Rubrics Unit 2 – Compulsory assignment: Spring

1. List all customers:

- Lists all customers
- Customer DAO getAll() method with Credential null: No transaction
- One query from database: Does not bring extra data
- JDBC and Spring implementation

2. Add a new customer:

- Adds a customer to the tableCustomer and the credential to the Credential table
- Customer DAO add(Customer c) method
- Uses a transaction
- Id of Customer and Credential is the same one
- UserName is unique. Manage exception if not
- Two access to database: Does not bring extra data
- JDBC and Spring implementation

3. Update customer:

- Updates everything but id nor Credentials
- Customer DAO update(Customer c) method with no transaction
- One access to database: Does not bring extra data
- JDBC and Spring implementation

4. Delete customer:

- Deletes a customer with no orders: Does not ask and does not use transaction
- Deletes a customer with orders: Asks the user and deletes orderItems and orders
- Uses a transaction for deleting a customer with orders
- Customer DAO delete(Customer c, boolean) method
- One access to database if no orders: Does not bring extra data
- 4 access to database if orders: Does not bring extra data
- JDBC and Spring implementation

5. Backup XML:

- Backup orders before deleting: Saves into XML file
- OrdersDAOXML save(List<orders> o) method with no transaction
- 1 access to database for getting orders of that specific customer: Does not bring extra data
- One xml file per customer

6. List orders with orderItems & total amount:

- List orders with orderItems (name item, quantity, price)
- Shows total amount
- Orders getAll() method with order items list null : No transaction
- OrderItem get(Order/int order_id) when selecting one order - No transaction
- Does not bring extra data
- JDBC and Spring implementation

7. Add new order:

- Add new order with oder items
- Orders add(Order) method
- Uses transaction
- UI: Order items can be added and removed
- UI: Combo boxes with customer names and table numbers
- Does not bring extra data: Item name with id

8. Update order:

- Update order with order items: Except for id order
- Orders update(Order) method
- UI: Order items can be added and removed
- UI: Combo boxes with customer names and table numbers
- Does not bring extra data: Item name with id

9. Delete order:

- Delete order with order items
- Order delete(Order) method
- Uses transaction

10. Login as administrator:

- All functionality except for adding orders
- id negative: Get all customer selects onlt positive ids

11. Login as a customer:

- A customer can only list, add and update orders
- List only the customer orders
- Add with customer combobox disabled
- Update only the customers orders
- Check username & password and save the id in RAM

12. Friendly and efficient codification:

- Uses standard DAO pattern
- Uses Layers structure
- Uses constants for sql queries and error messages
- Uses property file for database and files configuration
- English: Spelling
- Comments when needed
- ...