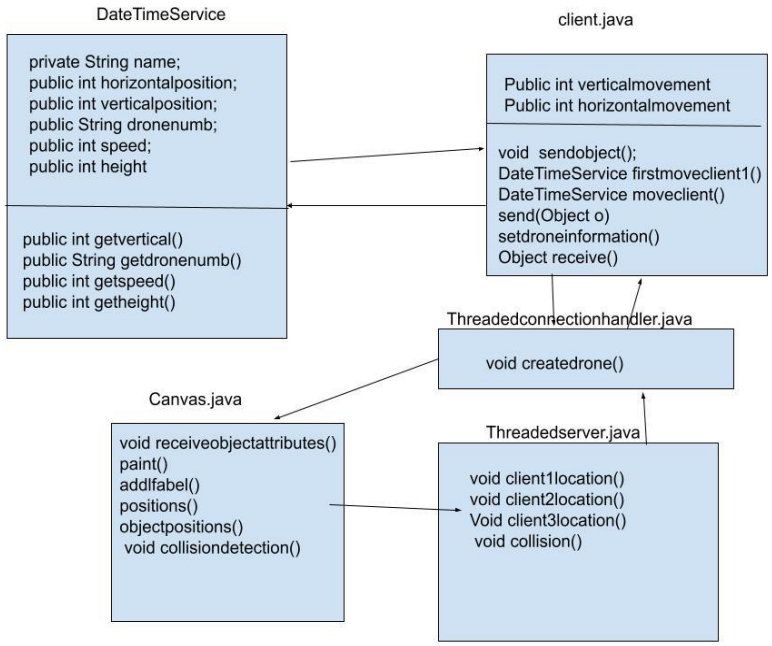
EE402 Assignment 2 Report

Overall Design

Figure below shows the workflow of classes.



Workflow and feature

*Sorry for inappropriate class name due to limited time have I didn't want to ruin the working code by changing class name.

DateTimeService object is created in client.java with some arguments passed in command line after creating object is sent to the connection handler

Received object in Threadedconnectionhandler sent each object feature to the canvas.java and Threadedserver.java

Canvas.java updates the gui in Threadedserver

Object sent back to the Threadedserver

Received object in client.java if any changes made, updates the object and sent to same object with new features to Threadedconnectionhandler.

Implementation/coding

Server gui Implemented features

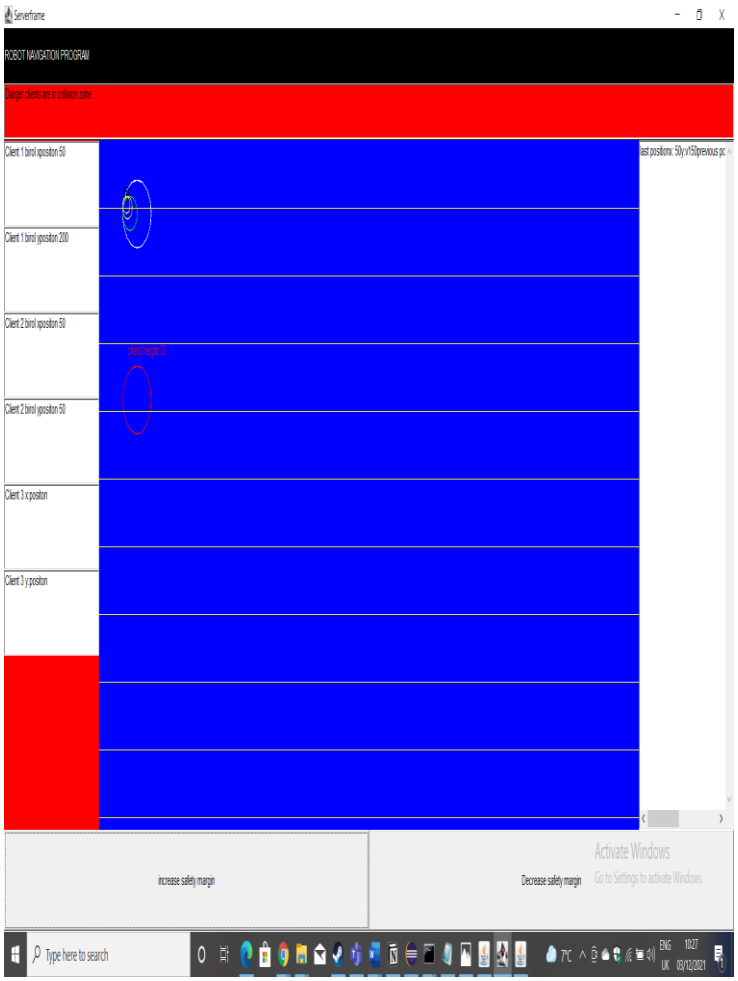
- **On execution opening server socket and waiting connections**
- **(X,Y graphical grid)**
- **Server accepting three connections**
- **Showing related properties of clients**
- **Showing the client name and height on mouse click related area to that drone, and deletes if clicked on another area.**
- **Collision distance is calculated based on distance and margin**
- **Margin is configurable**
- **Showing the previous 3 position with circles descending in size in proportional the number of previous moves as well as in textArea**
- **No novel design**

