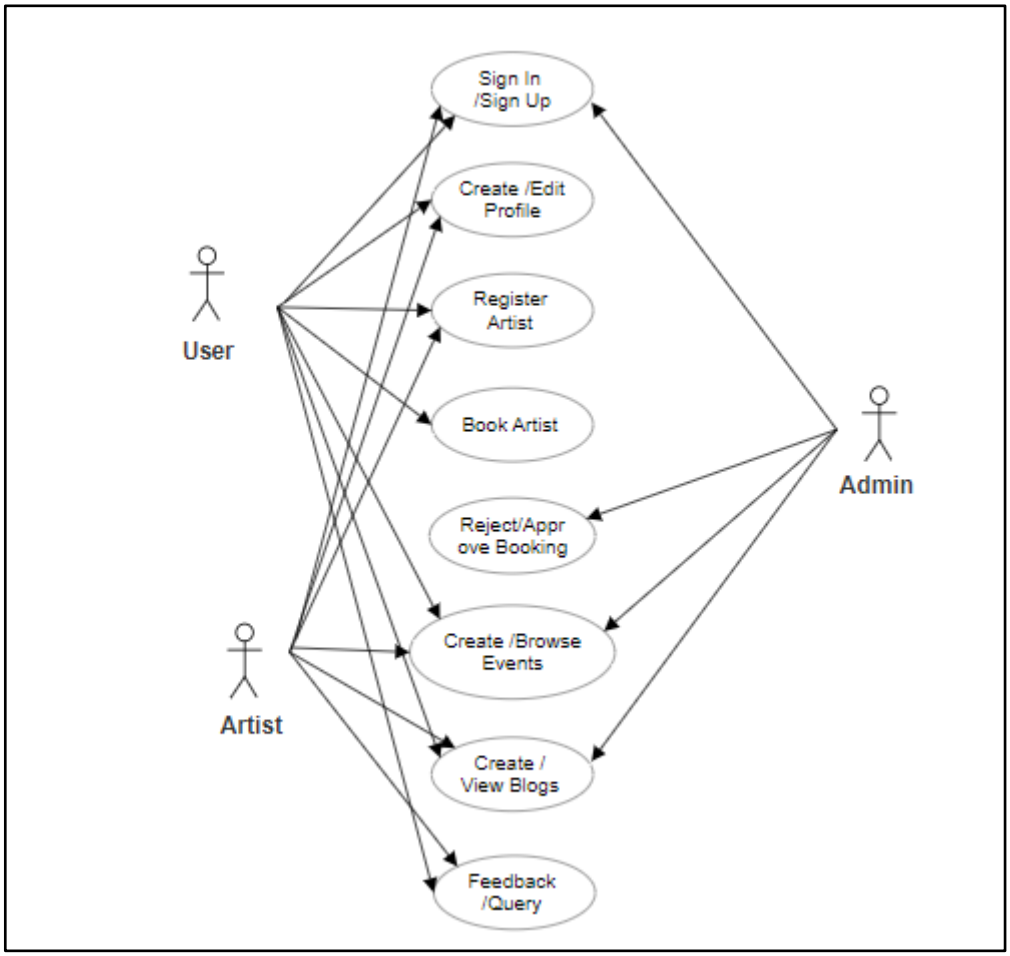


***Figure 7.2.1: System Use Case Diagram***

## 2. Login Module

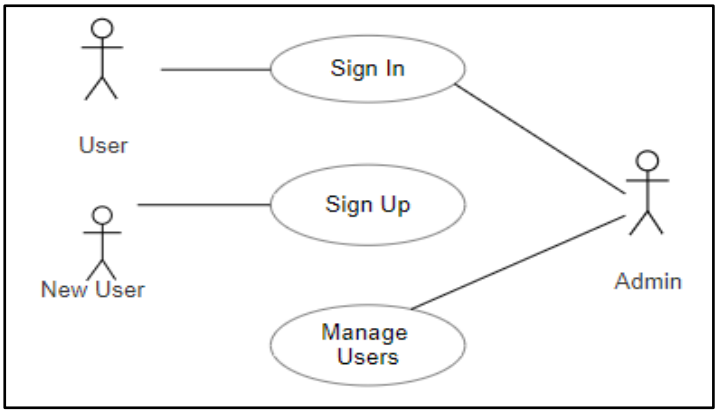


*Figure 7.2.2: Login Use Case Diagram*

## 3. Artist Booking and Management

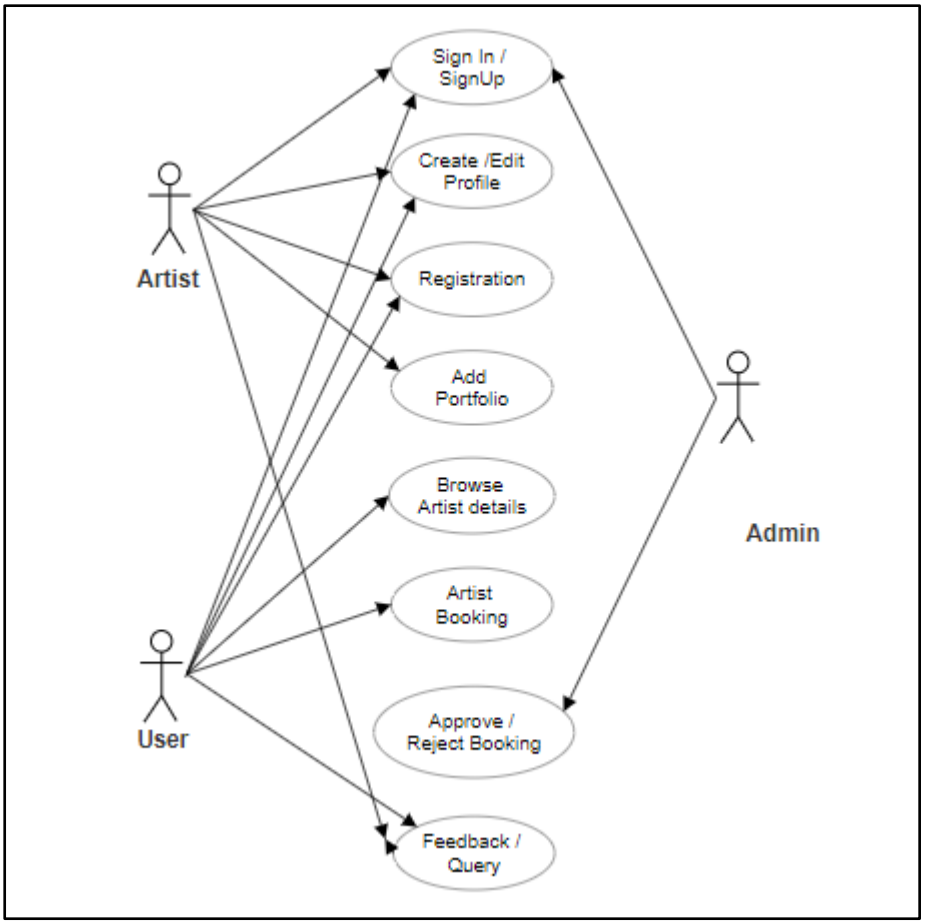


*Figure 7.2.3: Artist Booking and Management Use Case Diagram*

## 4. Events and Blogs Management

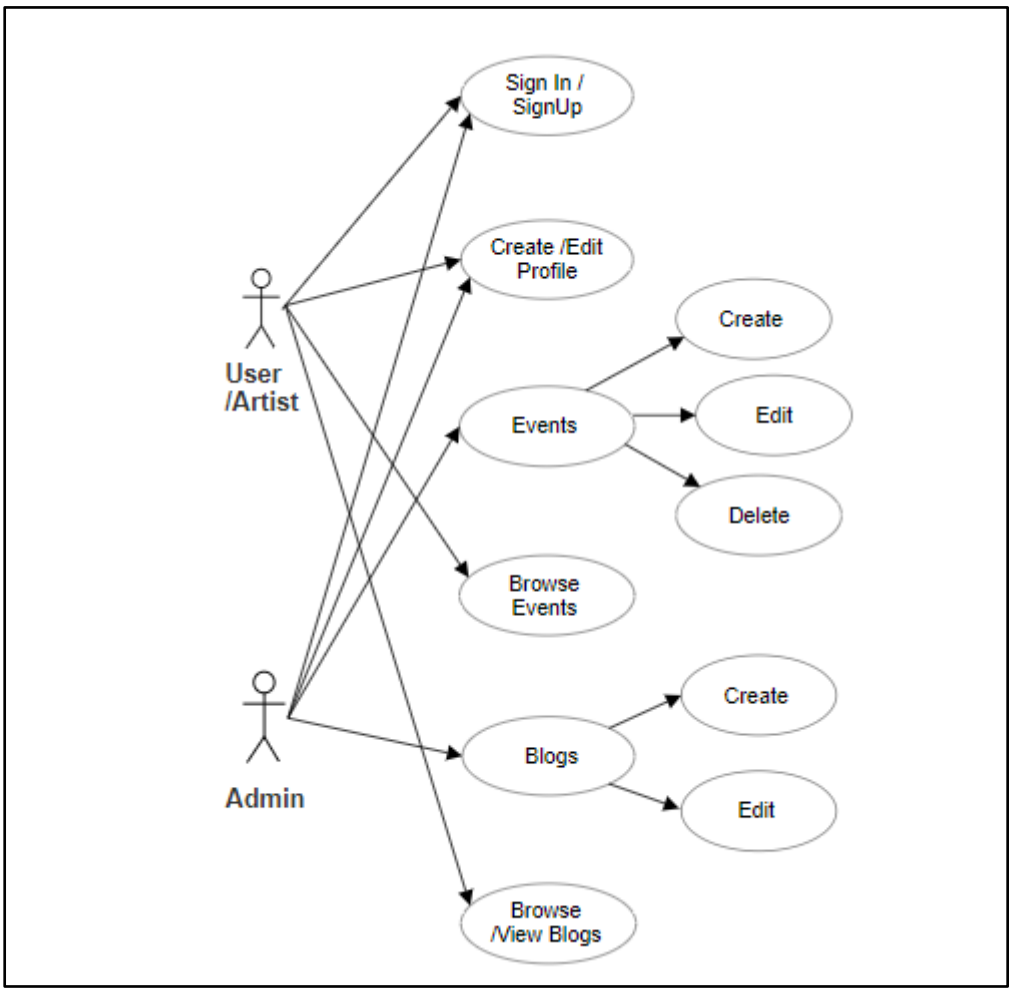


*Figure 7.2.4 : Events and Blogs Use Case Diagram*

## 5. Admin Management

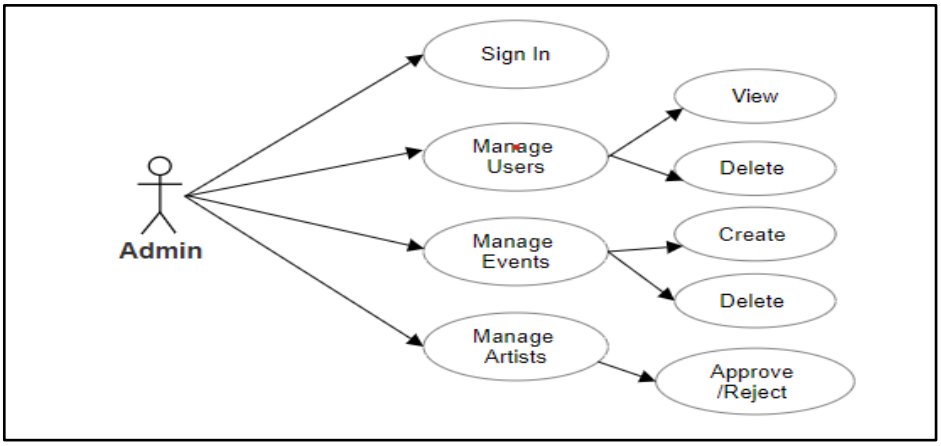


*Figure 7.2.5: Admin Use Case Diagram*

# 8.0 Data Design

The data structure of the GetYourArtist project focuses on the structure, relationships, and constraints of the data stored in the system. This section provides a detailed description of data entities, their attributes, and the relationships between them. Primary data types include user, artist registry, artist registry, comments, discussion history, and blog agencies.

**Table 01: User Entities**

The User entity stores information about the users who interact with the platform.

| Idx (Primary Key) | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | _id | Integer | A unique identifier for each user |

**Other users entities:**

| Idx | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | name | Varchar | The full name of the user. |
| 2 | email | Varchar | The email address of the user, used for login and notifications. |
| 3 | password | Varchar | The encrypted password for user authentication. |
| 4 | phone | Integer | The user's phone number. |
| 5 | aadhaar_card/ pan-card | Integer | The user's Aadhaar card number, with validation. |
| 6 | profile_image | image | A URL or file path to the user's profile image. |
