

FINAL PROJECT PROPOSAL

Project Participants: Eyusias Woldegiorgis

Title: Event Management System

Executive Summary:

Event Management System is used to manage events information in Las Vegas. The event includes different kinds of shows, conventions, and sports events like boxing, Las Vegas hockey and football teams in different hotels, stadiums, concert holes, etc. This Event Management System register and gives Las Vegas area event detail information including the category and venue of the event to the end users/applications.

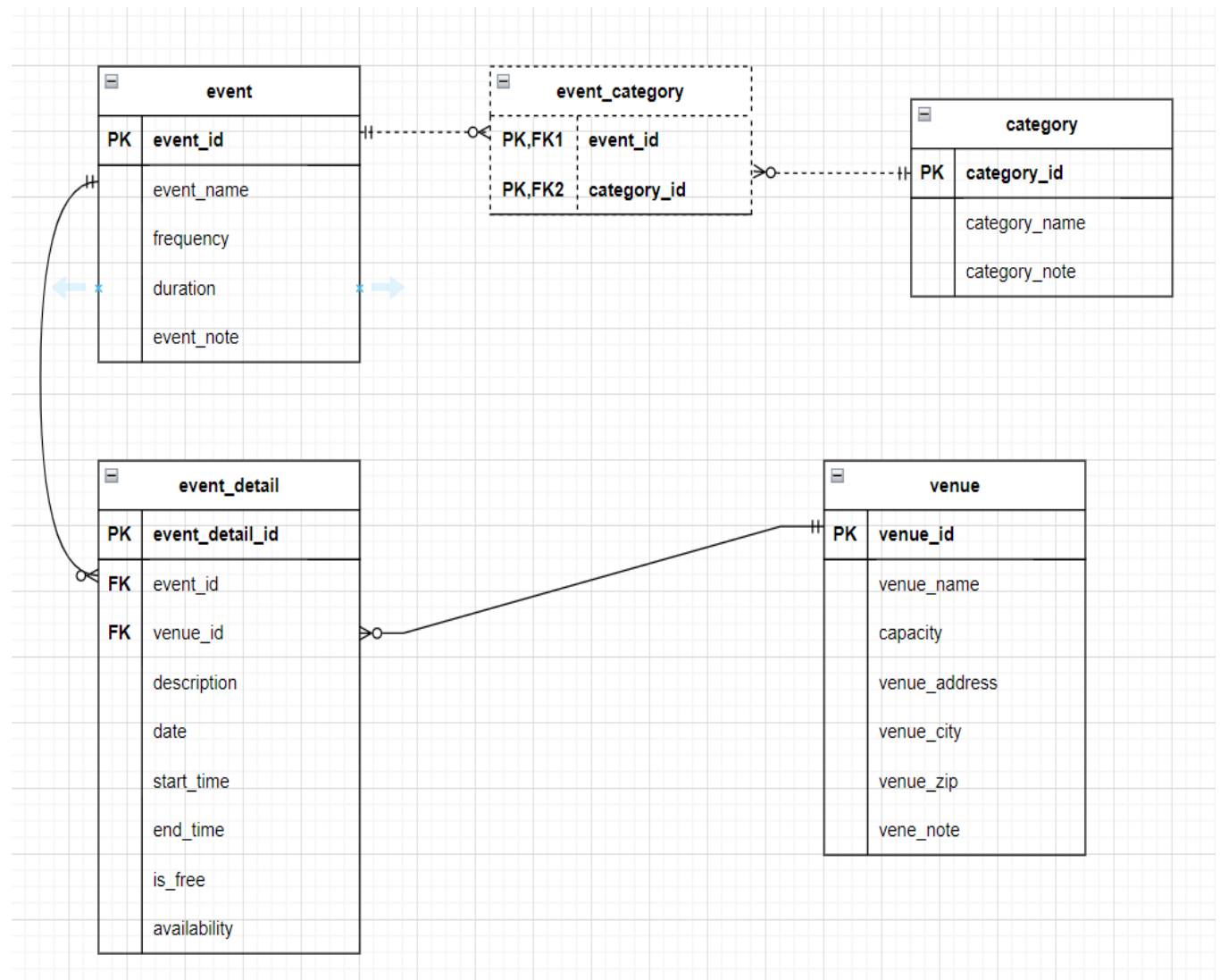
Entities: Event, Event Detail, Category, and Venue

- The **Category** table will contain information such as category name and note.
- The **Event** table will contain information such as the event name, frequency, duration, and event note.
- The **Event Detail** table will contain detail event information such as the event description, date of the event, start time of the event, end time of the event, is free status, and event availability status.
- The **Venue** table will contain information such as the venue name, capacity, venue address, city, zip, and note.

Relationships between the entities:

- The **Event** and **Category** table have an **optional many-to-many relationship**, and also use a **join table (event_category)**
- The **Event** and **EventDetail** tables have an **optional one-to-many relationship**. An Event may have zero or more EventDetail, and each EventDetail is only associated with a single Event. (Referenced by event_id)
- Finally, The **EventDetail** and **Venue** tables have an **optional one-to-many relationship**. A Venue may have zero or more EventDetail, and each EventDetail is only associated with a single Venue. (referenced by venue_id)

Entity Relationship Diagram:



Description of endpoints/features:

/categories

Verb	Response	Request Body
GET	A list of all categories and a 200 status code, or 404 status code if no categories are found.	<i>none</i>
POST	The created category entity and a 201 status code	JSON represents the category entity data model.