• Server GUI

- **Server starting point no client has been connected.**
- **GUI frame created as an Border Layout. Includes 5 different NORTH,SOUTH,CENTER,EAST,WESTpanels**
- **North panel shows if any drones are in collision zone**
- **East panel shows x,y locations of client drones**
- **West panel shows the previous locations of drones**
- **South panel allow to change safety zone area**
- **Figure on below one client and its previous 3 position with circles descending in size in proportional the number of previous moves**

- **Figure left shows Server Gui with multiple clients that have more distance between them than the collision area**
- **Figure right shows how north panel informs that two drones are in collision zone and changes background colour to red .**
- **(drones are in same position margin is increased)**

- 
Serverframe
ROBOT NAVIGATION PROGRAM
clients are not in collision zone. currentMargin:50
Client 1 birol xposition 50
Client 1 birol xposition 200
Client 2 birol xposition 50
Client 2 birol xposition 50
Client 3 x positon
Client 3 y positon
last positions: 50,y:1280previous p
increase safety margin
Decrease safety margin
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
UK
10:25
03/12/2021

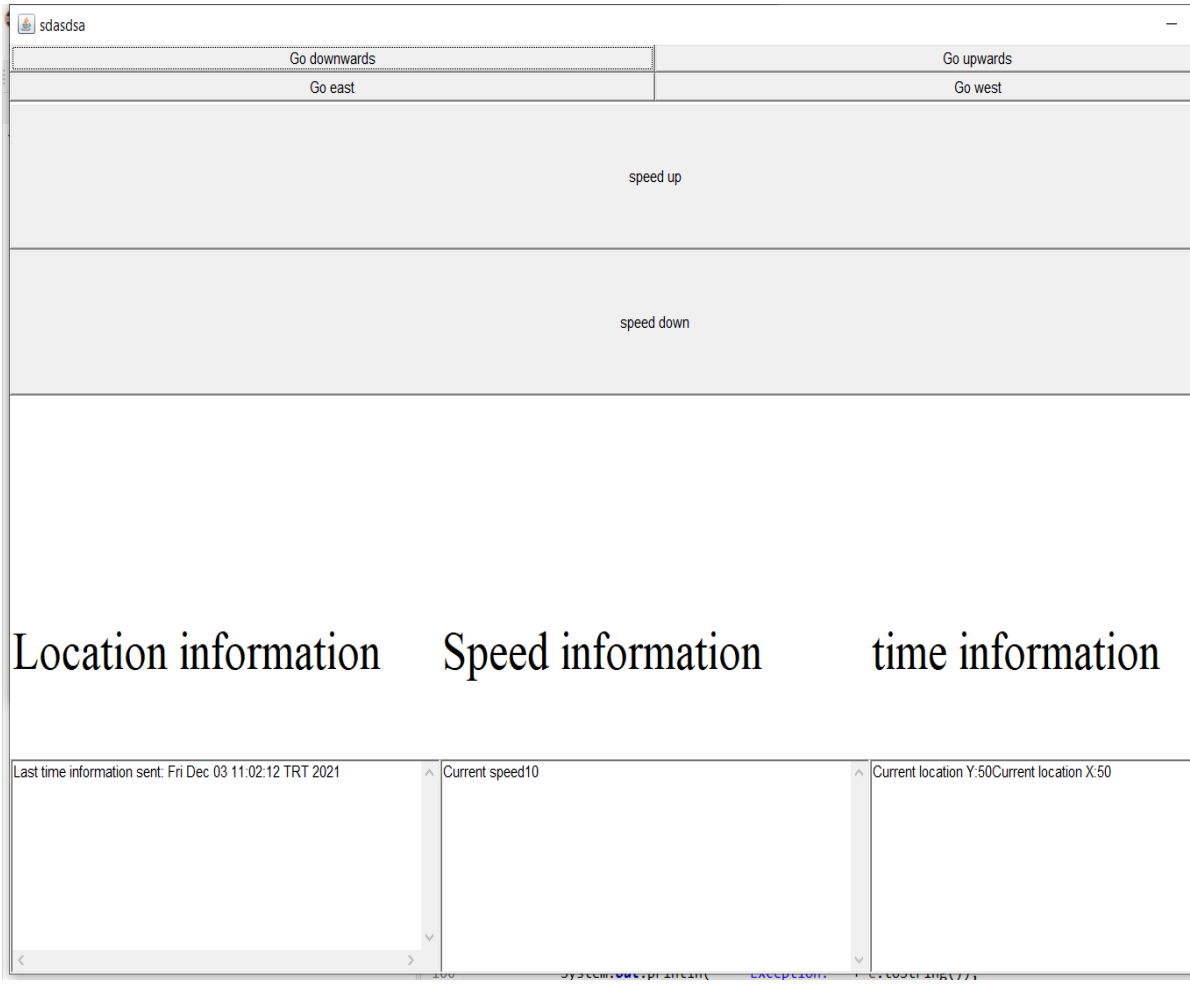


Client GUI

- **Server address, robot name is passed in command line argument also extra number should be passed indicating the client number (1,2,3)**

```
03. -- About to receive an object...
04. <- Object received...
05. <- The Server responded with:
    <- assignmentss.DateTimeService@6c3708b3
01. -> Sending Command ( DateTimeService  ) to the server...
^C
C:\Users\DEO\assignments\assignmentss\bin>java assignmentss.client localhost birol 1
```

- **Gui buttons allowing movements of objects and last time object, current speed and location that sent is displayed**
