FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF INFORMATION AND MEDIA TECHNOLOGY

SECOND SEMESTER 2016/2017 EXAMINATION

COURSE CODE:	CIT 323
COURSE TITLE:	JAVA DESKTOP PROGRAMMING
CREDIT UNITS:	3
TIME ALLOWED:	3 HOURS
COURSE LECTURER(S):	DR. H. O. ALIYU
NUMBER OF QUESTIONS:	4
NUMBER OF PAGES:	4

INSTRUCTIONS

- Answer all questions in Parts I and II
- Do **not** use red pen
- Please use a clear handwriting
- This exam is closed book, closed notes, **opened laptop** and closed cell phone
- Please use non-programmable calculators only

Part I: Essay

Considering that Figure 1 shows the ***view*** of a simple JavaFX application, use the figure to answer questions 1and 2.

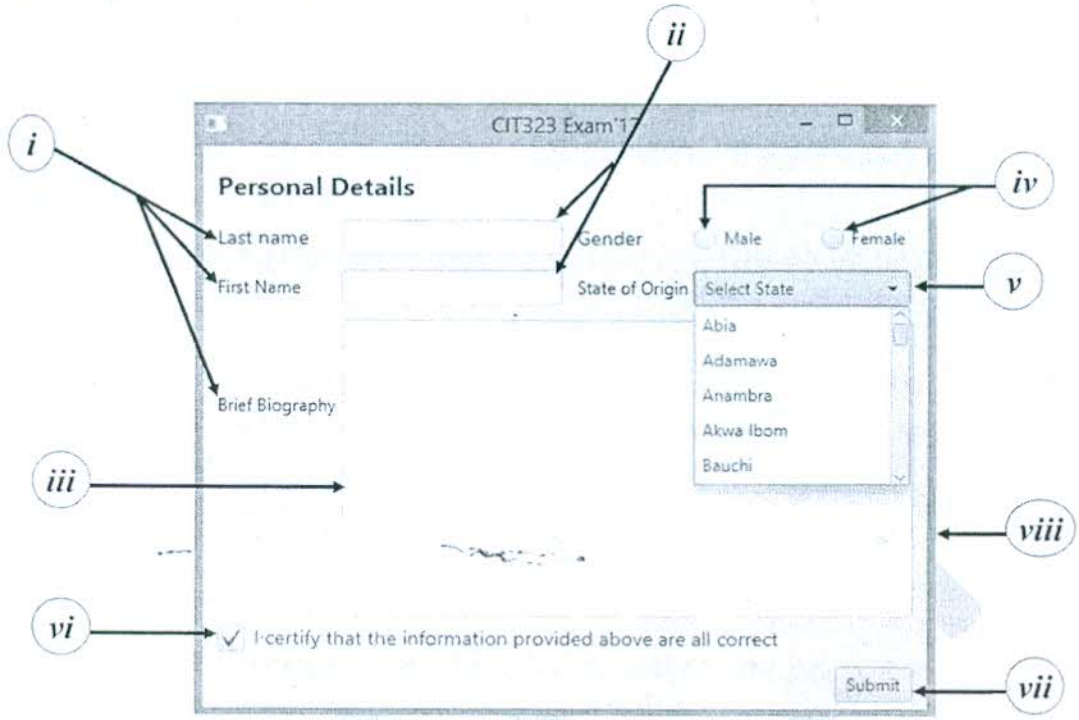

Figure 1. Use this figure to answer questions 1 and 2

Questions 1:

a. Identify the JavaFX components labelled (***i-viii***) in Figure 1 above **(8 marks)**

b. What is the JavaFX component that contains all the control elements displayed on component (***viii***) in Figure 1? **(1 mark)**

c. State one significant difference between when to use control elements (***ii***) and (***iii***) in a JavaFX application. **(1 mark)**

Questions 2: Given that the ***fx:id***s of control elements (***vi***) and (***vii***) in Figure 1 in the Scene Builder are set to `chbCertify` and `btnSubmit` respectively:

a. How would you declare the injectable fields (variables) for each of control elements `chbCertify` and `btnSubmit` in the Contoller class of the JavaFX application? **(2 marks)**

b. If the control element (***vii***) must be inactive by default (i.e., disabled at the launch of the application); name the special method of the controller class of the application in which this must be done and show, with a code snippet, how to do it. **(3 marks)**

c. If control element (*vii*) should be activated (i.e., enabled) and deactivated (i.e., disabled) whenever element (*vi*) is selected and deselected respectively, describe with a code snippet and a brief comment, how this can be achieved. **(5 marks)**

Question 3:

a. Briefly describe the HashMap data structure and the following methods in the HahshMap class. **(5 marks)**

i. `put()`
ii. `get()`
iii. `keySet()`
iv. `containsKey()`
v. `remove()`
vi. `clear()`

b. Briefly explain object serialization and state four (4) differences between text files and binary files **(5 marks)**

Part II: Practical

Question 4: **(30 marks)**

Assuming the FUTMinna's clinic needs a Java Desktop Application called HRecManager to manage the health records of the patients. Figure 2 presents an overview of the principles of operation of the application.

