UNIVERSITY OF TECHNOLOGY, JAMAICA

School of Computing and Information Technology (SCIT)

PROGRAMMING II PROJECT (GROUPWORK)

GROUP: Bsc

DATE GIVEN: Week of February 22, 2021

DATE DUE: Week of March 28, 2021

Group size 4--5 members

The objectives of this project are:

- To encourage research and self -study (as needed for the topics of structures and files in order to complete this project)
- To encourage team effort as encouraged in real world development companies
- For students to gain the experience to provide the solution code for current real-world problems

Solve the Following

To help with the battle against COVID-19, the Conglomerate Hospice Ltd. has asked for your help in developing a system that will not only help to keep track of their patients, but also use some artificial intelligence to place patients in to free rooms depending on their status.

Your system is to load existing patients and rooms from two flat files and then allow for the management of the contents whilst the system is running. At any point that the system is closed, there should automatically be new flat files that reflect the changes made whilst the system was running.

The loading is to occur in one of two ways: -

1. Use the command prompt arguments in the form of: `C:\> program_name.exe patient.txt rooms.txt` so that the program knows of the flat file locations as the program starts; or
2. As soon as the program loads up, ask the user to enter the file name and location for the patient flat file, then for the rooms flat file, before continuing to the menu.

The two flat files should look like the following: -

patient.txt

```
1000212 Patrick Hassery           mild      1972 admitted
1000214 Jessica Melissa Grant     severe    1946 admitted
1000217 Hoogland F. Thompson      severe    1968 admitted
1000219 G. T. Thompson            average   1984 admitted
```

where each patient has the following contents: id number, full name, COVID-19 status (none, mild, average, severe, critical), birth year and if the patient is admitted or discharged. Note that the spacing is set for each patient record whereby id numbers take up 7 bytes, names take up 25 bytes, statuses take up 9 bytes, 4 bytes for the birth year, and 9 bytes for admitted status. A space separates each field of contents.

rooms.txt

```
230 respirator 2 none   1000212,1000217
231 none       4 nurse  1000214
232 none       2 doctor
233 none       1 nurse  1000219
```

where each room has the following contents: room number, whether it has a respirator (respirator or none), the number of beds it has, which personnel is assigned (nurse, doctor or none) and which patients have been assigned to beds in that room. Note that there can never be more than five (5) beds assigned to any room. The spacing is as follows: 2 bytes for room number, 10 bytes for respirator, 1 digit for number of beds, 6 bytes for personnel and up to 40 bytes for patient id numbers (8 bytes x 5 patients maximum). A space separates each field of contents.

When your system loads it should provide the user with the following options: -

- Add A Patient: adds a new patient to the list of patients (note that these buildings can only accommodate a maximum of 35 patients). All validation protocols should take place to ensure that new entries follow the field requirements listed above.
- Add A Room: adds a new room to the list of rooms (note that these buildings can only accommodate a maximum of 21 rooms). All validation protocols should take place to ensure that new entries follow the field requirements listed above.
- Change Patient Status: allows the user to change the status of any patient currently in the system (even if they are discharged). The patient is located by their unique id numbers.
- AI Assignment: allows the system to assign those patients who do not yet have a room based on the following protocols: -

Age Groups	COVID-19 status	Type of Assignment
All groups	Critical	Room with bed, respirator and doctor
> 60	Severe	Room with bed, respirator and/or nurse
> 60	Mild, Average	Room with bed
40 to 60	Average, Severe	Room with bed and/or nurse
40 to 60	Mild	Room with bed
20 to 39	Severe	Room with bed and doctor
20 to 39	Mild, Average	Room with bed
< 20	Average, Severe	Room with bed and nurse
< 20	Mild	Room with bed
All groups	None	Discharge

- Note that the order of the priority depends on the top-down design of the table whereby those listed first get priority over the others
- Discharge: the patient's record is changed from admitted to discharged and the patient is removed from the room they were assigned (if they were assigned a room). Patients are discharged based on their unique id numbers.
- Room admittance by age group Report: A bar chart frequency report that will show the number of persons currently assigned to rooms grouped by their age groups: over 60s, 40 to 60, 20 to 39 and younger than 39. The user should then be given the option to save the report to a file (the system will request a file name from the user to save the file).
- Admitted patient list Report: A list of all of the patients currently admitted ordered by their id numbers (in ascending order) showing their id numbers, names, room they are currently assigned ("none" if not yet assigned a room) and current COVID-19 status. The user should then be given the option to save the report to a file (the system will request a file name from the user to save the file).
- COVID-19 status Report: bar chart frequency report that will show the number of persons grouped by each status: none, mild, average, severe and critical. The user should then be given the option to save the report to a file, which file name they supply to the system. This report should be ordered by frequency from highest to lowest (descending order). The user should then be given the option to save the report to a file (the system will request a file name from the user to save the file).
- Exit: close the system with automatic save of changes made to the file
- Exit without save: abruptly close the system disregarding all changes made to the system

Your system should also generate a log file for auditing purposes that keeps track of the following information: Date and time that the menu request was made, and which menu request was made.

