



**CSC 431 – Spring 2025**

## **Coral Gables Live Music Booking Application**

### **Software Requirements Specification (SRS)**

Alan Fiore	Software Developer
David Erulker	Software Developer
Tal Ram	Software Developer

# Version History

Version	Date	Author(s)	Change Comments
<b>1</b>	02/21/2025	Alan Fiore, David Erulker, Tal Ram	Initial Draft

# Table of Contents

1. System Requirements	6
1.1 Functional Requirements	6
1.1.1 Artist Profile Creation and Management	6
1.1.2 Venue Profile Management	6
1.1.3 Geolocation Services for Venue and Artist Discovery	7
1.1.4 In-App Messaging System	7
1.1.5 Integrated Payment Processing	8
1.1.6 Booking Management System for Venues	8
1.2 Non-Functional Requirements	9
1.2.1 Performance	9
1.2.2 Security	9
1.2.3 Usability	9
1.2.4 Availability	9
1.2.5 Scalability	9
2. System Constraints	10
2.1 Tool Constraints	10
2.1.1 Developer Tools	10
2.2 Language Constraints	10
2.2.1 Programming Languages	10
2.3 Platform Constraints	10
2.3.1 Mobile Platforms	10
2.4 Hardware Constraints	10
2.4.1 Device Compatibility	10
2.5 Network Constraints	11
2.5.1 Connectivity	11
2.6 Deployment Constraints	11
2.6.1 Deployment Platforms	11
2.7 Transition & Support Constraints	11
2.7.1 User Support	11
2.8 Budget & Schedule Constraints	11
2.8.1 Development Budget	11
2.8.2 Project Timeline	12
2.9 Miscellaneous Constraints	12
2.9.1 Legal Compliance	12
3. Requirements Modeling	13
1.1.1 Artist Profile Creation and Management	13
4. Evolutionary Requirements	14
4.1 Functional Requirements	14
4.1.1 Ticketing System	14
4.2 Non-Functional Requirements	14
4.2.1 Enhanced Scalability	14

# Table of Tables

Table Number	Table Title	Page Number
1.1.1	Artist Profile Creation and Management	6
1.1.2	Venue Profile Management	6
1.1.3	Geolocation Services for Venue and Artist Discovery	7
1.1.4	In-App Messaging System	7
1.1.5	Integrated Payment Processing	8
1.1.6	Booking Management System for Venues	8
1.2.1	Performance	9
1.2.2	Security	9
1.2.3	Usability	9
1.2.4	Availability	9
1.2.5	Scalability	9
2.1.1	Developer Tools	10
2.2.1	Programming Languages	10
2.3.1	Mobile Platforms	10
2.4.1	Device Compatibility	10
2.5.1	Connectivity	11
2.6.1	Deployment Platforms	11
2.7.1	User Support	11
2.8.1	Development Budget	11
2.8.2	Project Timeline	12
2.9.1	Legal Compliance	12
4.1.1	Ticketing System	14
4.2.1	Enhanced Scalability	14

# Table of Figures

Figure Number	Figure Title	Page Number
3.1	Use Cases for Artist Profile Creation and Management	13
3.2	Use Cases for Venue Profile Management	13
3.3	Use Cases for Geolocation Services for Venue and Artist Discovery	14
3.4	Use Cases for In-App Messaging System	14
3.5	Use Cases for Integrated Payment Processing	15
3.6	Use Cases for Booking Management System for Venues	15
3.7	Use Cases for Coral Gables Live Music Booking Application	16

# 1. System Requirements

## 1.1 Functional Requirements

### 1.1.1 Artist Profile Creation and Management

Title	Artist Profile Management
Description	The system shall allow musicians to create and manage comprehensive profiles including band/artist images, performance videos, social media integration (Instagram, TikTok), genre classification, and a detailed artist description. The system should allow artists to edit and update their profiles at any time.
Priority	0
Precondition(s)	Users must have a registered account and be logged in.
Basic Flow	User navigates to the "Profile" section. User selects "Edit Profile". System displays editable fields (band/artist image, video links, social media links, genre, description). User enters or modifies information in the fields. User clicks "Save". System validates the input data. System saves the profile information to the database. System displays a success message.
Postconditions(s)	Artist profiles are created or updated in the system. The updated information is visible to other users (venue owners) where appropriate.
Use Case Diagram	3.1, 3.7

