Music Event System

SWEN90007 Software Design and Architecture

Project: Part 1A

TEAM NAME - **MusicBandTeam**

Team Members	Student ID
Bingqing Zhang	1377017
Xin Xiang	1294725
Dhakshayani Perumal	1184165
Vivien Guo	1173459

Use case Diagram

List and description of Use Cases:

Use Case 1: Administrator Creates a Venue

Actors: **Administrator**

Basic Flow:

The Administrator logs into the application using their credentials and proceeds to access the administrative panel. Within the panel, the Administrator selects the option to "Create Venue." The System presents a form that prompts the

Administrator to input essential details of the new venue, including the venue name, address, and sections with their respective types and capacities.

Once the Administrator fills in all the required information, they can submit the form. The System will validate the input and create the new venue within the application. It will then display a confirmation message to the Administrator, indicating the successful creation of the venue.

Post-Condition:

The Administrator will be able to view the venue's information. The newly added venue will also be available for selection during event creation by Event Planners.

Use Case 2: Administrator Modifies a Venue

Actors: **Administrator**

Basic Flow:

The Administrator logs into the application using their credentials and accesses the administrative panel. Within the panel, the Administrator selects the option to "View Venues." A list of existing venues is displayed, and the Administrator chooses the venue they wish to modify.

The system shows the details of the selected venue, including its name, address, and section capacities. The Administrator can now make the desired modifications to the venue information.

Once the changes are made, the Administrator submits the updated details. The System validates the changes and proceeds to update the venue's information. It will then display a confirmation message to the Administrator, indicating the successful modification of the venue.

Post-Condition:

The updated venue information will be available to the Administrator and Event Planners.

Use Case 3: Administrator Deletes a Venue

Actors: **Administrator**

Basic Flow:

The Administrator logs into the application using their credentials and navigates to the administrative panel. Within the panel, the Administrator selects the option to "View Venues." A list of existing venues is displayed, and the Administrator chooses

the venue they wish to remove from the application.The system presents a confirmation message to ensure the Administrator wants to proceed with the deletion.

Upon getting the confirmation from the Administrator, the System removes the selected venue from the application along with its associated data. It will then display a confirmation message to the Administrator, indicating the successful deletion of the venue.

If any of the venue's associated events have not yet occurred, the System will cancel the event and the bookings to these events and send notifications to any Customers who have made them, as well as any Event Planners who are managing the event.

Post-Condition:

Any Event Planner affected by this venue deletion will no longer see this event in their Managed Events page. Any Customers affected by this venue deletion will no longer see their bookings from any event in this venue.

Use Case 4: Administrator Views All Events

Actors: **Administrator**

Basic Flow:

The Administrator logs into the application using their credentials and accesses the administrative panel. Within the panel, the Administrator selects the option to "View All Events." The system retrieves a comprehensive list of all events currently scheduled across Australia. The list includes essential event details, such as the venue, date, time, and ticket prices. The Administrator can further filter and sort the events for better viewing.

Use Case 5: Administrator View Users

Actors: **Administrator**

Basic Flow:

The Administrator logs into the application using their credentials and accesses the administrative panel. Within the panel, the Administrator selects the option to "View Users." The system retrieves a list of all users registered in the application, including Event Planners and Customers. User details such as usernames, emails, and roles (Event Planner or Customer) are displayed for review. The Administrator can also utilize filtering and sorting options to efficiently manage the user list based on specific criteria.

Use case 6: Event Planner creates a new event

Actors: **Event Planner**

Pre-condition:

The Event Planner has created an account as an “Event Planner” within the system.

Basic Flow:

Main Success Scenario

The Event Planner can log in to the system and proceed to create a new event within the application. The System will ask for essential details about the venue (selection from available venues registered within the system), date, time, artist, ticket prices, and a concise description of its content. When the Event Planner submits this information, the System will display a confirmation message regarding the event creation.

Alternate Scenario

If the entered time or date at the chosen venue is already occupied by another event, the System will reject the event creation and ask the Event Planner to select a new venue, date or time.

