Event Management Case Example

Background

You work for a event management company that organizes one-time events. Such events may span multiple days on one or more different event areas. Each event area is part of an event location and may provide a fixed number of seats. An event is organized in several categories, where each category may include other categories.
Customers buy tickets for a specific event and if seat reservation is required, also for specific seat. The ticket is not bound to the person that bought the ticket, so it is fine that one person buys multiple tickets for the same event.
The price of a ticket for an event is based on different price categories which also may differ from event to event. Not all of them require the reservation of seats and therefore seats are not necessarily assigned to price categories.

Models

The repository contains some Papyrus UML Models stored in the files *EventManagement.di*, *EventManagement.notation* and one or more *.uml*-files. The *.di* file contains data needed by the Papyrus editor and usually its content can be ignored. The *.uml* files contain the raw UML model, except diagram information like style, layout data etc. Such information is stored in the diagram model which is located in the *.notation* file. During the interview, the following main UML Models can be found in the repository:

> **Class Diagram:** a class diagram that describes the data stored in our event management system.
>
> **Order Tickets:** An activity diagram that describes the process of ordering a ticket through the ticketing system after the user has selected an event.
>
> **Update Event Process:** An activity diagram that describes the process of updating the properties of an event with the event management system. This activity can be triggered manually or by an external event that requires someone to start actions of this activity. However, in both cases a user has to interact with the forms of this activity.
>
> **Event States:** A state diagram that contains the state(s) a event may have in the event management system.

A change may introduce new models or depend on another base version that may contain additional models.

R3.1 Task “Event Cancellation” - Change 26

Change Id: I9575856e4d58a8bd4be66cb087400db9c5dc1133

Sometimes events get cancelled due to unforeseen reasons. Such events have been simply deleted from the database until now. However, we also want to keep track of such events, so adapt the model accordingly. At least the date of cancellation and the reason for the cancellation must be stored.

C2.1 Task “Event Subscription” - Change 29

Change Id: Ic368415b562368d9de807a2c36ecf1f7c43d8866

We got the feedback that our customers want to be subscribe to some of the event and get notified of changes to the event. For now, adapt the model that customers may subscribe to an event when ordering tickets. Also include the notification of the subscribers in all diagrams if necessary.