### 1.1.2 Venue Profile Management

Title	Venue Profile Management
Description	The system shall allow venue owners to create and manage profiles including venue images, availability calendar, pricing structure, seating capacity, room size, location details, and a list of pre-existing equipment. Venue owners should be able to update this information as needed.
Priority	1
Precondition(s)	Users must have a registered venue owner account and be logged in.
Basic Flow	User navigates to the "Venue Profile" section. User selects "Edit Venue Profile". System displays editable fields (venue images, availability calendar, pricing, capacity, location, equipment). User enters or modifies information in the fields, including setting availability on the calendar. User clicks "Save". System validates the input data. System saves the venue profile information to the database. System displays a success message.

Postconditions(s)	Venue profile is created or updated in the system. The updated information is visible to other users (artists) where appropriate.
Use Case Diagram	3.2, 3.7

### 1.1.3 Geolocation Services for Venue and Artist Discovery

Title	Geolocation Services
Description	The system shall integrate with Google Maps API to provide geolocation services, enabling artists and venues to discover each other based on proximity. Artists should also be able to filter venues based on their own metrics.
Priority	2
Precondition(s)	Artists and venues must have their location information set in their profiles. Google Maps API key must be configured.
Basic Flow	User (artist or venue) opens the map view in the app. System uses Google Maps API to display a map centered on the user's current location (if location services are enabled) or a default location (Coral Gables). System retrieves nearby artists/venues from the database, based on a defined radius. System displays markers on the map for each artist/venue. Users can filter the results based on criteria (e.g., genre, popularity, venue capacity). Users can click on a marker to view the artist/venue profile.
Postconditions(s)	Users can view nearby artists and venues on a map. They can filter the results based on their criteria.
Use Case Diagram	3.3, 3.7

### 1.1.4 In-App Messaging System

Title	In-App Messaging
Description	The system shall provide an in-app messaging system for direct communication between artists and venue owners.
Priority	3
Precondition(s)	Both artist and venue owner must have accounts.
Basic Flow	User (artist or venue owner) navigates to the messaging section. User selects a contact (another artist or venue owner). System displays the message history with the selected contact. User enters a new message. User clicks "Send". System sends the message to the recipient. System displays the sent message in the message history. Recipient receives a notification of a new message.
Postconditions(s)	Messages are sent and received between artists and venue owners within the application. Notifications are sent to the recipient.
Use Case Diagram	3.4, 3.7

### 1.1.5 Integrated Payment Processing

Title	Payment Processing
Description	The system shall provide integrated payment processing capabilities for booking transactions. Details on payment methods and security measures TBD.
Priority	5
Precondition(s)	Both artist and venue owner have agreed on booking terms (date, time, price). A payment gateway (Stripe, PayPal) is integrated.
Basic Flow	Venue owner sends a booking request to the artist. Artist accepts the booking request. System displays the payment details (date, time, price, payment method options). Venue owner selects a payment method and enters payment information. System securely transmits the payment information to the payment gateway. Payment gateway processes the payment. System receives a confirmation of the payment status from the payment gateway. If payment is successful, the booking is confirmed.
Postconditions(s)	Payment is processed securely. The booking is confirmed if the payment is successful.
Use Case Diagram	3.5, 3.7

