ImgClient.java

```java
package acsse.csc2b.client;

import javafx.application.Application;

public class ImgClient  extends Application {

    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws Exception {
        ImgClientPane root = new ImgClientPane(primaryStage);
        Scene scene = new Scene(root, 800, 600);
        primaryStage.setScene(scene);
        primaryStage.setTitle("Image Client");
        primaryStage.show();
    }
}

```

ImgClientPane.java

```
  1 package acsse.csc2b.client;
  2
  3 import java.io.BufferedReader;
 28
 29
 30
 31 public class ImgClientPane extends GridPane {
 32
 33     private Socket socket;
 34     private InputStream is;
 35     private OutputStream os;
 36     private PrintWriter pw;
 37     private BufferedReader br;
 38     private DataInputStream dis;
 39     private DataOutputStream dos;
 40     private String[] listData;
 41
 42     //GUI:
 43     private Button btnConnect;
 44     private Button btnPULL;
 45     private TextField txtIDToRetrieve;
 46     private Label lblID;
 47     private Button btnDownload;
 48     private Button btnUpload;
 49     TextArea listArea;
 50     private TextArea responseArea;
 51     private Label lblList;
 52     private Label lblResponse;
 53     private ImageView imgView;
 54     private Button btnDisplay;
 55     private String fileToGetName ="";
 56     public ImgClientPane(Stage stage) {
 57         setupUI();
 58
 59         btnConnect.setOnAction((e)->{
 60             try {
 61                 socket = new Socket("localhost",9876);
 62                 os = socket.getOutputStream();
 63                 is = socket.getInputStream();
 64                 br = new BufferedReader(new InputStreamReader(is));
 65                 pw  =new PrintWriter(os);
 66                 dis = new DataInputStream(is);
 67                 dos = new DataOutputStream(os);
 68             } catch (IOException e1) {
 69                 // TODO Auto-generated catch block
 70                 e1.printStackTrace();
 71             }
 72         });
 73
 74         //5 marks - requesting image list
 75         btnPULL.setOnAction((e)->{
 76             sendCommand(pw, "PULL");
 77             String response = "";
 78
 79             response = readResponse(br);
 80             System.out.println(response);
 81             listData = response.split("#");
```

ImgClientPane.java

```
 82
 83             for(int i = 0;i<listData.length;i++)
 84             {
 85                 listArea.appendText(listData[i]+"\n");
 86             }
 87
 88         });
 89
 90         //Retrieving image file - 10 marks
 91         btnDownload.setOnAction((e)->{
 92             int idToRetrieve = Integer.parseInt(txtIDToRetrieve.getText());
 93             pw.println("DOWNLOAD "+idToRetrieve);
 94             pw.flush();
 95             String response = "";
 96             //Server to respond with the file size and file:
 97             try {
 98                 response  = br.readLine();
 99                 int fileSize = Integer.parseInt(response);
100                 responseArea.appendText("File Received Size: " +response);
101                 //get the file name from the list data
102
103                 for(String s:listData)
104                 {
105                     StringTokenizer tok = new StringTokenizer(s);
106                     String id = tok.nextToken();
107                     String name = tok.nextToken();
108                     if(id.equals(txtIDToRetrieve.getText()))
109                     {
110                         fileToGetName = name;
111                     }
112                 }
113                 File fileDownloaded = new File("data/client/"+fileToGetName);
114                 FileOutputStream fos = null;
115
116                 fos = new FileOutputStream(fileDownloaded);//to write to the file
117                 byte[] buffer = new byte[1024];
118                 int n = 0;
119                 int totalBytes = 0;
120                 while(totalBytes!=fileSize)
121                 {
122                     n = dis.read(buffer, 0, buffer.length);
123                     fos.write(buffer,0,n);
124                     fos.flush();
125                     totalBytes+=n;
126                 }
127
128                 System.out.println("File saved on client side");
129                 }
130             catch (IOException e1) {
131                 // TODO Auto-generated catch block
132                 e1.printStackTrace();
133             }
134         });
135
136
137         btnUpload.setOnAction((e)->{
138             //10 Marks: Uploading an image file
```