| 7 | created_at | Date,Time | The timestamp when the user was created. |
| 8 | updated_at | Date,time | The timestamp when the user information was last updated. |

**Table 02: Artist Booking Entities**

The User entity stores information about the users who interact with the platform.

| Idx (Primary Key) | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | _id | Integer | A unique identifier for each user |
| **Idx (Reference)** | **Field Name** | **Data Type** | **Description** |
| 1 | user_id | Varchar | A reference to the User entity |
| 2 | artist_id | Varchar | A reference to the Artist Registration entity |

**Other Artist Booking Entities:**

| Idx | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | booking_date | Date | The date when the booking request was made. |
| 2 | event_date | Date | The date of the event for which the artist is booked. |
| 3 | status | Varchar | The current status of the booking (e.g., pending, confirmed, canceled). |
| 6 | profile_image | image | A URL or file path to the user's profile image. |

**Table 03: Artist Registration Entities**

The User entity stores information about the users who interact with the platform.

| Idx (Primary Key) | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | _id | Integer | A unique identifier for each user |

**Other Artist Registration Entities:**

| Idx | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | name | Varchar | The full name of the artist. |

| | | | |
|---|---|---|---|
| 2 | email | Varchar | The email address of the artist, used for login and notifications. |
| 3 | phone | Integer | The artist's phone number. |
