UNIVERSITY OF
BIRMINGHAM

# Software Workshop 2

—

# Real Time Multi-User Quiz System

Benjamin Crispin,
Samuel Farmer,
Deedar Fatima,
Rowan Stringer,
Josh Wainwright

**Group Osaka**

*Supervisor:* Joe Gardiner

*Date:* March 2014

# Contents

# 1 Protocol

The protocol for communicating between the different parts of the system is based around objects. There exists an object class that can be created for any of the several message types that could be needed to transfer information from the server to the client, or from the client to the server. Each of these objects implements the `Serializable` interface, allowing them to be converted to bytes and transfered across the socket connection.

**AnswerResponse object**
To respond to a question, the Student selects the desired option. This information is passed to the server using an AnswerResponse object which simply holds the response and the timestamp to indicate when they made the selection.

**DisplayQuestion object**
In order to signify that the allotted time for the current question has ended, and the next question should be displayed, the server sends a DisplayQuestion object to all of the clients and they should move on to the next question in the Quiz object and change the GUI accordingly.

**LoginReply object**
Once a loginRequest has been received by the server, a LoginReply will be sent back. This gives the client the information about the requested login, most importantly if the login was successful, as well as the type of user that made the login, Student or Admin. This information is used to display the correct user interface.

**LoginRequest object**
This is the first object that could be created. It is sent, by the client, to the server and contains the username of the Student that is attempting to login and the `java.lang.String` hash code of the inserted password. Though the security concerns of such a trivial system are non-existent, the password is never stored in plaintext.

**Question object**
There exist several question objects in each Quiz object. They store the information required to present a Student with a question and the possible answers. Again, there is very little functionality as the question only serves as a wrapper for the question text and the possible answers that the user could respond with.

**Quiz object**
This is the most important object. It has very little functionality, simply acting as a wrapper to hold and easily transfer several Question objects.

**QuizRequest object**

**Score object**
When the quiz has been completed, each of the clients will display the position of it's user relative to the other Students. This object contains the score of all of the users so that each of the clients can work out where they are in the ranking.

**StartQuiz object**
Once the user has successfully logged into the system, the next major event is the start of the quiz. This is signaled by an Admin user who is connected to the server, this information must then be relayed to each of the connected clients. A StartQuiz object is sent to each of the clients, who, on receiving it, will display the first question to the user.

# 2 Client

The client exists to accept messages sent by the server, and present them in an order and a format that the user interface can present to the user, as well as accept the information entered by the user into the user interface and pass it to the server.

## 2.1 System Design

When a client is started, a client object is created and starts the main loop. The initial stages set up the connection with the server and waits for the user to login.

# References