Post-Condition:

Following the successful creation of the event, it is published on the application, enabling all users to search and view this event. The Customer will also be able to book tickets to this event.

Use case 7: Event Planner modifies the existing event

Actors: **Event Planner**

Pre-condition:

The Event Planner has created an account as an “Event Planner” within the system.

Basic Flow:

Main Success Scenario

Once an event has been created, the Event Planner gains the flexibility to modify essential information about the event at any time by logging into the system. It includes the ability to change the event's date, time, artist, description and ticket prices. Once the Event Planner submits these changes, the System will send notifications to the Customers who have already booked tickets for the event, informing them of the event’s updates.

Alternate Scenario

The creator of an event can invite other existing Event Planners to join as co-Event Planners for the event. These invited Event Planners will be granted access to collaboratively manage the event alongside the original event creator.

Post-Condition

When any user views this event, the updated information will be displayed. If the event date has been changed, this will also be reflected in the 6-month calendar in the Customer view.

Use case 8: Event Planner cancels an upcoming event

Actors: **Event Planner**

Pre-condition:

The Event Planner has created an account as an “Event Planner” within the system, and has either previously created an event (as per Use Case 7) or been added as a co-Event Planner to an existing event (as per Use Case 8’s alternate scenario).

Basic Flow:

The Event Planner logs onto the system and navigates to their Managed Events page, where the System will display a list of events that they manage. From there, they can choose to cancel a particular event. The System will then show a confirmation pop-up asking if they would like to cancel the event. If the event Planner selects “Yes”, the event will be canceled, and the System will show a successful cancellation confirmation message. The System will also cancel all bookings associated with this event, and send a notification to any Customers who have already booked tickets for the event to inform them of the cancellation.

Post-Condition

This event will no longer be displayed in the Administrator view, nor will it show up when Customers are browsing events (in Use Cases 12 & 13).

If any bookings are made to this event, the affected Customers will no longer see these bookings in their Bookings page.

Use case 9: Event Planner views an event's bookings

Actors: **Event Planner**

Pre-condition:

The Event Planner has created an account as an "Event Planner" within the system, and has either previously created an event (as per Use Case 7) or been added as a co-Event Planner to an existing event (as per Use Case 8's alternate scenario).

Basic Flow:

The Event Planner logs onto the system and navigates to their Managed Events page, where the System will display a list of events that they manage. When they navigate to a particular event's Bookings page, the System will display a comprehensive overview of all bookings associated with the events they are managing. This detailed overview will include essential information such as the designated sections, pricing details, and customer information for each booking.

Use case 10: Event Planner cancels a booking

Actors: **Event Planner**

Pre-condition:

The Event Planner has created an account as an "Event Planner" within the system, and has either previously created an event (as per Use Case 7) or been added as a co-Event Planner to an existing event (as per Use Case 8's alternate scenario).

Basic Flow:

The Event Planner logs on and navigates to a particular event's booking page (as per Use Case 9). They select a booking to cancel. The System will cancel this booking and show a confirmation message and send a notification to the corresponding Customer affected by the cancellation.

Post-condition:

The affected Customer will no longer see this booking in their Bookings page.

Use case 11: Event Planner creates account

Actors: **Event Planner**

Basic Flow:

The Event Planner navigates to the Sign Up page, where the System first asks them to select if they want to create an account as a “Customer” or an “Event Planner”. When they select the “Event Planner” option, the System asks them for their username, email address, password, and a re-typed password confirmation. When the Event Planner submits this information, the System will show a confirmation message for successful account creation.

Post-Condition:

The Event Planner is able to log onto the System using their username and password, and be able to view the application as an “Event Planner”.

Use case 12: Customer searches events by name

Actors: **Customer**

Pre-condition:

Customer has created an account as a “Customer” within the system.

