|  |  |  |  |
| --- | --- | --- | --- |
|  | **max** |  | **Marking Notes** |
| Basic application functionality | 20 |  | How well the basic requirement are met, ie that the application functions in client-  server mode; and that pairwise chat is possible, tested on 2 separate terminals. The application must use UDP sockets, otherwise half the marks. |
| Creative & additional features | 15 |  | Creativity beyond required specification, e.g authentication, encryption, group chat.. |
| Robust protocol implementation (stress tests) | 10 |  | Implementation should include an internal simulator that introduces unreliability (e.g.  simulating that 10% of the messages are lost or erroneous). Stress the application, e.g  by having a client disconnect and reconnect while it is in a chat session? For some  error simulation and handling, 5 marks, for sufficient to excellent error simulation/  handling give between 6 and 10 marks. |
| Code quality | 10 |  | Quality of code, in terms of readability, efficiency. Check that the implementation is  actually client/server using UDP Sockets. (if this is not the case,  0 marks in this  category, and -half the marks  under 'Basic application functionality). |
| Reliability constraints considered in protocol design/implementation | 10 |  | Consideration for reliability - e.g message validation and loss recovery |
|  |  |  |  |
| Report: Description of system functionality and features | 10 |  | Description clearly explains what/how the chat is implemented. |
| Report: Specification (Protocol design & specification (sequence diagrams & message formats/structure)) | 15 |  | Sequence diagrams (8 marks), message formats (7 marks).  Sequence diagrams need  to indicate the different states that communicating parties (clients, server) would be  in at different times (eg. Listening, Connected, FileTransfer, etc), and the types of  messages/events that would move the clients/servers into different states (eg client  receives a "CHAT" type message from a server, or server receives a "BYE" type  message from client). The diagrams should show the sequence and messages  between servers and clients in different states,  and reactions to the messages.  Protocol messages to constitute at least the header and body, and may include  message type, the command, and recipient information.  Implementations will vary,  but descriptions should make sense, be clear, and reflect the implementation. |
| Report: Overall quality of report | 10 |  | Clarity, language, format... Also indicate the group members and individual contribution. |
| **Total** | 100 |  |  |
|  |  |  |  |
| student id |  |  |  |
| student id |  |  |  |
| student id |  |  |  |
|  |  |  |  |
|  |  |  |  |