- **Client sends object to server every 10 second**
- **As an novel feature speed is added , as speed changes the distance moved by object is changed aswell.**



Coding

- **Server Gui implementation part by part**

```
        Panel Main1 = new Panel(new GridLayout(1,1)); //other panels implemented here
        Panel East = new Panel(new GridLayout(8,2));
        Panel East2 = new Panel(new GridLayout(1,1));
        Panel South = new Panel(new GridLayout(1,3));
        East.setPreferredSize(new Dimension(200,10));
        South.setPreferredSize(new Dimension(200,200));
        East.setBackground(Color.RED);
        this.add(East,BorderLayout.WEST);
        this.add(East2,BorderLayout.EAST);
        this.add(South,BorderLayout.SOUTH);



        label2 = new TextField("Client 1 x:position");
        label3 = new TextField("Client 1 y:positon");
        label4 = new TextField("Client 2 x:position");
        label5 = new TextField("Client 2 y:position");
        label6 = new TextField("Client 3 x:positon");  //East part implemented here
        label7 = new TextField("Client 3 y:positon");
         East.add(label2);
        East.add(label3);
        East.add(label4);
        East.add(label5);
        East.add(label6);
        East.add(label7);

        drone1 = new TextArea();        //west part implemented here
        drone1.setText("Drone current position"+" previous position "+ "before previous");
        East2.add(drone1);



        incmargin = new Button("increase safety margin"); //south part implemented
        decmargin = new Button("Decrease safety margin");
        incmargin.addActionListener(this);
        decmargin.addActionListener(this);
        South.add(incmargin);
        South.add(decmargin);
```