| 4 | aadhaar_card/ pan-card | Integer | The artistAadhaar card number, with validation. |
| 5 | profile_image | Image | A URL or file path to the user's profile image. |

**Table 04: Feedback Entities**

The User entity stores information about the users who interact with the platform.

| Idx (Primary Key) | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | _id | Integer | A unique identifier for each user |
| **Idx (Reference)** | **Field Name** | **Data Type** | **Description** |
| 1 | user_id | Varchar | A reference to the User entity |
| 2 | artist_id | Varchar | A reference to the Artist Registration entity |

**Other Feedback Entities:**

| Idx | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | rating | Integer | The rating given by the user (e.g., 1 to 5 stars). |
| 2 | comments | Varchar | Additional comments or reviews provided by the user. |

**Table 04: Blogs Entities**

The User entity stores information about the users who interact with the platform.

| Idx (Primary Key) | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | _id | Integer | A unique identifier for each user |
| **Idx (Reference)** | **Field Name** | **Data Type** | **Description** |
| 1 | author_id | Varchar | A reference to the User entity |

**Other Blogs Entities:**

| Idx | Field Name | Data Type | Description |
|---|---|---|---|
| 1 | title | Varchar | The title of the blog post. |
| 2 | content | Varchar | The main content of the blog post. |
| 3 | images | Images (jpg,png,jpeg) | An array of URLs or file paths to images associated with the blog post. |
| 4 | created_at | Date,Time | The timestamp when the user was created. |
| 5 | updated_at | Date,time | The timestamp when the user information was last updated. |

