

**Figure1: Create object to connect with Database**

**Pattern:** Singleton pattern

**Explain:** Connection is created with a instance**,** avoid for programer open persistent connection and redundant memory.

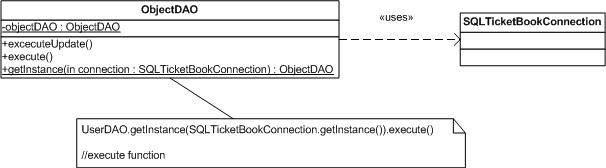
**Usage:**

SysParamTransferData dnew SysParamTransferData(“sql.com.driver…”,”servername:databasename”,”sa”,”123456”);

SQLParentConnection connSQL=SQLParentConnection.getInstance(driver);river=

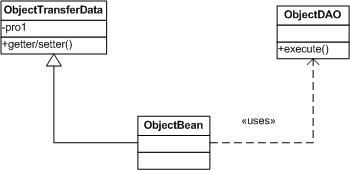
Connection con=connSQL.getConnection()

OR: SQLTicketBookConnection().getInstance().getConnect() ;//use for “ticketbook” database



**Figure 2: ObjectDAO execute database from Connection Input**

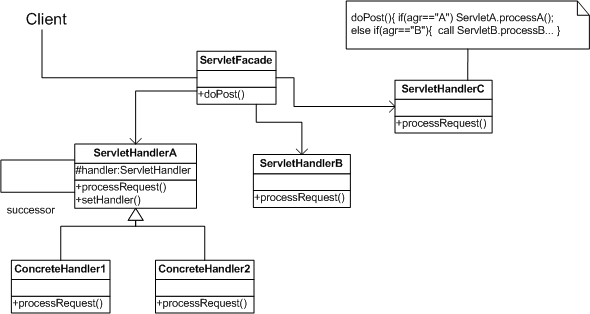
**Explain:** If connection opens, ObjectDAO can execute function on database, help coder easy control the errors connection



**Figure 3: ObjectBean as Adapter for ObjectTransferData**

**Pattern:** Adapter pattern

**Explain:**



**Figure 4: Client send request to ServletFacade, it will call sub servlet to process**

**Pattern:** Facade + Chain Of Responsibility pattern

**Explain:** When user submit, I will create a Servlet Façade accept request and send request to a other servlet to process. If servlet doesn’t process, it will send to a other servlet… Until request is processed.