COIT13229 Assessment item 1— Assignment 1

Due date: Friday of Week 6 (21st April 2022) 11.45 pm AEST

ASSESSMENT

Weighting: 30%

Length: NA 1

1. Objectives

The purpose of this assessment item is to assess your skills attributable to the following learning outcomes and your achievement of the expected graduate attributes of intermediate level communication, information literacy, and graduate level problem solving, critical thinking, and information technology competence.

- Develop software applications that can run in parallel and on multiple networked computers using approaches such as multi-threading, and client-server architecture
- Design and develop secure distributed applications using approaches such as RESTful webservices

2. Assessment task

You are part of a research project that is investigating the use of autonomous drones to find and report bushfire occurrences in remote regions of Australia. This new system is called Intelligent Bushfire Detection and Management System (IBDMS) here after.

Your task in this assessment is to analyse the given problem, model, design, implement, test and document a client/server application that allows multiple drones to communicate with a server (operations base) and access stored data or add new data. You will be implementing the software solution, applying efficient algorithms. The topics required for this assessment task are from Weeks 1-5. You should also write a report, as specified in this document, demonstrating your conceptual knowledge.

You will be required to use the topics learnt in the pre-requisite unit *Object-Oriented Programming.*

2.1 Problem

You will be developing a prototype software in two stages. In the first stage, the IBDMS should enable drone registration on the server, graphical display of all registered drones on a map on the on the server, and display of summary reports.

2.2 Design Guidelines

You can use the following guidelines in your modelling and design.

A. Server Side

The server should enable the following functionalities. The server creates a new thread for each drone connection (thread-per-connection model). The server side receives drone requests, processes them and returns results. Multiple drone requests are executed concurrently. Server side will have a GUI that displays the position of each drone on map

(background image). Server also displays and fire detected by any of the drones on the map.

1. *Load all previously saved data from existing binary and csv files in to the memory*
2. *Display all current information on a map*
3. *Receive Drone registration details*

Details of the drone is sent to the server for registration when a drone become operational at the start-up of the drone, this information is received from the drone side.

4. *Save Drone details*

   The received details of the Drone should be written to a binary file. The data should be appended so that previously entered data should not be over-written.

5. *Display the position of a registered drone on the map*

Server will redraw the map every 10 seconds with the existing information about all of the registered drones and any reported fire position.

6. *Send instructions to any drone*

The Server should also be able to send instructions to any drone. For example server is aware of the boundaries of the map and if a drone has crossed the boundaries of the map, server will send a return to previous position message to the drone, and drone must send an acknowledgement.

All possible fire information (detected and reported by drones) are kept in a text file (.csv)

Admin functionalities

i) Delete a fire report

An admin can delete a fire when the fire is attended and extinguished

ii) Recall all drones to the base

iii) Move a drone to a new position

iv) Shot down the server app

If admin user selects this function that the server sends a recall to all the drones and will only exit the program when the last drone acknowledges the recall message.

Note: A .csv file containing an example of fire and drone positions will be provided on the Unit website within the Assessment folder.

B. Client Side

The client side (Drone) Doesn't have GUI. It uses command line text to display the current information such as messages and information sent or received from the server.

1. *Register a Drone*

When a drone is started it immediately asks the user to enter the drone's name and drone ID then it connects to the server and send a request for registration. The registration information are four parts (drone ID, drone name, drone initial position consisting of x and y (usually 0, 0)). if a drone is already registered, the information will only be updated.

2. *Drone Position Message*

Drone must send its position to the server every 10 seconds. And receive acknowledgement from the server. If a drone has crossed the boundaries of the map, it will receive a return to previous position from the server.

3. Fire Detection Message

If a drone detects a fire reports the fire position to the server. The information is in three parts fire flag (9) and fire position [x, y]

4. Acknowledge the Recall Message

When a drone receives a "recall to base" massage from the server must acknowledge the message back to the server and return to the base (return to the base is just symbolic, drone only needs to print a message that is returning to the base and shut down automatically.