### 1.1.6 Booking Management System for Venues

Title	Booking Management
Description	The system shall provide venue owners with a booking management system to manage booking requests, confirm bookings, and manage their venue's availability.
Priority	4
Precondition(s)	Venue owner must be logged in. Venue profile must exist with availability set.
Basic Flow	Venue owner navigates to the "Booking Management" section. System displays a calendar view of the venue's availability. System displays a list of pending booking requests. Venue owners can view details of each booking request (artist profile, date, time, price). Venue owners can accept or reject a booking request. If the venue owner accepts the booking request, the system updates the calendar to mark the date/time as booked. System sends a notification to the artist about the booking status.
Postconditions(s)	Venue owners can manage their bookings effectively. Artists are notified of booking status changes.
Use Case Diagram	3.6, 3.7



## 1.2 Non-Functional Requirements

### 1.2.1 Performance

Title	Application Responsiveness
Description	The application shall respond to user requests within 3 seconds on average to ensure a smooth user experience.
Priority	1
Applicable FR(s)	1.1.1, 1.1.2, 1.1.3, 1.1.4, 1.1.5, 1.1.6

### 1.2.2 Security

Title	Data Security and Privacy
Description	The application shall protect user data and financial information using industry-standard security measures, including encryption and secure authentication, complying with relevant data privacy laws.
Priority	0
Applicable FR(s)	1.1.1, 1.1.2, 1.1.4, 1.1.5, 1.1.6

### 1.2.3 Usability

Title	User-Friendliness
Description	The application shall be user-friendly and easy to navigate for both musicians and venue owners, with a clear and intuitive interface, minimizing the learning curve.
Priority	1
Applicable FR(s)	1.1.1, 1.1.2, 1.1.4, 1.1.5

### 1.2.4 Availability

Title	System Uptime
Description	The application shall be available 24/7, with a target uptime of 99.9% to ensure continuous access for users.
Priority	2
Applicable FR(s)	All

### 1.2.5 Scalability

Title	Capacity for Growth
Description	The application shall be designed to handle a growing number of users and venues without significant performance degradation, allowing for future expansion.
Priority	3
Applicable FR(s)	All

## 2. System Constraints

### 2.1 Tool Constraints

#### 2.1.1 Developer Tools

Title	Developer Tools - IDE
Description	Use of specific IDE for development such as IntelliJ or Visual Code Studio.
Priority	0

### 2.2 Language Constraints

#### 2.2.1 Programming Languages

Title	Programming Languages
Description	The program backend must be built on Node.js and the front end must be built using JavaScript (React Native)
Priority	0

### 2.3 Platform Constraints

#### 2.3.1 Mobile Platforms

Title	Mobile Platforms
Description	The application must be compatible with both iOS and Android.
Priority	0

### 2.4 Hardware Constraints

#### 2.4.1 Device Compatibility

Title	Device Compatibility
Description	The application should be compatible with a wide range of iOS and Android devices, including different screen sizes and hardware capabilities.
Priority	0

## 2.5 Network Constraints

### 2.5.1 Connectivity

Title	Connectivity
Description	Stable internet connection will be required for messaging and payment.
Priority	0

## 2.6 Deployment Constraints

### 2.6.1 Deployment Platforms

Title	Deployment Platforms
Description	The application will be deployed on the Apple App Store and Google Play Store and follow their respective policies.
Priority	0

## 2.7 Transition & Support Constraints

### 2.7.1 User Support

Title	User Support
Description	User guides and FAQs will be provided to assist users during the transition to the new system.
Priority	0

## 2.8 Budget & Schedule Constraints

### 2.8.1 Development Budget

Title	Development Timeline
Description	The development budget is \$50,000.
Priority	0

## 2.8.2 Project Timeline

Title	Project Timeline
Description	The project should be completed within 9 months.
Priority	1