## 8.1 ER Diagram

1. Artist Booking Management

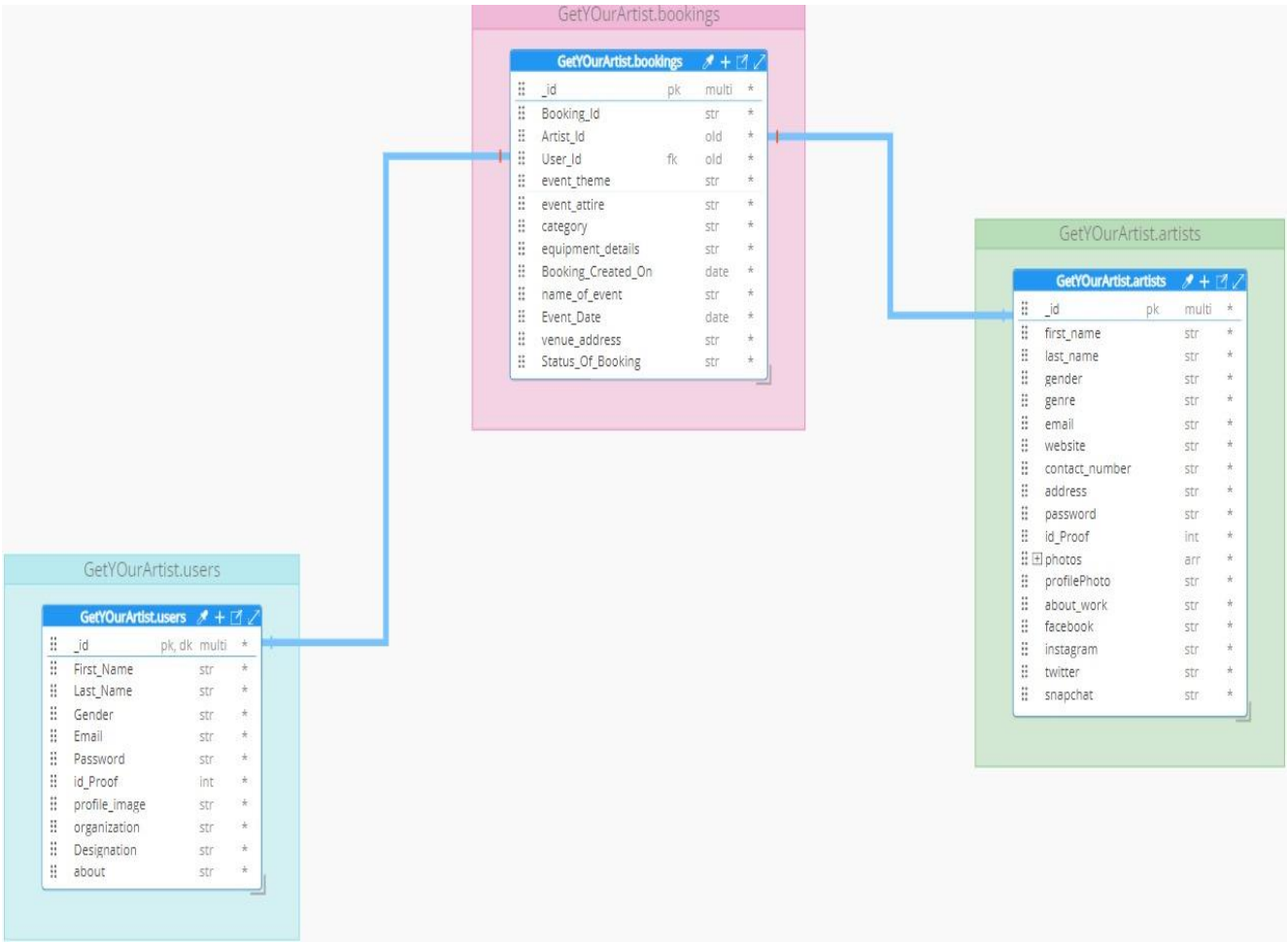


*Figure ERD1 : Artist Booking Management*

2. Blog Management

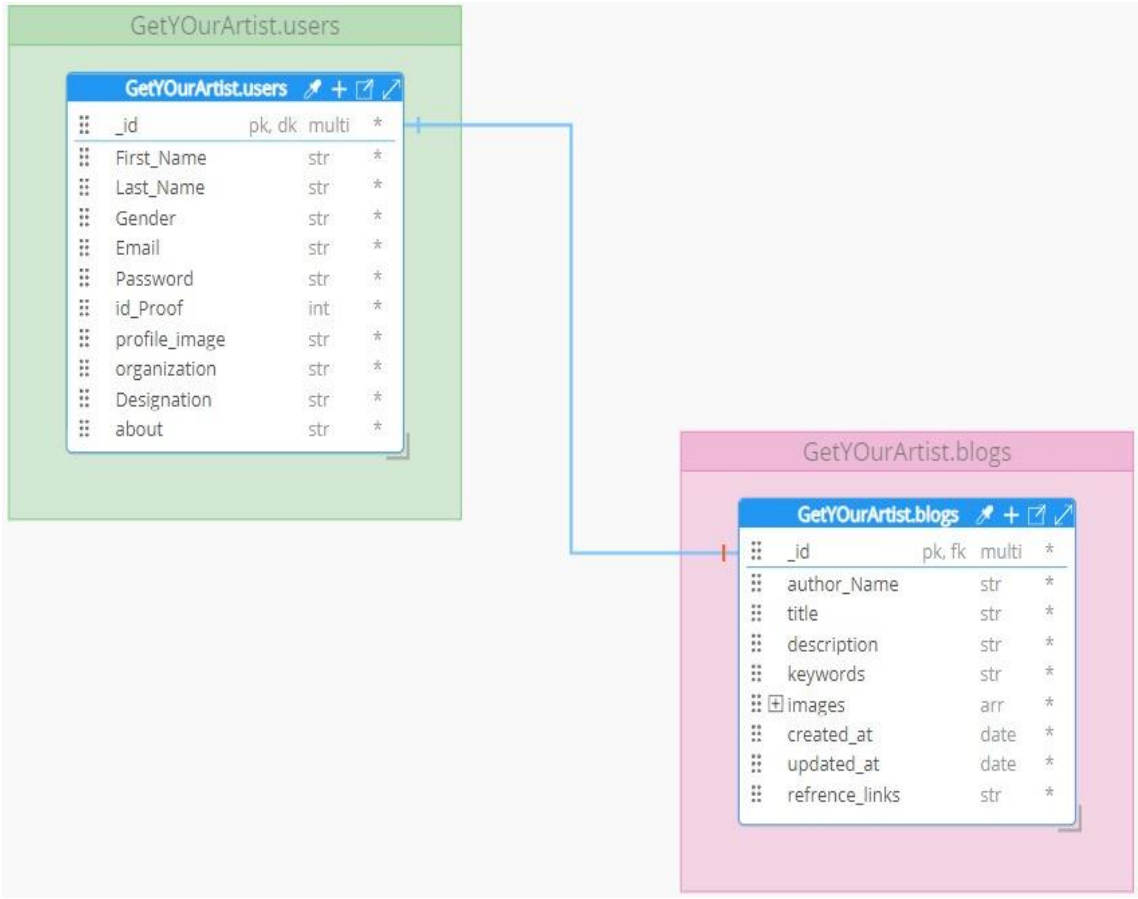


*Figure ERD 3 : Blogs Management*

3. Admin Management

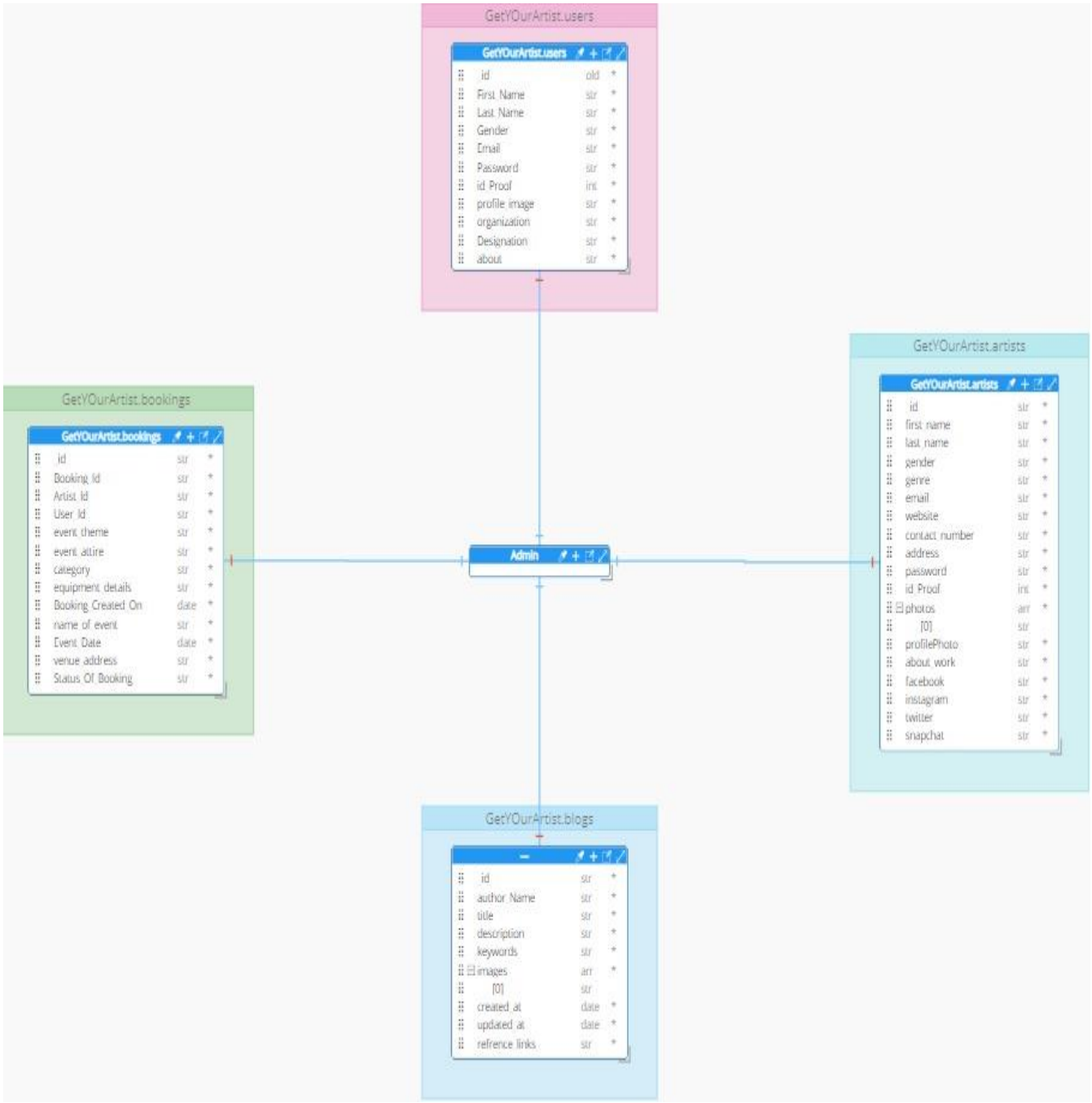


*Figure ERD 4 : Admin Management*

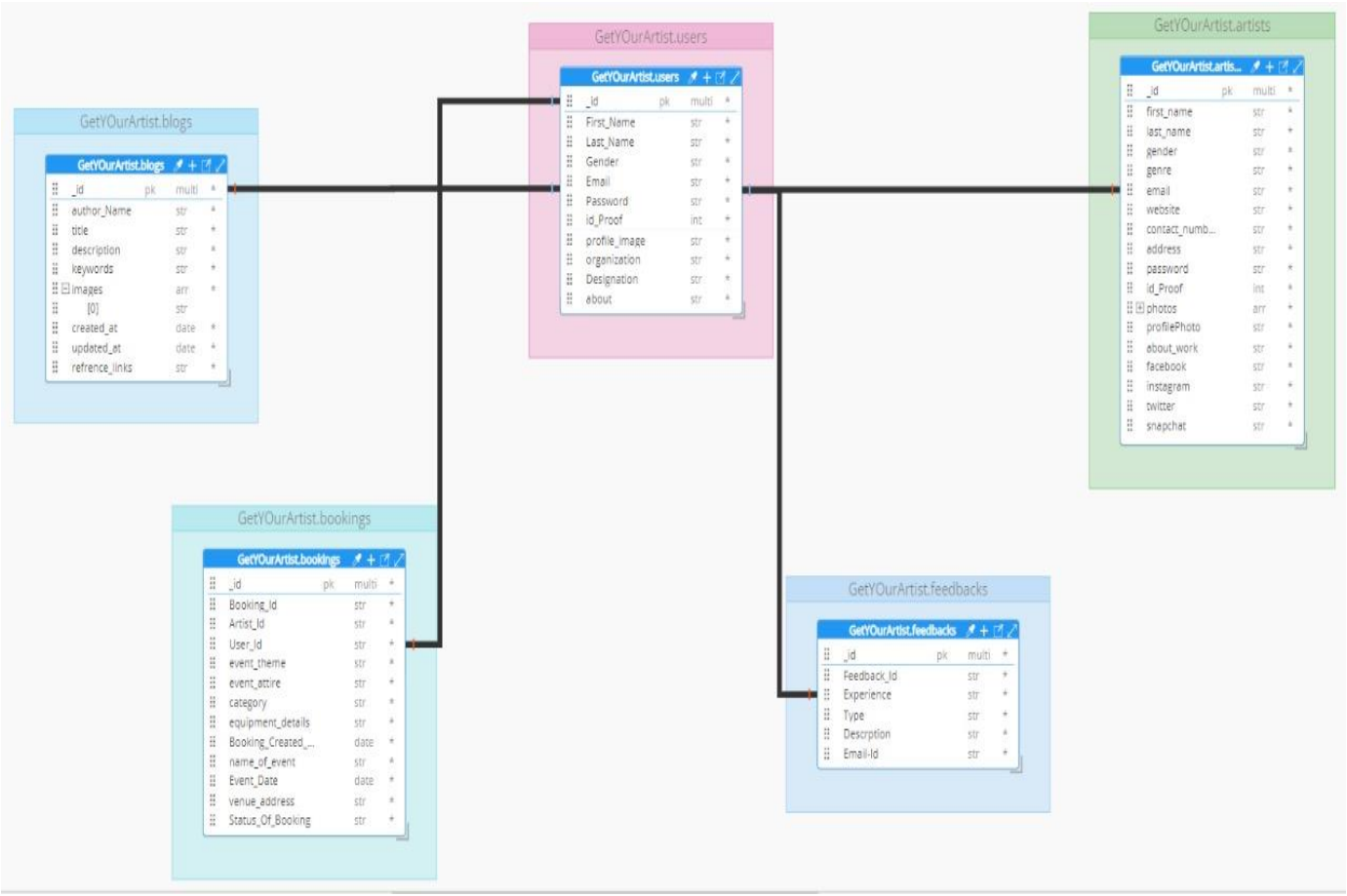


*Figure ERD 5: System Database ER Diagram*

# 9.0 Component Design

### 1. Frontend (React.js)

- **Description:** The frontend is built using React.js, which provides a dynamic and responsive user interface for the application.
- **Key Features:**
  - User-friendly navigation.
  - Responsive design for different devices.
  - Forms for user and artist registration.
  - Search functionality.
  - Booking and event management interfaces.
- **Process:**
  - Set up the React.js environment.
  - Create components for each page (Login, Sign Up, Home, About Us, Get Your Artist, etc.).
  - Integrate API calls to the backend.
  - Style components using CSS or styled-components.
  - Test the frontend for usability and responsiveness.

