Prof. Dr. Peter Thiemann
Stephan Arlt
Martin Wehrle

Sommersemester 2011

Softwaretechnik

`http://proglang.informatik.uni-freiburg.de/teaching/swt/2011/`

Exercise Sheet 6 (2-Week Project)

In this exercise sheet, you are supposed to provide an implementation of a simple event management system (EMS) based on the J2EE middleware architecture. The EMS will provide some of the functionality that has been specified in exercise sheet 3.

For the implementation, you can either download a virtual machine (VM) at `https://sotec.informatik.uni-freiburg.de/swt/XP.ova`[1] that provides the development environment that is needed, or set up the development environment on your local machine. There is also a tutorial document at `https://sotec.informatik.uni-freiburg.de/swt/Tutorial.pdf` that contains an installation guide (how to install the tools on your local machine) and an example project. If you set up the environment on your local machine, make sure to use exactly the same tool versions as provided by the VM and defined in the tutorial document.

Keep everything as simple as possible, but not simpler (Albert Einstein). For your implementation, ensure that the classes only contain the attributes and the methods that are needed to satisfy the requirements. Furthermore, always think about what kind of Enterprise Java Beans (EJB) should be used. In all of the exercises, you are supposed to implement the bean API only (i.e., do not focus on a fancy GUI).

Exercise 1

Download the provided virtual machine or set up the development environment on your local machine. Read the tutorial document.

Exercise 2: Work package *Account* (10 points)

Provide an implementation that satisfies the following requirements. Users can create an account in the EMS. Users can login and logout.

Exercise 3: Work package *Event* (10 points)

Provide an implementation that satisfies the following requirements. Users that are logged in can create new events and register to existing events. By default, the creator of a new event holds the role of an *organizer*, and registering to an event implies the role of a *participant*.

Note: As a starting point for this work package, consider the class diagram from exercise sheet 4, exercise 1, excluding *ConcertSeries*.

[1] Accessible only from the university's network

Exercise 4: Work package *Ticketing I* (10 points)

Provide an implementation that satisfies the following requirements. As a starting point, consider the class diagram from exercise sheet 4, exercise 2, that contains the classes *PrintedTicket* and *ETicket*. A registration to an event implies the purchase of at least one ticket (a participant may buy more than one ticket). Tickets have a price, where electronic tickets can have a discount. Participants can retrieve their purchased tickets (i.e., tickets can be displayed on the screen such that they can be printed by the browser).

Exercise 5: Work package *Ticketing II* (10 points)

Provide an implementation that satisfies the following requirements. Organizers can retrieve the following lists:

- For each event: A list of the registered participants and the total amount of money that has been paid by the participants for the tickets.
- For each registered participant: The number of tickets, the individual price per ticket, and the total amount of money for the tickets that have been bought by the participant.

Note: As a starting point for this work package, consider the class diagram from exercise sheet 4, exercise 2.