Note: This is a prototype only and as such, all the information sent to the server by the drones can be randomly generated, but it should be rationalised, so the drone doesn't behave erratically

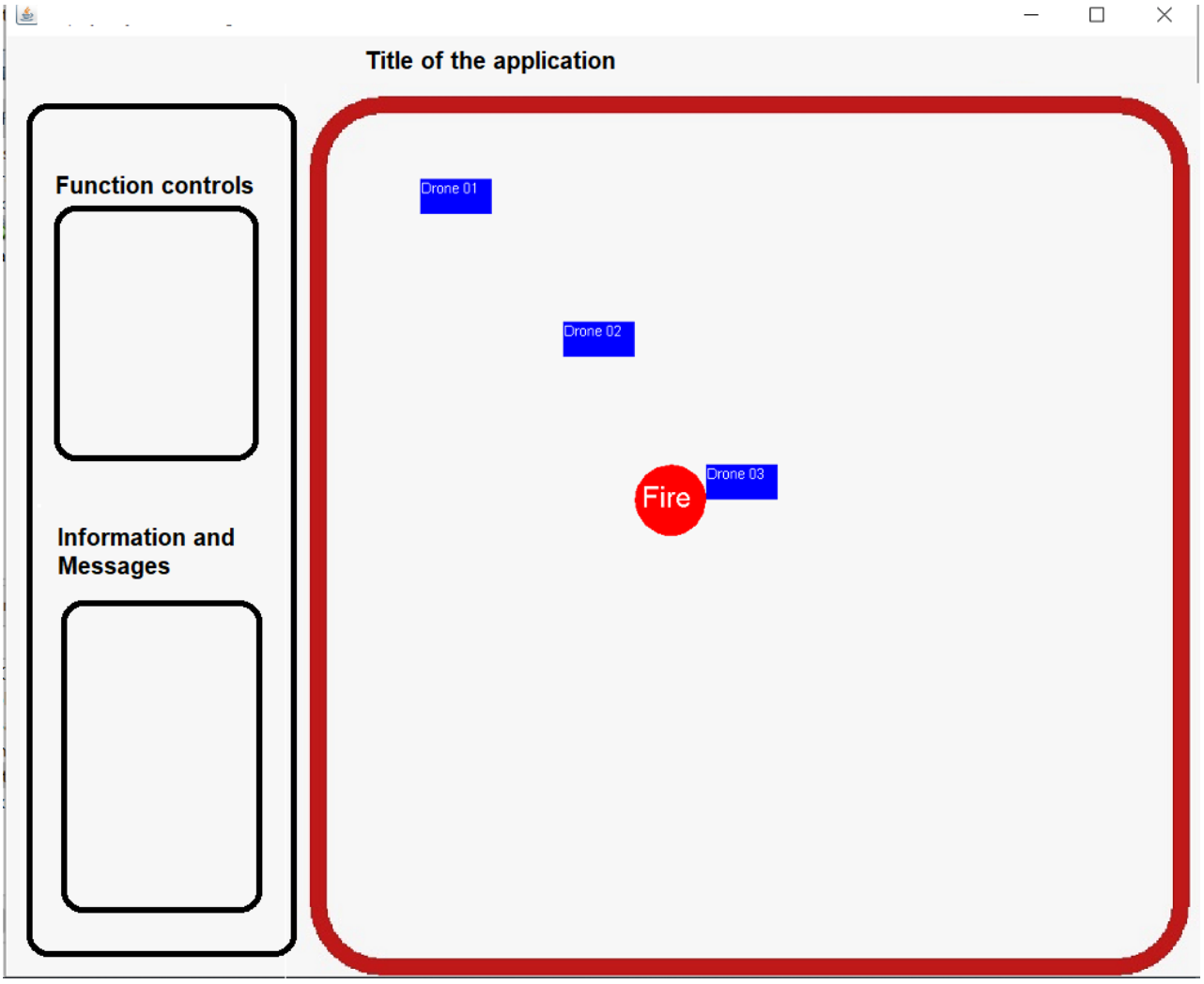
C. Server side Graphical User Interface

The GUI should have the necessary components to enable the admin user to execute all the functions as provided in Section 2.2. You may use the guidelines provided below for your GUI design and implementation. Variations to the provided guidelines are acceptable if it meets the user requirements. The GUI should be designed to provide an easy-to-use user interface that provides informative error messages, and clear instructions. This should be intuitive and easy to navigate.

The server GUI has two parts one is the map that displays the position of all registered drones and possibly fires. On the side of the map there should be an area to display all the necessary controls, messages and other relevant information

You use necessary Controls such as Buttons, TextFields, ComboBoxes and others.