ImgClientPane.java

```
139             FileChooser fileChooser = new FileChooser();
140             File selectedFile = fileChooser.showOpenDialog(stage);
141             String fileName = selectedFile.getName();
142             int fileID = listData.length +1;
143             pw.println("UPLOAD "+ fileID +" "+ fileName+ " "+selectedFile.length());
144             pw.flush();
145             System.out.println("Upload command sent from client");
146 
147             FileInputStream fis;
148             try {
149                 fis = new FileInputStream(selectedFile);
150                 byte[] buffer = new byte[1024];
151                 int n = 0;
152                 while((n = fis.read(buffer))>0)//read file into byte[]
153                 {
154                     dos.write(buffer,0,n); //write the buffer on dataoutputstream
155                     dos.flush();
156                 }
157                 fis.close();
158                 System.out.println("File sent for upload to server");
159                 String response = br.readLine();
160                 responseArea.appendText("Status of uploaded file: "+ response);
161 
162             } catch (FileNotFoundException e2) {
163                 // TODO Auto-generated catch block
164                 e2.printStackTrace();
165             }catch(IOException e1)
166             {
167                 e1.printStackTrace();
168             }
169         });
170 
171         //Display of image file: 5 marks
172         btnDisplay.setOnAction((e)->{
173             Image image = new Image("file:data/client/"+fileToGetName);
174             ImageView imgView = new ImageView();
175             imgView.setImage(image);
176             add(imgView, 0, 7,4,1);
177 
178         });
179     }
180 
181     private void setupUI()
182     {
183         setHgap(10); //Horizontal spacing between elements
184         setVgap(10); //Vertical spacing between elements
185         setAlignment(Pos.CENTER);
186         btnConnect = new Button("Connect");
187         btnPULL = new Button("PULL");
188         txtIDToRetrieve = new TextField();
189         lblID = new Label("File ID to retrieve:");
190         btnDownload = new Button("DOWNLOAD");
191         btnUpload = new Button("UPLOAD");
192         listArea = new TextArea();
193         listArea.setPrefHeight(50);
194         responseArea = new TextArea();
195         responseArea.setPrefHeight(50);
```

ImgClientPane.java

```
196         lblList = new Label("List: ");
197         lblResponse = new Label("Server Response:");
198         btnDisplay = new Button("Display Downloaded image");
199 
200 
201         add(btnConnect,0,0);
202         add(btnPULL, 1, 0);
203 
204         add(lblID, 0, 1);
205         add(txtIDToRetrieve, 1, 1);
206         add(btnDownload, 2, 1);
207         add(btnUpload, 3, 1);
208         add(lblList, 0, 2);
209         add(listArea, 0, 3,4,1);
210         add(lblResponse, 0, 4);
211         add(responseArea, 0, 5,4,1);
212         add(btnDisplay,0,6,4,1);
213 
214     }
215 
216     private String readResponse(BufferedReader br)
217     {
218         String response = "";
219         try {
220             response= br.readLine();
221         } catch (IOException e) {
222             // TODO Auto-generated catch block
223             e.printStackTrace();
224         }
225         return response;
226     }
227 
228     private void sendCommand(PrintWriter pw, String msg)
229     {
230         pw.println(msg);
231         pw.flush();
232     }
233 
234 
235 }
236 
```

ImgServer.java

```java
package acsse.csc2b.server;

import java.io.IOException;

public class ImgServer {

    private ServerSocket server;
    private boolean running;

    public ImgServer(int port) {
        try {
            server =  new ServerSocket(port);
            running = true;
            startServer();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

    //5 marks for multi-threaded server
    private void startServer() {
        System.out.println("Starting the server...");
        while(running)
        {
            try {
                Socket incomingConn = server.accept();
                System.out.println("New Client Connected");
                ImgHandler imgHandler = new ImgHandler(incomingConn);
                Thread t = new Thread(imgHandler);
                t.start();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }

    public static void main(String[] args) {
        ImgServer s = new ImgServer(9876);
    }
}

```

ImgHandler.java

```
 1 package acsse.csc2b.server;
 2
 3 import java.io.BufferedReader;
20
21 public class ImgHandler implements Runnable {
22     private Socket incomingConnection;
23     private OutputStream os;
24     private InputStream is;
25     private PrintWriter pw;
26     private BufferedReader br;
27     private DataOutputStream dos;
28     private DataInputStream dis;
29
30     public ImgHandler(Socket s) {
31         this.incomingConnection = s;
32         try {
33             os = incomingConnection.getOutputStream();
34             is = incomingConnection.getInputStream();
35             pw = new PrintWriter(os);
36             br = new BufferedReader(new InputStreamReader(is));
37             dos = new DataOutputStream(os);
38             dis = new DataInputStream(is);
39
40         } catch (IOException e) {
41             // TODO Auto-generated catch block
42             e.printStackTrace();
43         }
44
45     }
46
47     @Override
48     public void run() {
49         System.out.println("Handling Client Requests");
50         boolean processing = true; // handling commands for this client's session
51
52         try {
53             while(processing)
54             {
55                 String message = br.readLine();
56                 System.out.println("Message: "+ message);
57                 StringTokenizer st = new StringTokenizer(message);
58                 String command = st.nextToken().toUpperCase();
59                 switch (command) {
60                 case "PULL": // 10 Marks:
61                     pw.println(loadImgList());
62                     pw.flush();
63                     break;
64                 case "DOWNLOAD"://15 Marks Server returning requested image
65                     String fileID = st.nextToken();
66                     System.out.println("ID requested: "+fileID);
67                     String fileName = "";
68                     //Now read through the data to find the matching ID's fileName.
69                     File fileList = new File("data/server/ImgList.txt");
70                     Scanner sc = new Scanner(fileList);
71                     String line ="";
72                     while(sc.hasNext())
73                     {
```

