Проектиране и интегриране на софтуерни системи

Coursework: Distributed Application

Summary

This is an individual coursework worth 40% of the total module mark. It requires developing of a prototype of a distributed system for exchanging information over chosen network protocol between clients, servers and/or peers. It completes with a written report to be submitted by the deadline and demonstration of the software in a computer Lab.

Tasks

1. **Design an application to develop picking up one of the following alternative choices**
   a) Choice of one of the following architectures to be designed: **client/server, peer-to-peer** or **mixed** (***Examples:*** entirely peer-to-peer chat application which allows direct chatting between the users, client/server chat application which uses central server to exchange the messages between the users, or a chatroom, which allows to use private rooms for exchanging messages as well as common meeting room, etc.)
   b) Choice of one or more of the following data structures to be used for exchanging information in the distributed application: **text messages, multi-media messages**, **text files, binary files, text feeds, multi-media streaming** (***Examples:*** text messaging with binary file attachments, multimedia messaging, uploading/downloading binary files to/from the server, broadcasting multimedia messages with file attachments, etc.)
   c) Choice of one of the following protocols to be used: **TCP**, **UDP, HTTP, IIOP** or **RTP** (***Examples:*** TCP for desktop messaging application with attachments, UDP for multimedia broadcasting over the Internet, HTTP for instant chatting on the Web, etc.)
   d) Choice of one of the following interfaces to be used: **command prompt input/output** (using `System` package), **Swing-based input/output** (using `javax.swing` and `java.awt` packages), **Web form-based input/output** (using `javax.servlet`), **Applet-based** input/output (using `java.applet`) or any applicable combination (***Examples:*** applet for the client and servlet for the server, window-based client and command prompt interface for the server, browser-based peer-to-peer application)
2. **Develop software to match your design**
   a) Develop the basic server-side application, test it using suitable test client and create deployable version. (***Hint:*** Place all files of the client in a single Eclipse or NetBeans project).
   b) Develop the client-side application and test it using the server-side. (***Hint:*** Create a separate project for the client application, different from the server application project).
   c) Extend the server-side and the client-side applications to allow any additional functionality whenever applicable (***Hint:*** Add multi-threading, graphical UI, password control, attachments, operations log file, conversation thread, etc.)
3. **Test the system**

   Some tests can be already reported as part of the development process (i.e,, unit tests), but this section should include the tests of the separate components (integration testing) and the entire application (functional testing)
4. **Write the report**

   The report structure should follow the process of working - titlepage, table of contents, introduction, design, development, deployment, testing, application walktrough, conclusion.

Requirements

1. Your choices must be registered with the teaching team by Week 4. Your submission must match the registered choices.
2. The deployed software should run using the Java runtime engine only and the demonstration must be done using the deployed version, not using the development tools and the projects (i.e., no Eclipse, NetBeans, etc). All non-standard libraries must be included in the archives
3. The written report must provide information on the following issues:
    - **Application design** with suitable illustration of the software architecture (at least one UML diagram – package, component, deployment or mixed).
    - **Program Development** (at least one class/package/component diagram for the structure and one collaboration/sequence/statechart diagram for the functionality).
    - **System Deployment** (how to install, configure and run the application).
    - **Software and/or System Testing** with suitable test cases (test strategy, testing scenarios, test data, test results and analysis)
    - **Application Walkthrough** of the working application (set of screen dumps from the client and the server)
4. In the demonstration you can use either a desktop PC in the Lab, or your own laptop. The demonstration will not be marked, but if the application is not demonstrated the marks for the software will be set to 0 and only the report will be marked.

Deliverables

1. Written report, submitted by the deadline (one .pdf file)
2. Java application, submitted by the deadline (Eclipse ir NetBeans project plus deployment files in .jar and/or .war format in a single .zip file)
3. Demonstration, presented during the workshops in Week 12 (using university computer or laptop)

Marking Scheme

• Basic server-side functionality	up to 4 marks
• Basic client-side functionality	up to 4 marks
• Additional functionality	up to 16 marks
✓ Multi-threading	
✓ Multi-part communication and/or attachments	
✓ User virtualization (chatrooms, file folders, conversation trails, etc.)	
✓ Operation logging	
• Design, Development and Testing	up to 8 marks
• Deployment, Documentation and Quality	up to 8 marks
Total	**max 40 marks**

Notes:

1. *All files must be submitted in a single .zip archive (.arj, .rar or any other formats are not supported and using them may cause the entire submission to be discarded)*
2. *The demonstration is not marked but without demonstration the mark for the software will be set to 0 and only the report will be marked with max mark 40% (practically fail)*
3. *The report is not compulsory but without it the max mark will be 60% (no first class)*
4. *The basic functionality may be sufficient to pass but without additional functionality the max mark will be 60% (no first class)*