The following image is just a guide not a requirement to build your GUI



3. Coding

(Do not use java FX) The GUI of this application is very simple and can be easily done using swing and graphic libraries.

GitHub

You must use GitHub as your code repository and version control.

All your projects (Server project and Drone Project) must be placed on GitHub repository and a link to the repository must be provided in your report for inspection.

The assignment markers will inspect your code repository in GitHub and they must see a gradual development of the code. If your code appears over very short period of time, then your assignment will be treated as a possible case of plagiarism and the necessary steps will be taken to assess it as such.

Include necessary accessor, mutator methods, constructors, and toString() methods for each class. Also, follow good coding practices, using meaningful names, camel case notation for naming, constants as necessary, and include meaningful comments. You can use NetBeans to develop your application. *Follow the coding standards given in the Unit website.*

You can have top level generic classes and extend them to create specialized classes. You need not include login information and store such details.

3.1 Guidance to the level of assistance you can use.

You are expected to understand several concepts and apply those concepts to design and build a software solution. At this level you can use the provided materials, online resources for further reading and take assistance from your classmates or teammates to develop deeper understanding of the concepts. You can also sort help to debug the implemented program. But you should design, implement and test your program on your own.

Table 1 Allowed Assistance

S.No	Source	Activity
1	Unit Textbook, Unit notes and examples	Understanding concepts, design
2	Instructors	Understanding concepts, design, debugging
4	Classmates, Online resources	Understanding concepts
5	Everyone else	Understanding concepts
6	**No help acceptable**	**Designing and implementing code**

4. Report

You should submit a report containing the following details.

1. UML class diagrams for the classes
   Note: UML class diagrams generated using a software tool after completing the coding will not be accepted.
2. A diagram showing the system architecture illustrating the client server connection and the multiple clients connecting to the server.
3. Test plan showing input data, expected results, and actual results. Show testing of erroneous entries also.
4. Clearly write the steps of client – server communication happening during the program run and data going between client and server for the Customer Registration event. You can use a diagram.

5. Assignment Submission

You should submit the following files and word document using the Moodle online submission system. (Note: the file names/class names could be changed to meaningful names).

1. ***Server project - zip the whole project folder and submit it.***
2. ***Drone project- zip the whole project folder and submit it.***

   Your submitted project must be self-sufficient and contain all necessary files and libraries to run it in another computer without modifications.
3. *Report.doc – Your word document containing a well-presented report with student details, UML class diagram, system architecture, demonstration of testing all functions of the program (screen shot of the client server communication during program run).*

Assessment Item 1 Marking criteria

S.No	Total Marks - 30	Marks Allocated	Marks Scored

1	Graphical User Interface Presentation: User Friendly Design (1 Mark) Map is displayed and refreshed with correct information every 10 second (2 Marks)	3	
2	Design and use of appropriate data structures	2	
3	Correct use of inheritance	1	
	Drone	0	
4	A drone class enabling connection to the server (1 mark) Drone is moving rationally and not behaving erratically (1 mark)	2	
5	Appropriate event handling to send/receive messages to the server for the drone registration functionality	2	
6	Appropriate event handling to send/receive messages to the server for the drone to send new position information	2	
7	Appropriate event handling to send/receive messages to the server for the fire reporting	2	
8	Appropriate event handling to send/receive messages to the server for the recall and shutdown	2	
	Server	0	
9	Server class that creates thread per connection. (1 mark) Reads data from a file at start and saves to files in exit (1 marks) Correctly displays map, drone and fire positions (2 marks)	4	
10	Correct implementation of object serialization and saving data in binary format in the file and reading and loading data from the file	2	
11	Correct use of exception handling	2	
12	Good coding practices including comments, indentation, use of constants as required, use of meaningful names, and camelCase notation for names	2	

13	Well-presented report with student details (1 mark), UML class diagram (1 mark), system architecture (1 mark), demonstration of testing all functions of the program, and the client server communication during program run (1 Mark)	4	
	Penalties		
14	Late Penalty (-1 mark: 5% of total allocated marks per calendar day)		
15	Source code is not in GitHub or is uploaded in a very short period of time, also if it is found entirely different from the styles followed in the Unit or containing constructs outside of this Unit will be penalised (-1 to -20 marks depending on the problem)		
16	Plagiarism (penalty as per the plagiarism policy)		