### 2. Backend (Spring Boot, Postman)

- **Description:** The backend is developed using Spring Boot, which handles business logic, API endpoints, and database interactions.
- **Key Features:**
  - RESTful API creation.
  - Business logic implementation.
  - Integration with MongoDB.
  - User and artist authentication.
- **Process:**
  - Set up the Spring Boot project.
  - Define API endpoints for each functionality (User Management, Artist Management, etc.).
  - Implement service layers for business logic.
  - Connect to MongoDB using Spring Data.
  - Test API endpoints using Postman.
  - Secure APIs with authentication and authorization mechanisms.

### 3. Database (MongoDB)

- **Description:** MongoDB is used as the primary database for storing user, artist, booking, and event data.
- **Key Features:**
  - Document-based storage.

- Flexible schema design.
- Scalability and high performance.

- **Process:**
  - Set up a MongoDB instance.
  - Define schemas for User, Artist, Booking, Feedback, Chat, and Blog collections.
  - Implement CRUD operations for each collection.
  - Ensure data validation and indexing for optimized performance.
  - Test database operations through backend services.

4. **User Management**

- **Description:** Handles user registration, login, and profile management.
- **Process:**
  - Create user registration and login forms on the frontend.
  - Develop APIs for profile management using CRUD operations (view, update).
  - Ensure secure password storage and validation.

5. **Artist Management**

- **Description:** Manages artist registration, login, profile creation, and updating.
- **Process:**
  - Develop artist registration and login forms.
  - Implement backend services for artist profile management.
  - Allow artists to upload portfolio images.
  - Validate artist data and ensure secure storage.

**6. Search and Booking**

- **Description:** Enables users to search for artists based on criteria and submit booking requests.
- **Process:**
  - Implement search functionality with filters (location, availability, etc.).
  - Create booking request forms.
  - Develop backend services for handling booking requests and confirmations.
  - Notify users and artists of booking status.

