```
using System;
using System.Data;
using Caissa.Classes;

namespace Caissa.Persistence
{
    public class SystemUnderTestDB
    {

        public int system_id
        {
            get;
            set;
        }

        #region Insert System
        public void Insert(SystemUnderTest system)
        {
            string storedprocedure = "sp_InsertSystem";

            System.Data.IDbConnection connection;
            System.Data.IDbCommand command;

            connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

            command = Mapped.Command(storedprocedure, connection);
            command.CommandType = CommandType.StoredProcedure;
            command.Parameters.Add(Mapped.Parameter("p_sys_title", system.Title,   ↩
              DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sys_view_title",            ↩
              system.ViewTitle, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sys_description",           ↩
              system.Description, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sys_id", "", DbType.Int32,  ↩
              ParameterDirection.Output));
            command.ExecuteNonQuery();

            int id = 0;

            foreach (var parameter in command.Parameters)
            {
                IDbDataParameter p = (IDbDataParameter)parameter;
                if (p.Direction == ParameterDirection.Output)
                    if (p.ParameterName == "p_sut_codigo")
                        id = Convert.ToInt32(p.Value);
            }

            connection.Close();

            system_id = id;
        }
        #endregion

        #region Update System
        public void Update(SystemUnderTest system)
        {
            string storedprocedure = "sp_UpdateSystem";

            System.Data.IDbConnection connection;
            System.Data.IDbCommand command;

            connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

            command = Mapped.Command(storedprocedure, connection);
            command.CommandType = CommandType.StoredProcedure;
            command.Parameters.Add(Mapped.Parameter("p_sys_id", system.Id,
              DbType.Int32, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sys_title", system.Title,
              DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sys_view_title",
              system.ViewTitle, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sys_description",
              system.Description, DbType.String, ParameterDirection.Input));
            command.ExecuteNonQuery();

            connection.Close();
        }
        #endregion

        #region Delete System
        public void Delete(int system_id)
        {
            string storedprocedure = "sp_DeleteSystem";

            System.Data.IDbConnection connection;
            System.Data.IDbCommand command;

            connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

            command = Mapped.Command(storedprocedure, connection);
            command.CommandType = CommandType.StoredProcedure;
            command.Parameters.Add(Mapped.Parameter("p_sys_id", system_id,
              DbType.Int32, ParameterDirection.Input));
            command.ExecuteNonQuery();

            connection.Close();
        }
        #endregion

        #region Select System
        public DataSet Select()
        {
            System.Data.IDbConnection connection;
            System.Data.IDbCommand command;
            System.Data.IDataAdapter adapter;

            DataSet ds = new DataSet();

            string sql = string.Empty;

            connection = Mapped.Connection(Mapped.conexao.ServerExplorer);
            sql = "select sys_id, sys_title, sys_view_title, sys_description"
                + " from tbl_system"
                + " where sys_active = 0;";

            command = Mapped.Command(sql, connection);

            adapter = Mapped.Adapter(command);
            adapter.Fill(ds);

            connection.Close();

            return ds;
        }
        #endregion

        #region Select Especified System
        public DataSet SelectSystem(int system_id)
        {
            System.Data.IDbConnection connection;
            System.Data.IDbCommand command;
            System.Data.IDataAdapter adapter;

            DataSet ds = new DataSet();

            string sql = string.Empty;

            connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

            sql = "select sys_id, sys_title, sys_view_title, sys_description"
                    + " from tbl_system"
                    + " where sys_id = ?system_id"
                    + " and sys_active = 0;";

            command = Mapped.Command(sql, connection);
            command.Parameters.Add(Mapped.Parameter("?system_id", system_id));

            adapter = Mapped.Adapter(command);
            adapter.Fill(ds);

            connection.Close();

            return ds;
        }
        #endregion

        public SystemUnderTestDB()
        {
            //
            // TODO: Add constructor logic here
            //
        }
    }
}
```