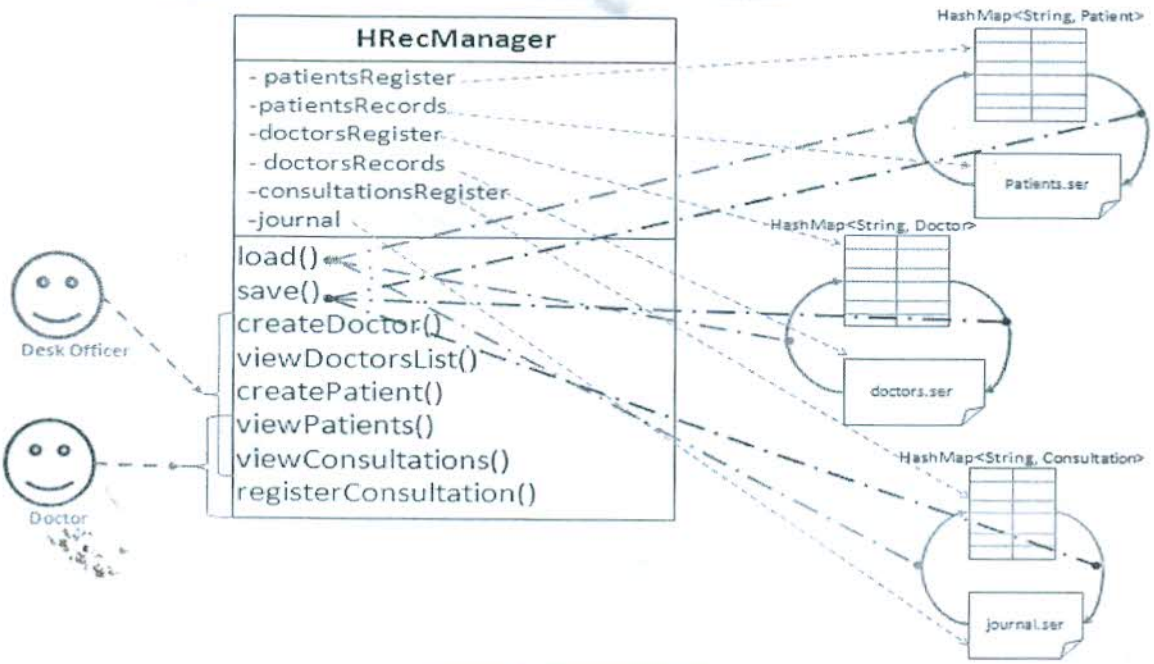

Figure 2. HRecManager

HRecManager has two(3) active parts of its memory called ***patientsRegister, doctorsRegister*** and ***consultationRegister***. It also has three passive storages, ***patientsRecords, doctorsRecords*** and ***journal,*** which are binary files used for persistent storage of serialized ***patients***, ***doctors*** and ***consultations*** objects respectively.

When the application is launched, it loads (**method load()**) the contents of the records (binary files) into their respective registers, each of which is a hash map - a collection data structure that allows dealing with key-value pairs in the memory. For example, ***patientRegister*** holds "*patient numeber - patient object*" pairs such that the unique account numbers serve as the keys to access account objects in the memory. Similarly, ***doctorsRegister*** holds "*doctorId- doctor object*" pairs and ***consultationRegister*** holds "*consultation number - consultation object*" pairs to keep track of all consultations for future references and follow ups of treatments. HRecManager can also save (**method save()**) the contents of its registers to the record files when changes have been made to them.

MFBManager has two categories of users ***Desk Officer*** - a Clinic Administrative Staff that serves as an interface between the clinic and its patients- and ***Doctors*** with whom the patients consult with for medical treatments. The set of operations that can be performed by each category of users is as indicated in Figure 2.

Develop a JavaFX application for HRecManager, providing suitable GUIs for welcoming page and the different operations (or groups of operations) and their respective controllers.

Hint: each of *Patient*, *Doctor* and *Consultation* is a class with suitable attributes and methods.

Class Patient may contain attributes such as `patientNumber, name, age, gender and department`.

Class Doctor may contain attributes such as `doctorId, name and gender`

The Consultation class may have attributes like `reportedSymptoms, diagnoses` and `prescriptions` each of which can be a multiline string that can be entered in a text area. It should also contain the name of the doctor consulted and the patient's name.

Best of Luck!