

## COM373 – Programming II (CRN: 41811)

### Assignment: The Medical Resource Management System

*This assignment constitutes 60% of the total mark for the module.*

**This assignment should be completed in small teams of between 2 and 4 people.**

Providing remote medical diagnosis and consultation for countries with limited medical facilities is already happening. In many war-torn countries around the world there are too many casualties and too few qualified doctors to cover the need. In such circumstances, less qualified medical staff may often be required to cover situations locally, where a fully qualified doctor or specialist is unavailable. The idea is that a group of doctors with particular medical specialities would be available, in another country, to undertake skype/facetime sessions with the medical staff in the war-torn country. Medical staff in the war-torn country would call a central phone number for assistance. An administrator would take a few details and then provide the caller with the contact details for an appropriately skilled and available doctor.

You have been asked to develop a graphical Java application to assist an Administrator in the management of available doctors. Your application should satisfy the following requirements:

1. An administrator will have sole access to the Medical Resource Management system. The system should be stand-alone – no internet access is required for this initial system.
2. The administrator will handle all requests for assistance received by phone.
3. The system will need to handle up to a maximum of 10 registered specialist doctors in each supported medical discipline. The following 10 medical disciplines are to be supported by the centre:
  - a. Cardiology
  - b. Pulmonology
  - c. Infectious Disease
  - d. Hematology
  - e. Intensive Care Medicine
  - f. Neurology
  - g. Ophthalmology
  - h. Orthopedics
  - i. Urology
  - j. Surgery
4. The system must enable doctors to be registered with the service. The Administrator will enter the doctor's details, to include:
  - a. Name
  - b. Contact Address
  - c. Preferred means of contact (telephone, skype or facetime) with the contact ID required to connect with them.
  - d. Their medical discipline.

5. The system must store the doctor's details to a textual file called the "Doctor Records File". The format of text within this file is not specified, the developer is at liberty to decide this detail. If the system is stopped and restarted, any registered doctors and their details should be loaded back into the system from this "Doctor Records File".
6. In addition to the details given in point 4 for all doctors, for surgeons, the date of their latest certification must also be recorded and displayed along with their other details. If the date of their latest certification is more than one year from today's date, then this must be flagged up to the administrator when their details are accessed.
7. A given doctor may only be registered as a specialist in one of the medical disciplines given in point 3 above.
8. Periodically, the Administrator will be given a text file containing details of the availability of all registered doctors for that next week. This file will be created through a separate system, which is beyond the scope of your work. However, your system must read this file and use it to update the availability of all registered doctors. The file (called the "Availability Update File") will have the following format (with all times specified using the standard 24-hour clock), with each item occupying a separate line in the file:
  - a. Doctor's Name
  - b. Sunday<space> start time (hh:mm) <space> end time (hh:mm)
  - c. Monday<space> start time (hh:mm) <space> end time (hh:mm)
  - d. Tuesday<space> start time (hh:mm) <space> end time (hh:mm)
  - e. Wednesday<space> start time (hh:mm) <space> end time (hh:mm)
  - f. Thursday<space> start time (hh:mm) <space> end time (hh:mm)
  - g. Friday<space> start time (hh:mm) <space> end time (hh:mm)
  - h. Saturday<space> start time (hh:mm) <space> end time (hh:mm)

For example:

<Start of File>

John Smith

Sunday 12:00 15:00

Monday 08:00 08:00

Tuesday 15:00 16:00

Wednesday 19:00 23:00

Thursday 00:00 03:45

Friday 07:45 08:55

Saturday 13:00 17:30

Amanda Murphy

Sunday 01:00 03:25

Monday 11:00 13:30

Tuesday 17:00 18:00

Wednesday 09:30 11:45

Thursday 07:00 07:00

Friday 09:45 12:55

Saturday 10:00 18:45

<End of File>

This pattern will repeat for each registered doctor in the system. It is possible that an updated version of the file may be provided at any time of any day of the week. When the updates are applied to the system, it is necessary to determine which day of the week the update is happening on and to only update the system with the records from that day onwards from within the file. Note also that if a doctor is unavailable on a particular day, then the start and end times will be set to the same value. You can also assume that there will be only one contiguous block of availability for any doctor on any given day.

