```
            System.Data.SqlClient.SqlCommand sqlCmd = new System.Data.SqlClient.SqlCommand( ↙
    "spPostMsgAttachment", sqlConn);
            sqlCmd.CommandType = System.Data.CommandType.StoredProcedure;

            // Add our parameters to the stored procedure
            sqlCmd.Parameters.Add("@sessionID", SqlDbType.Int).Value = sessionID;
            sqlCmd.Parameters.Add("@messageID", SqlDbType.VarChar).Value = messageID;
            sqlCmd.Parameters.Add("@mediaType", SqlDbType.VarChar).Value = mediaType;
            sqlCmd.Parameters.Add("@mediaLocation", SqlDbType.VarChar).Value = mediaLocation;

            // Execute the stored procedure
            sqlCmd.ExecuteScalar();
        }

        /// <summary>
        /// Gets the user name associated with a session ID (this is used mostly for file     ↙
    storage)
        /// </summary>
        /// <param name="sessionID">The user's current session ID</param>
        /// <returns>The user's user name</returns>
        public String GetUserFromSession(long sessionID)
        {
            // The stored procecure we're executing
            System.Data.SqlClient.SqlCommand sqlCmd = new System.Data.SqlClient.SqlCommand( ↙
    "spGetUserFromSession", sqlConn);
            sqlCmd.CommandType = System.Data.CommandType.StoredProcedure;

            // Add our parameters to the stored procedure
            sqlCmd.Parameters.Add("@sessionID", SqlDbType.Int).Value = sessionID;

            // Execute the stored procedure
            System.Data.SqlClient.SqlDataReader reader = sqlCmd.ExecuteReader();

            reader.Read();

            return reader["UserName"].ToString();
        }

        /// <summary>
        /// Clears all expired sessions from the database
        /// </summary>
        private void ClearExpiredSessions()
        {
            // The stored procedure we're executing
            System.Data.SqlClient.SqlCommand sqlCmd = new System.Data.SqlClient.SqlCommand( ↙
    "spClearExpiredSessions", sqlConn);
            sqlCmd.CommandType = System.Data.CommandType.StoredProcedure;

            // Execute the stored procedure
            sqlCmd.ExecuteScalar();
        }

        /// <summary>
        /// Verifies SQL input to protect against injection attacks.
        /// </summary>
        /// <param name="sql">The SQL variable we'll be cleaning</param>
        private String CleanSQL(String sql)
        {
            // Make sure our SQL string is alphanumeric
            if (reg.IsMatch(sql))
            {
                return sql;
            }
            else {
                return "";
            }
        }
    }
}
```