Basic Flow:

Main Success Scenario

The Customer logs onto the application and navigates to the search page. They enter the name of an event they wish to seek information about, and the System returns a list of events whose names contain the string entered. (E.g. The customer searching for “Melbourne 2018” may return “Ed Sheeran Tour in Melbourne 2018” and “Melbourne 2018 Tour - Taylor Swift”.)

Alternate Scenario

When the Customer sees this list of events, they may click on one of the events, upon which the System will redirect them to the information page for this particular event.

Use case 13: Customer views the event calendar

Actors: **Customer**

Pre-condition:

Customer has created an account as a “Customer” within the system.

Basic Flow:

The Customer logs onto the application and navigates to the calendar page. The System will display a calendar view of the current month, with the dates showing the events that are scheduled to run on each day. The Customer may choose to view the future months and see the events running up to 6 months.

Alternate Scenario

When the Customer clicks on an event annotation on the calendar, the System will redirect them to the information page for this particular event.

Use case 14: Customer books ticket to an event

Actors: **Customer**

Pre-condition:

Customer has created an account as a “Customer” within the system.

Basic Flow:

The Customer logs onto the application and navigates to an event page (as per the alternate scenarios in Use Case 12 & 13). The System will display the relevant event information (listed in Use Case 6).

When the Customer clicks the “Book Tickets” page, they will be redirected to the booking page. The System will first display the ticket prices for each section, then ask for the number of tickets in each section that the Customer wishes to purchase.

Once the Customer has entered and submitted their total number of tickets, the System will show a booking confirmation page displaying the number of tickets in each section along with tickets’ prices and the total price.

Post-condition:

The System will save information regarding this booking. When the Customer logs in and navigates to their Bookings page, the System will display information regarding this event and the tickets they have purchased as individual bookings. (i.e. Each ticket bought during this process is regarded as one singular booking.)

Event Planners to this event are also able to see these bookings through the event’s Bookings page.

Use case 15: Customer cancels existing booking

Pre-condition:

The Customer has previously made a booking to an event.

Basic Flow:

Main Success Scenario

The Customer logs in and navigates to their Bookings page, where the System will display information regarding the booking they wish to cancel. When they try to cancel the booking, the System will ask if the Customer wishes to cancel the booking. If the Customer selects “Yes”, the System will cancel this booking and show a successful cancellation confirmation noting the event and tickets of this booking.

Alternate Flow

When the System asks if the Customer wishes to cancel their booking and the Customer selects “No”, the System will close the confirmation box. The Customer will be able to view the booking as per usual.

Post-Condition:

When the Customer refreshes the Booking page after a successful cancellation, they will no longer see this booking. The Administrator and Event Planners to this event will also stop seeing this booking in the event’s Bookings page.