9. The system must also enable the Administrator to scroll through the list of registered doctors and to de-register any doctor as needs be.
10. For any given medical discipline, the display of doctors registered against that medical discipline should appear as follows, with red indicating unavailability at present, green indicating current availability and grey indicating a gap in the list of registered doctors for that discipline:



*Figure 1 – Medical Discipline Doctors Register.*

11. The system must provide a suitable graphical display making it easy and intuitive for the Administrator to navigate through the screens to locate an appropriate available doctor and their contact details.
12. The system must enable a report to be generated showing all periods of time in the current week when there is no availability across the set of registered doctors for each medical discipline.

If you have to make any assumptions about the operation of the application to cover any issues of either imprecise or unclear requirements then please state these clearly in a comment at the beginning of your code listing.

### **Hand In Instructions:**

This assignment should be completed in self-selected teams of between 2 and 4 people. Each team must submit a signed 'team-assessment declaration' form, to indicate what percentage contribution each team member has made to the overall effort. For any team submission, individual effort

declarations deviating from an equitable split in the effort will result in the marks for this assignment being scaled according to the declarations made. The scaling will happen according to the following example of the scheme: Consider a 2-person team submission that is deemed worthy of a mark of 50%. Consider also that one of the pair has a declared contribution of 60% and the other declared at 40%. The person whose contribution is the lesser will have their mark reduced by 10/50 or 20%, i.e. to 40% in this example. The person whose contribution was set at 60% will receive the full awarded mark of 50% for the work.

Only **ONE** copy of the team's solution should be submitted as a zip file to Blackboard Learn. Please ensure that ALL team member names and corresponding B-Codes are clearly indicated together with screen shots of each part of the user interface in a separate Word/pdf document. This document is to be included in the zip file. **PLEASE NOTE: the program MUST be submitted as a Netbeans project to assist with the marking procedure.** The submitted zip file should contain all of the files and sub-folders in the NetBeans project so that all source code can be inspected and the entire program can be executed if necessary. Note also that Blackboard will only permit you to make ONE submission, so please ensure it is the finalized version of your team's software. **Do not** make use of other team member's Blackboard accounts in order to re-submit revised versions of your team's solution.

Failure to submit a readable zip file may result in a mark of zero being attributed to the work.

### Specific submission guidelines (PLEASE READ CAREFULLY)

1. The completed assignment (including a document declaring team member names and user interface screens; plus all source code (.java), compiled files (.class) and all Netbeans project files in their respective sub-folders, should be zipped and uploaded as one zip file to Blackboard Learn on or before 12:00 noon on **Tuesday 5<sup>th</sup> December 2017**.
2. Please ensure that you keep a secure backup electronic copy of your assignment.
3. Your team will be required to give a short demonstration of your working program in the scheduled lab sessions in weeks 11 and 12. This demonstration will be used to assess the 'Program execution correctness and completeness' criteria in the following marking scheme. It is important that **ALL** team members are present for the program demonstration.
4. Remember that each team must also submit a completed and signed team-assessment declaration form. This should be placed in Dr Wilkie's physical mailbox located in the school office (16G24), on or before the assignment due date.

### Marking Scheme:

Marks will be awarded according to the following criteria.

Component	Marks
Overall Program Structure, code style and readability	25%
Program execution correctness and completeness	25%
Use of GUI Components and user interface layout	20%
Use of Inheritance and Polymorphism	10%
Use of Exception Handling	10%

Use of suitable Searching techniques.	10%
---------------------------------------	-----

**PLEASE NOTE:** This is a team programming assignment and so the work submitted must be your own or that of your team. Open source software is not to be used in the creation of your solution. You may, of course, make use of any classes contained in the JDK or Swing libraries. You may also make use of the NetBeans GUI design utility and the auto-generated code thereby produced.

Plagiarism is a serious offence. Please refer to [www.ulster.ac.uk/academicservices/student/plagiarism.pdf](http://www.ulster.ac.uk/academicservices/student/plagiarism.pdf) for further information.