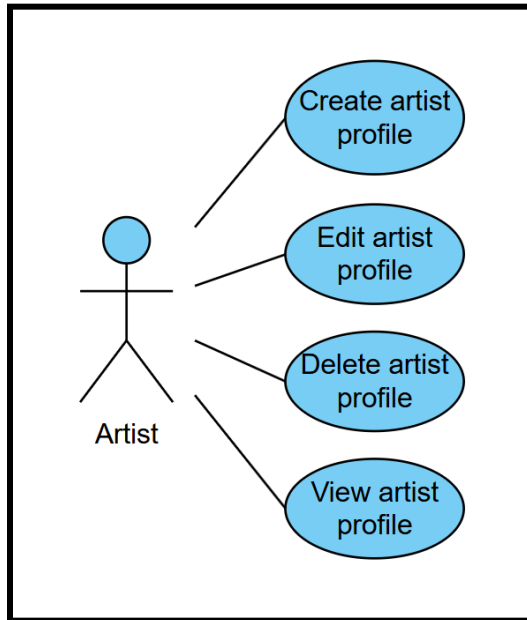
## 2.9 Miscellaneous Constraints

### 2.9.1 Legal Compliance

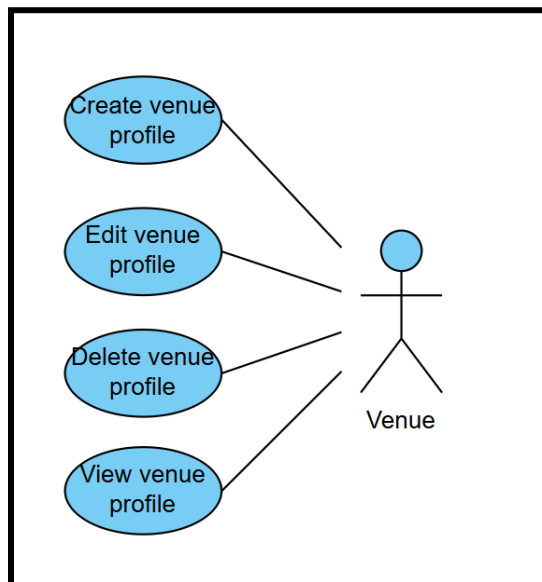
Title	Data Privacy
Description	The application must comply with all relevant legal and regulatory requirements, including data privacy laws (e.g., GDPR, CCPA).
Priority	0

### 3. Requirements Modeling

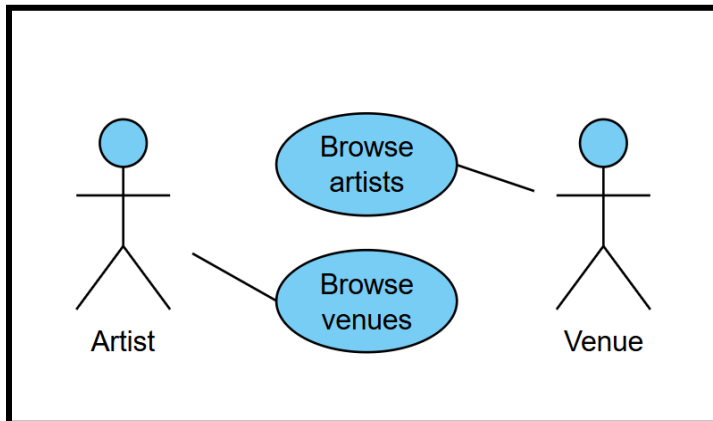
#### 3.1 Use Cases for Artist Profile Creation and Management



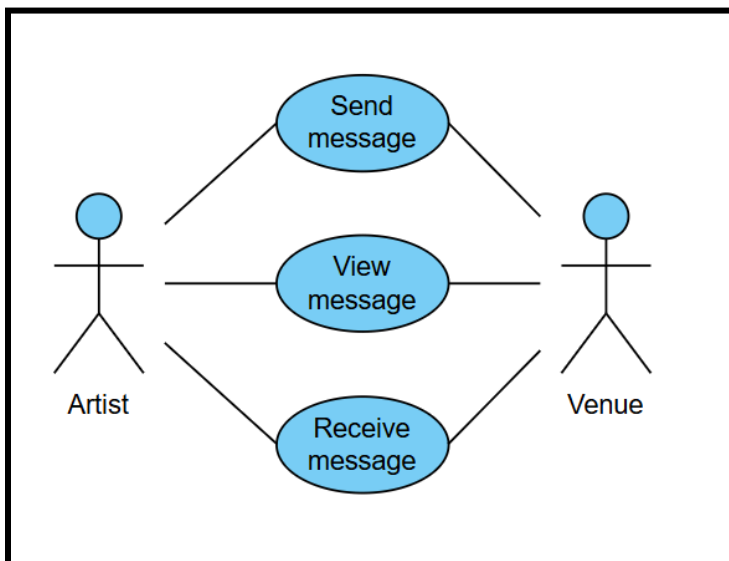
#### 3.2 Use Cases for Venue Profile Management