- **Object received by threadedcontrolor.java sent to canvasd.java**

```
43    private boolean readCommand() {
44        DateTimeService xx;
45        try {
46      xx = (DateTimeService) is.readObject();
47
48        }
49        catch (Exception e){    // catch a general exception
50            this.closeSocket();
51            return false;
52        }
53        System.out.println("01. <- Received a String object from the client (" +  xx
54
55
56    int vertical   =    xx.getvertical();
57    int horizontal =    xx.gethorizontal();
58    int height     =    xx.getheight();
59    String dronename = xx.getdronename();
60    String dronenumb = xx.getdronenumb();
61
62    createdrone(horizontal,vertical,dronename,dronenumb,height);
63
64
65    send(xx);
66
67
68    return true;
69    }
70private void createdrone(int horizontal,int vertical,String dronename,String dronenum
71
72
73     ThreadedServer.canvas.receiveobjectattributes(horizontal,vertical,dronenumb,heig
74    ThreadedServer.canvas.addlfabel(dronename,horizontal,vertical,dronenumb);
75
76 }
```

Last three positions are gathered by this loop in canvas.java

```
if(client1positons.size() == 8)
{
    int    h1;
    h1  =        client1positons.elementAt(0);
    int v1 =     client1positons.elementAt(1);
    int h2 =     client1positons.elementAt(2);
    int v2 =     client1positons.elementAt(3);
    int h3 =     client1positons.elementAt(4);
    int v3 =     client1positons.elementAt(5);
    this.giu.clientslastposition(h1,v1,h2,v2,h3,v3);
    this.repaint();

       client1positons.remove(1);
       client1positons.remove(2);

    }
```

On mouse click ,position of the click will change the string variable in paint()

If mouse click is close to the drone position than ,that drones name will appear on the screen

```
198 public void objectpositions(int x,int y) {
199     System.out.println(x);
200     System.out.println(y);
201     System.out.println(client2horizontal);
202     System.out.println(client2vertical);
203     int c2xdifference = x - client2horizontal;
204     int c2ydifference = y - client2vertical;
205     int c1xdifference = x - client1horizontal;
206     int c1ydifference = y - client1vertical;
207     int c3xdifference = x - client3horizontal;
208     int c3ydifference = y - client3vertical;
209     int c2doublex = c2xdifference*c2xdifference;
210     int c2doubley = c2ydifference*c2ydifference;
211     int c1doublex = c1xdifference*c1xdifference;
212     int c1doubley = c1ydifference*c1ydifference;
213     int c3doublex = c3xdifference*c3xdifference;
214     int c3doubley = c3ydifference*c3ydifference;
215 
216     dronenameclient2="";
217     dronenameclient1="";
218     dronenameclient3="";
219 
220     if(Math.sqrt(c2doublex+c2doubley) < 30) {
221     dronenameclient2="client2";
222     dronenameclient1="";
223     dronenameclient3="";
224     }
225 
226     if(Math.sqrt(c1doublex+c1doubley) < 30) {
227     dronenameclient2="";
228     dronenameclient1="client1";
229     dronenameclient3="";
230     }
231     if(Math.sqrt(c3doublex+c3doubley) < 30) {
232         dronenameclient2="";
233         dronenameclient1="";
234         dronenameclient3="client3";
235         }
236 
```

How collision zone calculation is done between client1 and client2 ,can be seen below in canvasd.java

```
242 };
243 public void collisiondetection(int client1horizontal,int client1vertical,
244         int client2horizontal,int client2vertical,int client3horizontal,int
245         client3vertical)
246 {
247     int xdifference;
248     int ydifference;
249     if(client1horizontal > client1horizontal)
250     xdifference = client1horizontal - client2horizontal;
251     else
252     xdifference = client2horizontal -client1horizontal;
253     if(client1vertical > client2vertical)
254     ydifference = client1vertical - client2vertical;
255     else
256     ydifference = client2vertical - client1vertical;
257 
258     int ysquare = ydifference*ydifference;
259     int xsquare = xdifference*xdifference;
260     double distancecl1andcl2 = Math.sqrt(xsquare+ysquare) + 15;
```