**7. Event Management**

- **Description:** Allows users to create, update, view, and cancel events.
- **Process:**
  - Design event management interfaces on the frontend.
  - Develop APIs for event CRUD operations.

- Integrate calendar and scheduling features.
- Ensure proper validation and notification for event changes.

**8. Reviews and Ratings**

- **Description:** Users can leave reviews and ratings for artists and events.
- **Process:**
  - Create forms for submitting reviews and ratings.
  - Implement backend services to store and retrieve reviews.
  - Display reviews and ratings on artist and event pages.
  - Validate and sanitize user input.

**9. Notification System**

- **Description:** Sends notifications for booking requests, confirmations, and updates.
- **Process:**
  - Implement notification services in the backend.
  - Integrate email and SMS notification providers.
  - Trigger notifications on relevant actions (booking, event updates).
  - Ensure notifications are user-friendly and timely.

**10. Data Security**

- **Description:** Ensures secure login, authentication, and data encryption.
- **Process:**
  - Implement secure login and authentication mechanisms.
  - Use encryption for sensitive data storage.
  - Regularly update security protocols and monitor for vulnerabilities.
  - Conduct security audits and penetration testing.

**11. Blog Page**

- **Description:** Users can create and view blog posts related to events and artists.
- **Process:**
  - Develop interfaces for blog creation and viewing.
  - Implement backend services for managing blog posts.
  - Allow image uploads and content management.
  - Ensure proper moderation and content validation.

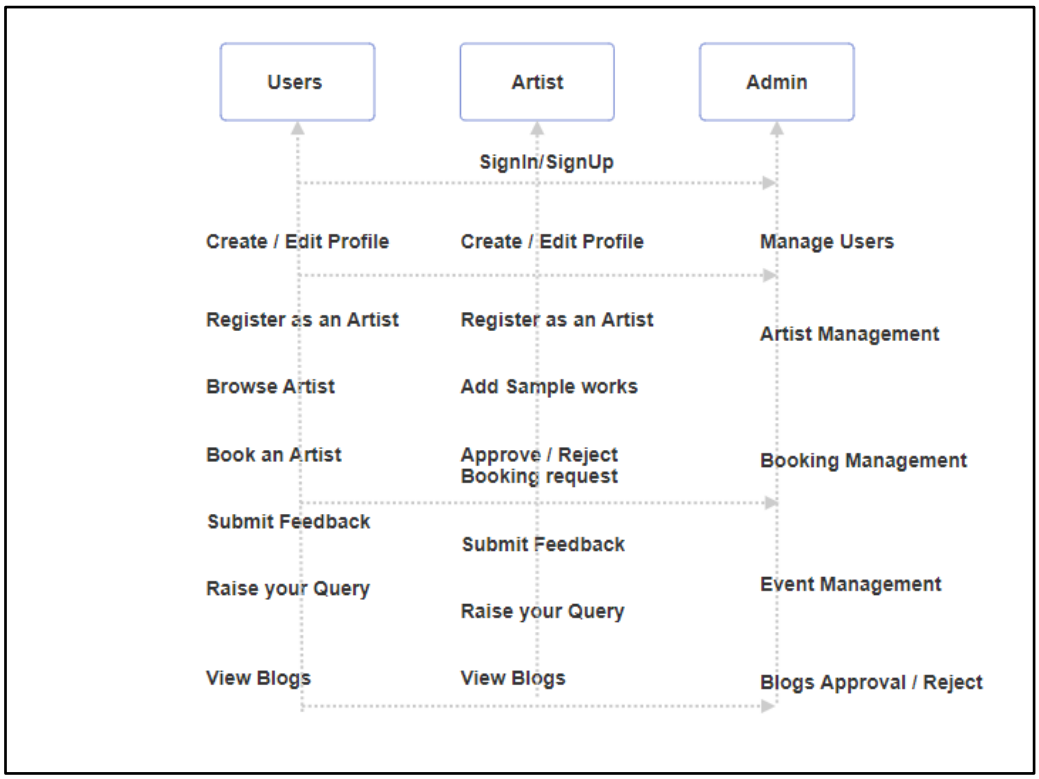


***Figure 9.1: Component Design***

# 10.0 Any Specific Design Considerations

When designing the GetYourArtist platform, several specific settings must be considered to ensure that the system is robust, user-friendly, and scalable. Here are the main design considerations:

### 1.1. Scalability:

- The system should handle an increasing number of users and data without performance degradation.
- Use microservices architecture and cloud-based infrastructure (e.g., AWS, Azure) to allow for horizontal scaling. Implement load balancers to distribute traffic efficiently.

### 1.2. Performance:

- The platform should provide a responsive and fast user experience.
- Optimize database queries, use caching mechanisms (e.g., Redis), and employ Content Delivery Networks (CDNs) for faster content delivery. Implement asynchronous processing where applicable.

### 1.3. Security:

- Protect user data and ensure secure interactions within the platform.
- Implement HTTPS for all communications, use strong encryption for sensitive data, and follow best practices for secure coding. Regularly conduct security audits and penetration testing.

### 1.4. User Experience (UX):

- The platform should be intuitive and easy to use for both artists and users.
- Invest in professional UI/UX design, conduct usability testing, and gather user feedback to continuously improve the interface. Ensure consistency in design elements and navigation.

### 1.5. Accessibility:

- The platform should be accessible to users with disabilities.
- Follow Web Content Accessibility Guidelines (WCAG) and ensure that all features are usable by people with various disabilities. Provide alternative text for images, ensure keyboard navigability, and use ARIA landmarks.

### 1.6. Mobile Responsiveness:

- The platform should work seamlessly across various devices and screen sizes.
- Implement responsive web design using CSS frameworks like Bootstrap or Tailwind CSS. Test the platform on different devices and browsers to ensure compatibility.