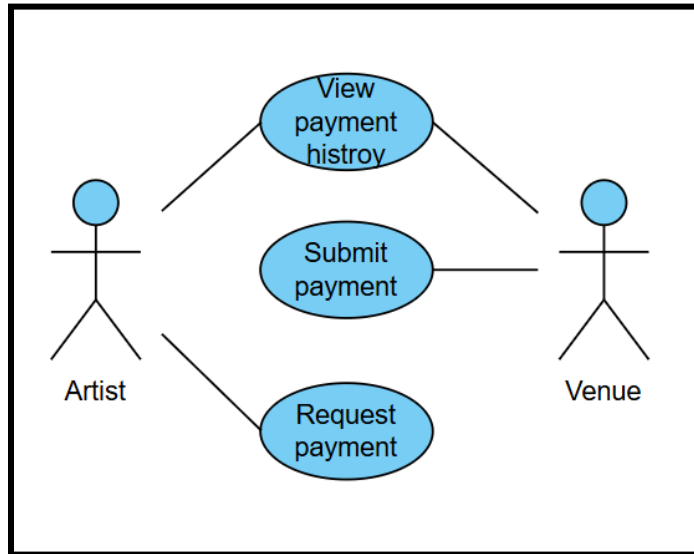
### 3.3 Use Cases for Geolocation Services for Venue and Artist Discovery



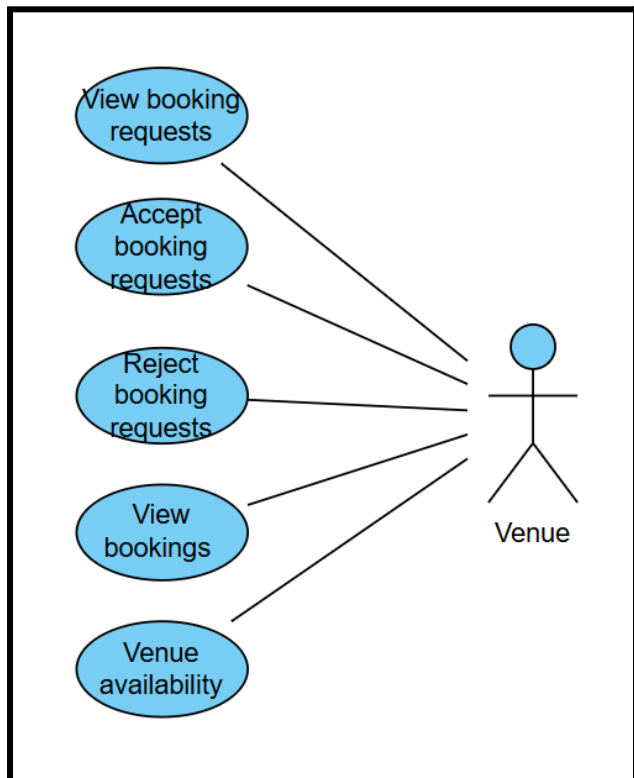
### 3.4 Use Cases for In-App Messaging System