Client gui, client.java

```
47
48          b_up = new Button("Go downwards");    //panel1 elements
49          b_up.addActionListener(this);
50          panel1.add(b_up);
51          b_down = new Button("Go upwards");
52          b_down.addActionListener(this);
53          panel1.add(b_down);
54          b_east = new Button("Go east");
55          b_east.addActionListener(this);
56          panel1.add(b_east);
57          b_west = new Button("Go west");
58          b_west.addActionListener(this);
59          panel1.add(b_west);
60
61          b_speedup = new Button("speed up");
62          b_speedup.addActionListener(this);     //panel2 elements
63          b_speedown = new Button("speed down");
64          b_speedown.addActionListener(this);
65          panel2.add(b_speedup);
66          panel2.add(b_speedown);
67          label1 = new Label("Location information");
68          label1.setFont(new Font("Serif", Font.PLAIN, 40));
69          label2 = new Label("Speed information");
70          label2.setFont(new Font("Serif", Font.PLAIN, 40));
71          label3 = new Label("time information");
72          label3.setFont(new Font("Serif", Font.PLAIN, 40));
73          location = new TextArea();
74          speed = new TextArea();
75          server = new TextArea();
76
77
78
79          panel3.add(label1);
80          panel3.add(label2);   //panel3 elements
81          panel3.add(label3);
82          panel3.add(location);
83          panel3.add(speed);
84          panel3.add(server);
85               if (!connectToServer(serverIP)) {
86                   System.out.println("XX. Failed to open socket connection to: " + serverIP);
87               }
88                this.addWindowListener(this);
```

Step by step how the message is received in client will updated and sent back

1.Received object sent again to sendobject function from there object sent to moveclient function.

```
        return true;
    }

    void  sendobject(DateTimeService client1) {
        //String theDateCommand = "GetDate", theDateAndTime;
        System.out.println("01. -> Sending Command ( DateTimeService  ) to the
        this.moveclient(client1); //objected passed through moveclient functio
        try{                        //for updating object
        DateTimeService yq = (DateTimeService) receive();
            System.out.println("05. <- The Server responded with: " );
            System.out.println("    <- " + yq);
            sendobject(yq);
        }
        catch (Exception e){
            System.out.println("XX. There was an invalid object sent back from
```

2.Object values are updated by the according the client selections.After update with Thread.sleep(10000) function 10 second delay is achieved and object sent to send function which sends the object to the server

```
132
133     private DateTimeService moveclient(DateTimeService x) {
134
135           x.verticalposition = verticalmovement;
136           x.horizontalposition = horizontalmovement;
137
138
139
140        try {
141             Thread.sleep(10000);
142          setdroneinformation(x);
143
144         } catch (InterruptedException e) {
145             // TODO Auto-generated catch block
146             e.printStackTrace();
147         }
148
149        send(x);
150
151       return x;
152 };
```

Below shows how client and drone object is first created.First function allows object to remains its parameters second function sent object to the server and connection starts

*args[2] is crucial to perform fully functional program.

```
 public static void main(String args[])
{

    System.out.println("**. Java Client Application - EE402 OOP Module, DCU"
    if(args.length==3){


        client theApp = new client(args[0]);
        DateTimeService xxx  = new DateTimeService(50,args[1],50,args[2],20)

         theApp.firstmoveclient1(xxx);
        theApp.send(xxx);
               theApp.sendobject(xxx);

    }
```