ImgHandler.java

```
 74                          line = sc.nextLine();
 75                          StringTokenizer tokenizer = new StringTokenizer(line);
 76                          String id = tokenizer.nextToken();
 77                          String fName = tokenizer.nextToken();
 78                          if(id.equals(fileID))
 79                          {
 80                              fileName = fName;
 81                          }
 82                      }
 83                      sc.close();
 84
 85                      System.out.println("Name of the requested file:"+fileName);
 86                      File fileToReturn = new File("data/server/"+fileName);
 87                      if(fileToReturn.exists())
 88                      {
 89                          pw.println(fileToReturn.length());//send the file's size to the client
 90                          pw.flush();
 91                          FileInputStream fis = new FileInputStream(fileToReturn);
 92                          byte[] buffer = new byte[1024];
 93                          int n = 0;
 94                          while((n=fis.read(buffer))>0)
 95                          {
 96                              dos.write(buffer,0,n);
 97                              dos.flush();
 98                          }
 99                          fis.close();
100                          System.out.println("File sent to client.");
101
102
103                      }
104                      break;
105
106                  case "UPLOAD": //15 Marks: Uploading of an image
107                      //Command structure received: UPLOAD <ID> <NAME> <SIZE> <IMAGE>
108                      //file to receive details
109                      String fileRecID = st.nextToken();
110                      String fileRecName = st.nextToken();
111                      //NOTE protocol to send size a 3rd parameter.
112                      int size = Integer.parseInt(st.nextToken());
113
114
115
116
117                      PrintWriter pout = new PrintWriter(new BufferedWriter(new FileWriter
   ("./data/server/ImgList.txt",true)));
118                      pout.println(fileRecID +" "+ fileRecName);
119                      pout.flush();
120                      pout.close();
121                      System.out.println("File appended to list");
122
123                      File fileToRec = new File("data/server/" + fileRecName);
124                      FileOutputStream fos = null;
125                      System.out.println(" Still Receiving bytes from client...");
126                      try
127                      {   fos =new FileOutputStream(fileToRec);
128                          byte [] buffer = new byte[1024];
129                          int n =0;
```

ImgHandler.java

```
130                         int totalbytes = 0;
131                         while(totalbytes !=size)
132                         {
133                             n = dis.read(buffer, 0, buffer.length);
134                             fos.write(buffer,0,n);
135                             fos.flush();
136                             totalbytes+=n;
137                         }
138 
139                         pw.println("HAPPY");
140                         pw.flush();
141                         System.out.println("DONE File uploaded to server");
142 
143                     }
144                     catch(IOException e)
145                     {
146                         pw.println("SAD");
147                         pw.flush();
148                         e.printStackTrace();
149                     }
150                     finally
151                     {
152                         if(fos != null)
153                         {
154                             try {
155                                 fos.close();
156                             }
157                             catch(IOException e)
158                             {
159                                 e.printStackTrace();
160                             }
161                         }
162                     }
163                     break;
164                 }
165 
166             }
167 
168         }
169         catch(IOException e)
170         {
171             e.printStackTrace();
172         }
173 
174     }
175     private String getFileNameFromID(String searchID)
176     {
177         String ret = "";
178         File imglist = new File("data/server/ImgList.txt");
179         try {
180             Scanner sc = new Scanner(imglist);
181             String line = "";
182             while(sc.hasNext())
183             {
184                 line = sc.nextLine();
185                 StringTokenizer st = new StringTokenizer(line);
186                 String id = st.nextToken();
```

ImgHandler.java

```
187                 String fname = st.nextToken();
188                 if(id.equals(searchID))
189                 {
190                     ret = fname;
191                 }
192             }
193             sc.close();
194 
195         } catch (FileNotFoundException e) {
196             // TODO Auto-generated catch block
197             e.printStackTrace();
198         }
199 
200         return ret;
201     }
202 
203     //Marks for Returning of image files list
204     private String loadImgList() {
205         String ret = "";
206         try {
207         Scanner sc = new Scanner(new File("./data/server/ImgList.txt"));
208         while(sc.hasNextLine())
209         {
210             String img = sc.nextLine();
211             ret += img + "#";
212         }
213         System.out.println("Image list loaded");
214         sc.close();
215         }catch (FileNotFoundException e)
216         {
217             e.printStackTrace();
218         }
219         return ret;
220     }
221 
222 }
223 
```