UECS2344 Software Design: Practical 3

1) The following application is used to manage a registry of students. It consists of the following:
   - Student *class* that represents a student
   - IStudentRegistry *interface* which contains the methods for managing registry of students
   - RegistryArrayList *class* which *implements* the IStudentRegistry interface and manages a list of students in an ArrayList
   - RegistryApp *class* which contains the main() method – all input and output is handled in this class

Take note that the classes and interface are organized into two **packages**:

- *registry.app* package
- *registry.domain* package.

Also the classes in *registry.domain* package are **imported** for use in *RegistryApp* class.

(a) Run the application to see how it works.

(b) Draw the following diagrams for the application:

(i) Class Diagram

(ii) Sequence Diagram

```
// in file Student.java

package registry.domain;      // registry.domain package

public class Student {

   private String name;

   private int id;

   public Student(String name, int id) {
         this.name = name;
         this.id = id;
   }

   public String getName() {
         return name;
   }

   public int getId() {
         return id;
   }
```

```
}

// in file IStudentRegistry.java
package registry.domain;      // registry.domain package

import java.util.List;

public interface IStudentRegistry {

      public void addStudent(String name, int id);

      public Student searchStudent(String name);

      public int getNumberOfStudents();

      public List<Student> getStudents();

}
```

```
// in file StudentRegistryList.java

package registry.domain;      // registry.domain package

import java.util.List;
import java.util.ArrayList;

public class StudentRegistryList implements IStudentRegistry {

      private List<Student> students;

      public StudentRegistryList() {
            students = new ArrayList<Student>();
      }

      public void addStudent(String name, int id) {
            Student aStudent = new Student(name, id);
            students.add(aStudent);
      }

      public Student searchStudent(String name) {
            boolean found = false;
            int i = 0;
            int count = students.size();

            Student theStudent= null;
            while (i<count && !found) {
                  theStudent = students.get(i);
                  if (theStudent.getName().equals(name))
                        found = true;
                  else
                        i++;
            }
```

```
            if (!found)
                  theStudent = null;
            return theStudent;
      }

      public int getNumberOfStudents() {
            return students.size();
      }

      public List<Student> getStudents() {
            return students;
      }

}
```

```
// in file RegistryApp.java

package registry.app;                  // registry.app package

import java.util.List;
import java.util.Scanner;

import registry.domain.*;              // import classes from registry.domain

public class RegistryApp {

      private static IStudentRegistry studentList;

      private static Scanner scanner;

      public static void main(String[] args) {

            studentList = new StudentRegistryList();
            scanner = new Scanner(System.in);

            int choice;

            do {
                  System.out.println("Do you want to:");
                  System.out.println("1. Register new student");
                  System.out.println("2. Search for student");
                  System.out.println("3. View all students");
                  System.out.println("4. Exit");

                  System.out.print("Enter your choice (1-4): ");
                  choice = scanner.nextInt();
                  // read the enter key after integer input
                  String skip = scanner.nextLine();

                  while (choice < 1 || choice > 4) {
                        System.out.println("Invalid choice.");
```

```
                        System.out.print("Enter your choice (1-4): ");
                        choice = scanner.nextInt();
                        // read the enter key after integer input
                        skip = scanner.nextLine();
                  }

                  switch(choice) {
                        case 1: addStudent(); break;
                        case 2: viewAStudent(); break;
                        case 3: viewAllStudents(); break;
                  }
                  System.out.println();

            } while (choice != 4);
      }

      public static void addStudent() {
            System.out.print("Enter student name: ");
            String theName = scanner.nextLine();
            System.out.print("Enter student id: ");
            int theId = scanner.nextInt();
            // read the enter key after integer input
            String skip = scanner.nextLine();

            studentList.addStudent(theName, theId);
            System.out.println("Student added");
            System.out.println();
      }

      public static void viewAStudent() {
            System.out.print("Enter name of student: ");
            String theName = scanner.nextLine();

            Student theStudent = studentList.searchStudent(theName);

            if (theStudent == null)
                  System.out.println("No student with that name found");
            else
                  System.out.println("Name: " + theStudent.getName()
                                    + "\tId: " + theStudent.getId());
            System.out.println();
      }

      public static void viewAllStudents() {
            List<Student> theStudents = studentList.getStudents();

            for (int i=0; i< theStudents.size(); i++) {
                  Student aStudent = theStudents.get(i);

                  System.out.println("Name: " + aStudent.getName()
                                    + "\tId: " + aStudent.getId());
            }
      }
}
```

2) “Z-PrelovedClothforBaby is a web application for buyers and sellers of preloved clothes for babies. A person must register with Z-PrelovedClothforBaby and provide a current physical address, telephone number, and email address. Each user must insert an unregistered email to create a new account. Otherwise, the system displays the error and lets the user reinsert another email address. Once registered, the user will receive a confirmation email.”

Draw a sequence diagram for this scenario.