Use case 16: Customer creates account

Actors: **Customer**

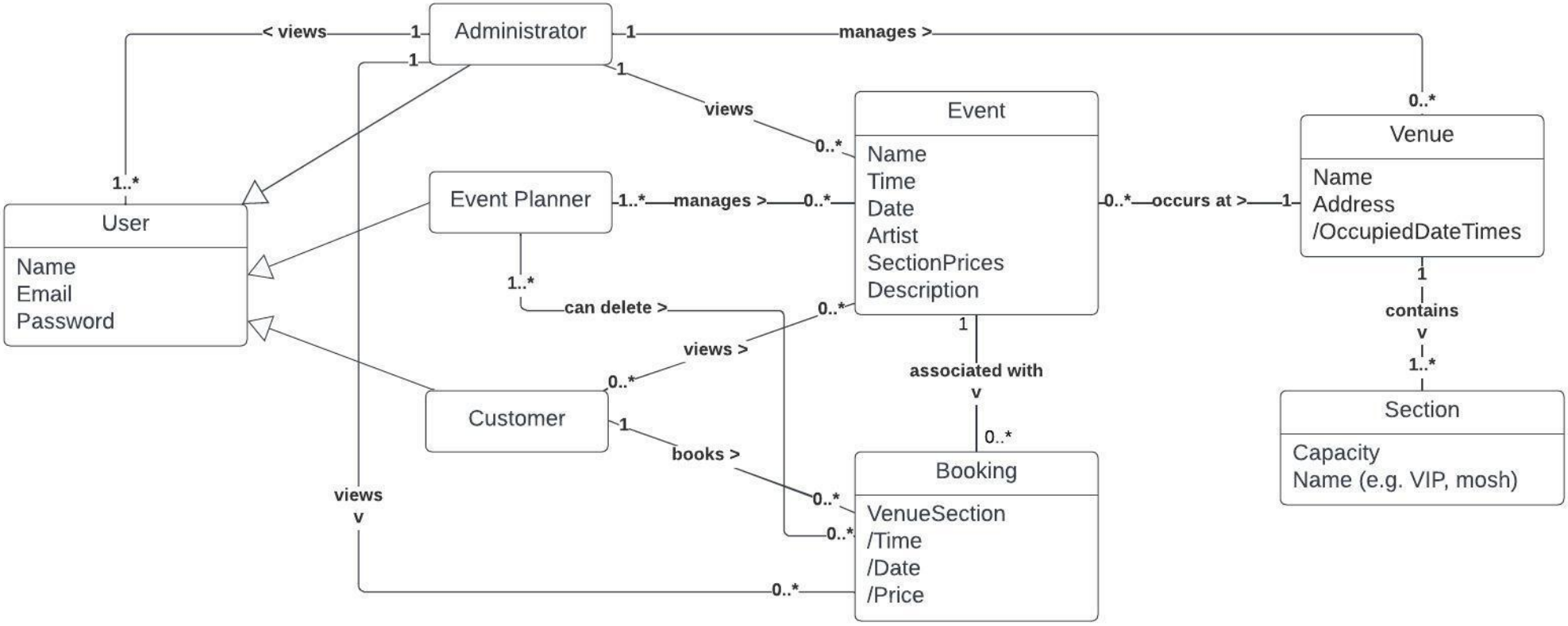
Basic Flow:

The Customer navigates to the Sign Up page, where the System first asks them to select if they want to create an account as a “Customer” or an “Event Planner”. When they select the “Customer” option, the System asks them for their username, email address, password, and a re-typed password confirmation. When the Customer submits this information, the System will show a confirmation message for successful account creation.

Post-Condition:

The Customer is able to log onto the System using their username and password, and be able to view the application as a “Customer”.

Domain Model Diagram



Domain model Explanation

The domain model of the Music Event System consists of the conceptual classes namely the User, the Administrator, the Event Planner, the Customer, the Event and the Booking. It also contains other classes to represent the physical objects namely the Venue and the Section.

The User is the base class for the three actors of the System, which are the Administrator, the Customer, the Event Planner. The attributes of the User class are Name, Email, and Password.

The Administrator class is derived from the User class, so it inherits the attributes from the User class. It can view all Users in the System. They can also view all the Events and their associated Bookings. They are also responsible for managing (i.e. creating, modifying and deleting) Venues.

The Event Planner class is also derived from the User class. It is responsible for creating Events and modifying/deleting the Events it has created. It keeps track of all the Events it has created or has been granted access to. They can also view and cancel all Bookings related to the Events that they manage. Each Event Planner contains one or more events.

The Customer class is also derived from the User class. It can view the Event details and can create Bookings to Events, as well as delete the Bookings they have created. A Customer may keep track of zero or more Bookings.

The Venue class is where Events are located at. It has a name and an address as attributes. It also keeps track of the date and times during which it is occupied (by Events), since a Venue can only host one Event during a particular date and time. Each Venue instance also contains a number of Sections.

The Section class has the following attributes: section capacity and name (e.g. “Mosh”, “VIP”, “Standard-1”, “Standard-2”). Each Section only belongs to one Venue.

The Event class has the following attributes: name, date, time, artist, Venue that it is located at, as well as a mapping from the Venue’s Sections to ticket prices to this Event.

Each Booking instance is a singular ticket to an Event that a Customer has purchased. It records its allocated Section within the Event’s Venue, and it is able to derive the time, date and price from the Event.