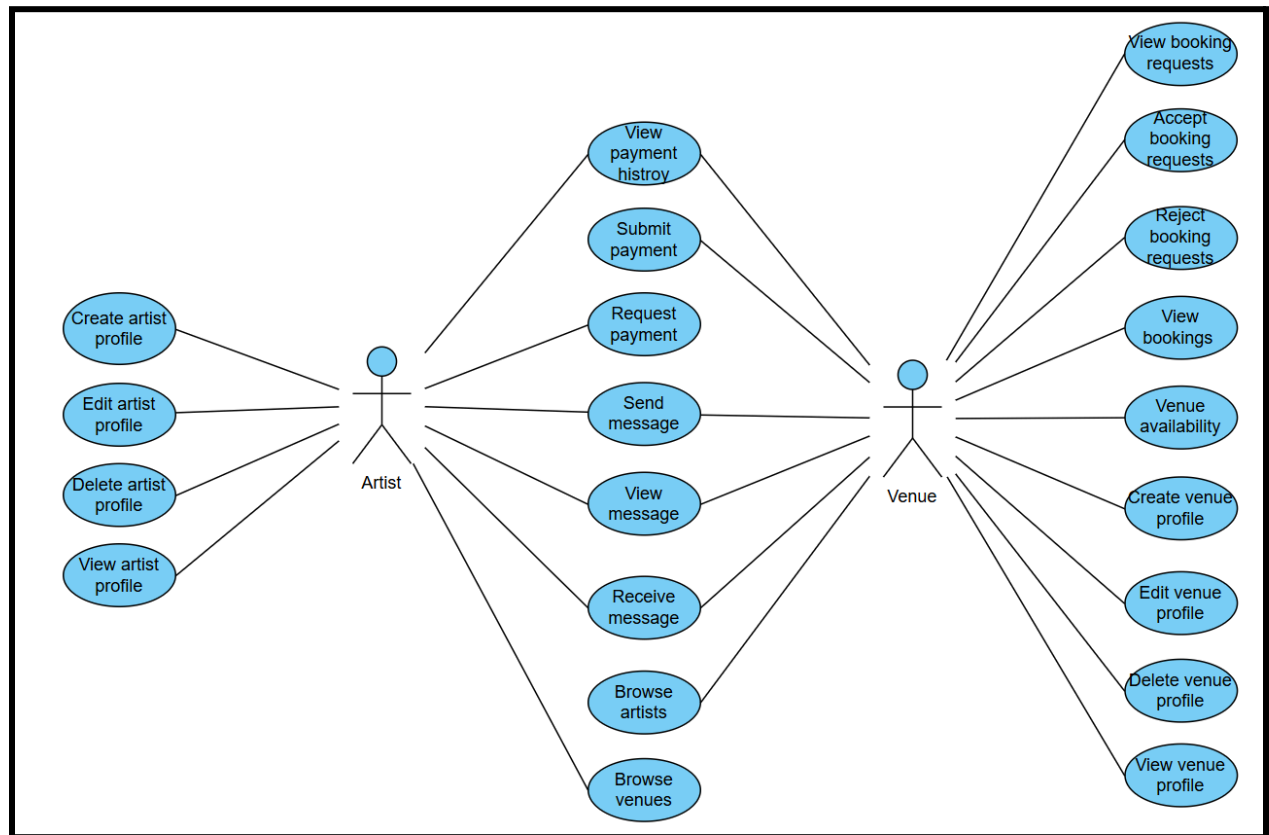
### 3.5 Use Cases for Integrated Payment Processing



### 3.6 Use Cases for Booking Management System for Venues



### 3.7 Use Cases for Coral Gables Live Music Booking Application





## 4. Evolutionary Requirements

### 4.1 Functional Requirements

#### 4.1.1 Ticketing System

Title	Ticketing System
Description	In future releases, the system should support integrated ticketing functionality for venues to sell tickets to events directly through the app.
Priority	0
Precondition(s)	The venue must have a complete and validated profile within the system, including accurate seating capacity, event space details, and contact information.
Postconditions(s)	Tickets for the event are successfully listed and available for purchase by users of the app.
Use Case Diagram	N/A

### 4.2 Non-Functional Requirements

#### 4.2.1 Enhanced Scalability

Title	Enhanced Scalability
Description	Future versions of the system should be designed to handle a significantly larger user base and transaction volume as the application expands to other regions.
Priority	0
Applicable FR(s)	Ticketing System