### 1.7. Modularity and Maintainability:

- The system should be easy to maintain and extend.
- Use modular architecture, write clean and well-documented code, and follow SOLID principles. Implement automated testing and continuous integration/continuous deployment (CI/CD) pipelines.

### 1.8. Data Management:

- Efficient storage, retrieval, and management of user and artist data.
- Use NoSQL databases like MongoDB for flexible data models. Implement indexing and sharding for improved performance. Ensure regular data backups and implement data integrity checks

## 11.0 Cross Reference with System Requirement Specification

| System Requirement Specification Features | | | |
|---|---|---|---|
| **Functional Requirement (FR) ID** | **Description** | **API Endpoint** | **Adherence/ Limitations** |
| FR1.1 | Users must provide valid email, password, and personal details to register. | /register | Meets the FR completely |
| FR1.2 | Users must enter a valid email and password to log in. | /login | Meets the FR completely |
| FR1.3 | Users can request a password reset link via email. | /password-reset/request | Meets the FR completely |
| FR1.4 | Users can change passwords from their profile settings. | /password-reset/confirm | Meets the FR completely |
| FR2.1 | Artists can upload photos, videos, and other media to their profiles. | /profile/media/upload | Meets the FR completely |
| FR2.2 | Users can browse artist profiles and view detailed information, including media and reviews. | /profile/view/{artistId} | Meets the FR completely |
| FR2.3 | Users can search for artists based on various criteria such as genre, location, and availability. | /search/artists | Meets the FR completely |
| FR3.1 | Users can use filters to refine search results. | /search/artists | Meets the FR completely |
| FR3.2 | Users can provide event details and submit booking requests to artists. | /booking/request | Meets the FR completely |
| FR3.3 | Artists can view, accept, or | /booking/manage | Meets the FR |

| | | | |
|---|---|---|---|
| | decline booking requests. | | completely |
| FR4.1 | Users can view a list of upcoming and past bookings. | /booking/history | Meets the FR completely |
| FR4.2 | Users can update event details or cancel bookings, subject to artist approval. | /booking/update | Meets the FR completely, additional functionality required for artist approval workflow |
| FR5.1 | Users can leave reviews and ratings for artists after events. | /reviews/add | Meets the FR completely |

# 12.0 Conclusion

- **User-Centric Platform:** GetYourArtist has broken ground in booking artistry by providing a friendly user interface that connects the artists, event organizers, and the end-user. This has further increased the pace of making things simple in booking and extending access to a vast array of different artists.
- **Deep Features:** Round Artist Profiles, Advanced Search, Efficient Management software for events, Support for Multimedia Content, Scheduling, and Real-Time Updates—all combine to ensure a seamless experience in booking.
- **Community Engagement:** GetYourArtist encourages community engagement through interesting features like blogs by artists and interactive user profiles. There's what sets up engagement, the sharing of ideas, and networking between the artistic community.
- **Technical Excellence:** Gundam-backed to scale on infrastructure, RESTful APIs, and MongoDB assure high performance and reliability; seamless integration across web and mobile platforms is assured.
- **User Benefits:** More visibility and management tools for artists, and the event organizer has at his or her disposal an extremely large pool of diverse talent with easy booking options. An artist, literally at every single user's fingertips to perform in personal events, makes the entire experience of event planning a delight.
- **Future Prospects:** The platform will get updated and improved continuously according to the feedback from users, new technologies, and enhancing its capabilities to match growing user demand and the changing requirements of the industry.
- Such conclusions allow drawing a line under the fact that GetYourArtist is a pack of functionality, building community, technical robustness, user benefits, and bright prospects for further development, hence changing the face of artist booking

# 13.0 Appendix

### Appendix A: Glossary of Terms

This section provides definitions for key terms and acronyms used throughout the document.

- **API**: Application Programming Interface
- **UI**: User Interface
- **UX**: User Experience
- **SRS**: System Requirement Specification
- **DBMS**: Database Management System
- **REST**: Representational State Transfer

---

### Appendix B: Use Case Diagrams

This section includes detailed diagrams depicting various interactions between users and the system.

- **Use Case Diagram 1: User Registration**
  - Illustrates the steps involved when a user registers on the platform.
- **Use Case Diagram 2: Booking Process**
  - Shows the interactions and steps for booking an artist for an event.

---

### Appendix C: Data Entity Relationship Diagram (ERD)

This section presents the Entity Relationship Diagram depicting the database structure and relationships.

- **Entity Relationship Diagram (ERD)**
  - Diagram illustrating how different entities (users, artists, bookings) relate to each other within the database.

---

### Appendix D: API Documentation

This section provides detailed documentation for each API endpoint used in the system.

- **API Documentation**
  - Includes endpoints, parameters, responses, and sample requests for each API (e.g., /register, /login, /book-artist).

## Appendix E: System Architecture Diagram

This section presents an architectural diagram of the system, showing components and their interactions.

- **System Architecture Diagram**
  - Illustrates the overall structure of the GetYourArtist platform, including servers, databases, APIs, and user interfaces.

---

## Appendix F: User Interface Mockups

This section includes mockups of key user interfaces to visualize the design and layout.

- **User Interface Mockups**
  - Mockups for user registration, artist profiles, event booking, and admin dashboard interfaces.

---

## Appendix G: Performance Testing Results

This section provides results from performance testing conducted on the platform.

- **Performance Testing Results**
  - Includes metrics such as response times, throughput, and resource utilization under various loads.

---

## Appendix H: References

This section lists all the external sources and references used in developing the SRS.

- **References**
  - Includes research papers, documentation, and any other sources consulted during the development of the system.