Expected Weekly Progress

In order to complete the project on time, the following expected weekly progress should be made and should be accountable to your selected project leader and vice project leader.

Week 8 - Algorithm - Design for project (Flowchart & pseudocode code)
(Due date and upload of pseudo code or flow chart to the appropriate Moodle link is March 10, 2021)
Week 9- Complete and demo menu & Complete and demo Add functionalities
Week 10 - Complete and demo Status Change and Discharge functionality
Week 11 - Complete and demo AI Assignment to room Functionality
Week 12 - Complete and demo reports
Week 13 - Final Project presentation.

Tutors will request to see your progress at least twice during the span of the project development process.

Submission

You are required to:

1. **Submit and upload pseudo code or flowchart to the appropriate Moodle link is March 10, 2021**
2. Submit all required project documents to the appropriate Moodle upload link, do not send them to your tutor's email
3. Submit a cover sheet indicating the names and identification numbers of all group members, lab tutor's name and occurrence, along with a copy of this project paper
4. Submit a signed copy (by each group member) of the "Declaration of Authorship" form.
5. Submit a detailed description and illustration of each component (for example, function, struct, et cetera) of the project implemented by each group member.
6. Submit, flowchart or pseudocode code an executable and source code (source code should have authors name - each group member's name also lab tutor's name and occurrence.)

Group Project Rubric Table

Grading Area (100%)	**Excellent**	**Good**	**Fair**	**Not Good**
15% - **Documentation (source code)**	**13 – 15%** comments describe every scope clearly but succinctly	**10 – 12%** comments describe some scopes clearly	**5 – 9%** comments exist but are not clear or accurate	**0 – 4%** little to no comments exist
10% - **proper control structures**	**8.1 – 10%** decisions and iterations used efficiently to solve project's issues	**5.1 – 8%** decisions and iterations exist but can be more efficient	**2.6 – 5%** decisions and iterations exist but not used correctly	**0 – 2.5%** little to no use of decision and iteration

10% - **use of arrays**	**8.1 – 10%** arrays exist and are efficient and solves project's issue	**5.1 – 8%** arrays work but are not efficient, do not help solve project's issue	**2.6 – 5%** arrays exist but do not work	**0 – 2.5%** no arrays used in solution
10% - **use of files**	**8.1 – 10%** files are saved and loaded correctly	**5.1 – 8%** import / export correctly, bad report file contents	**2.6 – 5%** files import / export correctly, no report files	**0 – 2.5%** no file import / export services
10% - **correct functionality**	**8.1 – 10%** efficient use of memory, no unnecessary code	**5.1 – 8%** efficient use of memory, some unnecessary code	**2.6 – 5%** inefficient use of memory, unnecessary code	**0 – 2.5%** bad use of memory, unnecessary code
10% - **presentable reports**	**8.1 – 10%** reports are accurate and presentable	**5.1 – 8%** reports exist, output is accurate, not presentable	**2.6 – 5%** report charts exist but are not correct nor presentable	**0 – 2.5%** little to no reports generated
15% - **proper use of functions**	**13 – 15%** all modules have a purpose, all modules used correctly	**10 – 12%** all modules properly defined but not used correctly	**5 – 9%** some modules exist, but either not used or not defined properly	**0 – 4%** very little to no modules used
20% - **user interface**	**18 – 20%** friendly menu, clear errors, validation messages	**11 – 17%** friendly menu, error messages, little to no validation	**6 – 10%** menu and some messages exist but not user friendly	**0 – 5%** broken/no menu, broken/no messages, no validation checks

Marks may be deducted for the following:

• Unauthorized late submission – 10% per day (weekend included). After three days, NO assignment will be accepted.

• With regard to identical assignments. Please read your Student Handbook for the University's Policy on Academic Misconduct. • Lack of neatness, and disorganization of the deliverables. CMP1025 Semester Project for Sem2 AY2016-17 Page 6 of 6

• The inability of a group member to explain the code and logic approach taken. Note that any member of the group may be called upon to explain any area of the system regardless of the tasks performed in the project.

• Unauthorized lateness or absence from interview will result in the project NOT being graded.

• The program does not run and compile without syntax error it WILL NOT be graded. (Compile often to make sure program runs)