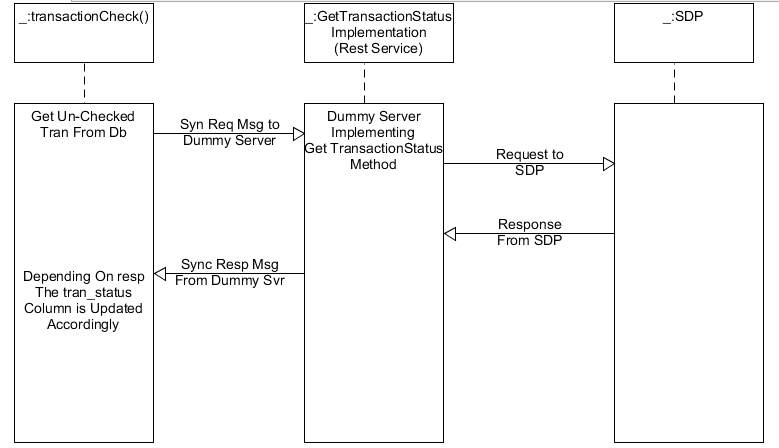
**Get Transaction Status Implementation**

**Baseline Architecture**

****

**Application Description**

**The entire application consists of 4 java classes with one being a servlet. The servelet, getTransactionStatus, is started independently in web container (tomacat v7.0 in this case). It serves as the dummy server to implement the SDP gettransactionstatus method.**

**The other 3 classes, Reader, taskTimer and transactionUpdate, operate in sync as one unit. Execution begins at transactionUpdate class main method, initializes reader class to fetch database connection parameters from a text file stored in the file system and also initializes taskTimer class which ensures the transaction check operation runs at preset intervals until program execution is halted. Initialization of the two classes occurs at transactionUpdate class constructor.**

**public** transactionUpdate (){

Reader reader = **new** Reader();

*date* = **new** Date();

System.***out***.println("--------------------------------New Cycle StartTime :"+**new** Timestamp(*date*.getTime())+"----------------------------------------------------");

timer = **new** Timer();

timer.schedule(**new** taskTimer(), 0, 30 \* 1000); // subsequent rate at 30 seconds

}

**One Database table(transaction table). Is used to hold transactions with the column tran\_status indicating whether the transaction has been confirmed ok (indicated as state 00) ,not (indicated as state 05) or confirmed invalid (indicated as state 100). A transactions state will only be updated if a definite response from the dummy server else no update shall occur**