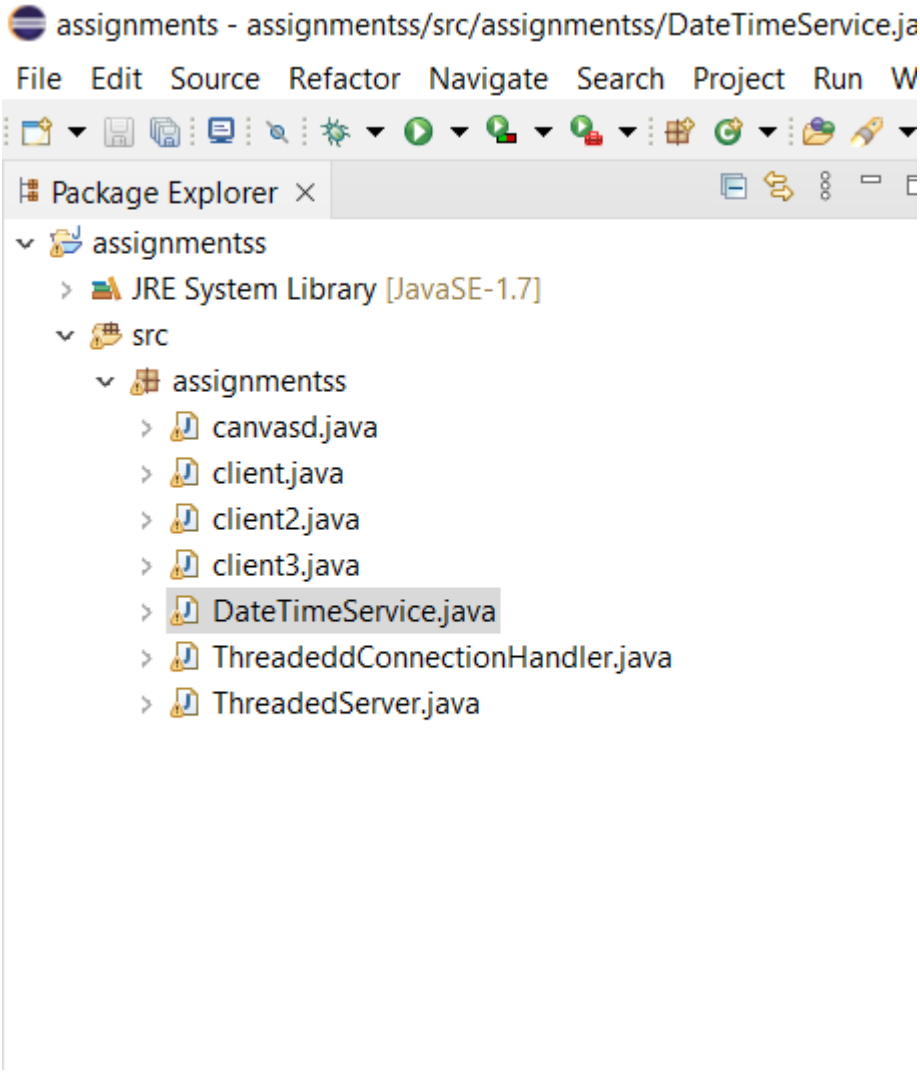
DateTimemessage.java, is the object that is sent through the connection,

Functions allow the server to reach individual features of the object

```java
import java.util.Calendar;
public class DateTimeService implements Serializable
{

   private String name;
   public int horizontalposition;
   public int verticalposition;
   public String dronenumb;
   public int speed =10;
   public int height;

   //constructor creates the Calendar object, could use the constructor:
   //   Calendar(TimeZone zone, Locale aLocale) to explicitly specify
   //     the time zone and locale
   public DateTimeService(int horizontal,String namez,int vertical ,String dronenumb ,int heig
   {

     this.horizontalposition = horizontal;
     this.verticalposition = vertical;
     this.name = namez;
     this.dronenumb = dronenumb;
     this.height = height;
     this.speed = speed;

   }

   //method returns date/time as a formatted String object
   public int getvertical()
   {
    return this.verticalposition;
   }
   public int gethorizontal()
   {

    return horizontalposition;
   }

   public String getdronenumb() {

       return dronenumb;
   };
   public String  getdronename()
```

Discussion/Testing

The program directory can be seen below



How to run the program?
1-Running threading server will open the server gui
2-To connect the client1

```
^C
C:\Users\DEO\assignments\assignmentss\bin>java assignmentss.client localhost birol 1
```

3-To connect the client2

```
C:\Users\DEO\assignments\assignmentss\bin>java assignmentss.client2 localhost bir
```

4-To connect the client3

```
^C
C:\Users\DEO\assignments\assignmentss\bin>java assignmentss.client3 localhost birol 3
```

5-All clients should work simultaneously and capable to do described tasks.

6-Sometimes due to 10 second delay , some actions might need to take another round.
7-when client disconnects the last position of drone still displayed until the client connected again.