/categories/{category_id}

**Note the application will return a 404 status code if the category Id is not found or a 409 status code if the category Id is not valid format with an additional message to the user.*

Verb	Response	Request Body
GET	A specific category entity, or a 404 status code if the category id is not found, or a 409 status code if the category id is not a valid format.	<i>none</i>
PUT	The updated category entity and a 200 status code, or a 404 status code if the category id is not found.	JSON containing the category entity data model with modified field value(s)
DELETE	JSON format deletion message with a 200 status code, if successful. If a category has associated with any eve, a 405 (method not allowed) status code will be returned instead	<i>none</i>

/events

Verb	Response	Request Body	Notes
GET	A list of all events and their associated category and a 200 status code, or 404 status code if no events are found.	<i>none</i>	
POST	The created event and a 201 status code, or a 405 (method not allowed) status code will be returned if a list of categories is null or not in the category table	JSON represents the event entity data model.	

/events/{event_id}

Verb	Response	Request Body	Notes
GET	A specific event entity and 200 status code, or a 404 status code if the event id is not found, or a 409 status code if the event Id is not a valid format.	<i>none</i>	

/events/{event_id}/event-details

Verb	Response	Request Body	Notes
GET	A list of event details associated with a specific event Id and 200 status code, or a 404 status code if the event id or event details is not found, or a 409 status code if the event Id is not a valid format.	<i>none</i>	An alternative way of retrieving event details is by using event Id as a path variable.

/events/categories

**Note: only one parameter would be included in any given GET request or the application will return a 400 response code with an additional message to the user specifying this requirement. And also the application will return a 404 status code if the event is not found, a 400 status code if the parameter name is not valid, and a 409 status code if the parameter value is not valid. All parameters sent to GET /events/categories would take the form of query strings.*

Verb	Parameters	Response	Request Body
GET	none	A list of all events with their associated categories and a 200 status code.	none
	category_id	A list of all events with their associated categories by a specific category Id and a 200 status code.	
	category_name	A list of all events with their associated categories by specific category name and a 200 status code.	

/venues

Verb	Response	Request Body
GET	A list of all venues and a 200 status code, or 404 status code if no venues are found	none
POST	The created venue entity and a 201 status code	JSON represents the venue entity data model.

/venues /{venue_id}

**Note the application will return a 404 status code if the venue by Id is not found or a 409 status code if the venue Id is not valid format with an additional message to the user.*

Verb	Response	Request Body
GET	A specific venue entity, or a 404 status code if the venue id is not found	<i>none</i>
PUT	The updated venue entity and a 200 status code, or a 404 status code if the venue id is not found.	JSON containing the venue entity data model with modified field value(s)
DELETE	JSON format deletion message with a 200 status code, if successful. If a venue has associated with any event details, a 405 (method not allowed) status code will be returned instead.	<i>none</i>

/venues/{venue_id}/event-details

Verb	Response	Request Body	Notes
GET	A list of event details associated with a specific venue Id and 200 status code, or a 404 status code if the venue id or event details is not found, or a 409 status code if the venue Id is not a valid format.	<i>none</i>	An alternative way of retrieving event details is by using venue Id as a path variable.

/event-details

**Note: only one parameter would be included in any given GET request or the application will return a 400 response code with an additional message to the user specifying this requirement. And also the application will return a 404 status code if the event detail is not found, a 400 status code if the parameter name is not valid, and a 409 status code if the parameter value is not valid. All parameters sent to GET /event-details would take the form of query strings.*

Verb	Parameters	Response	Request Body
GET	none	A list of all event details and their associated data.	<i>none</i>
	date	A list of all event details by a specific date and a 200 status code.	
	event_id	A list of all event details by a specific event Id and 200 status code.	
	event_name	A list of all event details by a specific event name and 200 status code.	
	venue_id	A list of all event details by a specific venue Id and 200 status code.	
	venue_name	A list of all event details by a specific venue name and 200 status code.	
POST	event_id and venue_id	The created event detail and a 201 status code. 404 status code if the event or/and venue id is not valid, found, or provided. 400 status code: If the request body isn't provided.	JSON represents the event detail entity data model.

PUT	date	<p>JSON format event detail availability field modification status message. The message contains a number of rows affected.</p> <p>409 status code if the parameter date value is not a valid format.</p>	<p>none</p> <p>*note Based on the project requirement we only modify the event availability field to false by a specific date when an event is passed.</p>
-----	------	---	--

/event-details/{event_detail_id}

**Note the application will return a 404 status code if the event detail by Id is not found or a 409 status code if the event detail Id is not valid format with an additional message to the user.*

Verb	Response	Request Body	Note
PUT	The updated program entity data model and a 200 status code.	none	Based on the project requirement we only modify the event availability field to false by Id when the event is passed.
DELETE	JSON format deletion successful message.	none	

/events/{event_id}/venues/{venue_id}/event-details

Note: Alternative way of creating an event detail by using event Id and venue Id as a path variable.

Verb	Response	Request Body	Notes
POST	The created event detail and a 201 status code. 404 status code if the event or/and venue id is not found. 409 status code if the event or/and venue id is not valid.	JSON representing the event detail entity	

Event Management System enhancement modules:

- Adding Spring Security Features
- Adding user account management
- Adding customer management
- Handling tickets based on seat and availability
- Processing payment by using payment APIs (Stripe)