Prof. Dr. Stefan Tai
Markus Klems

Your name: ________________

Enterprise Computing (WS 2014)

Exercise 4 (3 Portfoliopunkte)

Info:

- The solution to this exercise must be handed in by Tuesday, Nov 25th 2014, 2PM to Markus Klems.
- The solution must be printed out. Please write your name on the solution sheet.

Task 1 – Java EE Concepts (50%)

a) Your multi-tiered Java EE application needs to integrate with a legacy Enterprise Resource Planning (ERP) system. Which tier in the Java EE application is most suitable for the ERP system?

Solution:

b) Shortly describe in your own words the responsibilities of web tier and business tier.

Solution:

c) Shortly explain: which method of the `HttpServlet` is most suitable for loading initialization parameters that are specified in the deployment descriptor?

Solution:

d) Given the following two Java classes:

```
@Entity
public class Employee {
  @Id
  private long id;
  private String status;
  ...
  @ManyToMany
   private List<Project> projects;
  .....
}

@Entity
public class Project {
  @Id
  private long id;
  ...
  @ManyToMany(mappedBy="projects")
  private List<Employee> employees;
  ...
}
```

How many tables are created in the database to store these entities and their relationship?

Solution:

e) Based on the entities given in e), please write a query using the Java Persistence Query Language that deletes employees from the database which have no projects and who have the status “fired”.

Solution:

Task 2 – Google Cloud Platform (20% + 30% = 50%)

Create a GWT+AppEngine project in Eclipse with generated GreetingService sample code.

a) Add a logging filter servlet that logs every interaction with the service. The logging filter must output the statement “A new greeting!” with the log level “INFO” every time the GreetingService has been called by a user.

Please enter your solution below (your code additions or modifications are sufficient – no need to copy the entire source code):

// web.xml

// LogFilterImpl.java

b) Use JPA to persist each incoming user name in an `Employee` Entity. The Entity has an autogenerated Key and in addition to the name also a hire date which is equal to the time when the server has received the service call. Before saving the new employee, query all existing employees and return them in the response message.

Please enter your code additions or changes below:

// Employee.java

// GreetingServiceImpl.java

References:

- https://cloud.google.com/appengine/docs/java/gettingstarted/introduction
- https://cloud.google.com/appengine/docs/java/datastore/jpa/overview-dn2