# LAB 211 Assignment

| | |
|---|---|
| **Type:** | **Long Assignment** |
| **Code:** | **J1.L.P0020** |
| **LOC:** | **400** |
| **Slot(s):** | **20** |

## Title

Aboard Programs Management System

## Background

**FPT University's IC department** provides information **about study programs and experience aboard** for FPT students. Currently, the IC department at FPT University HCM provides some study aboard programs to students on Facebook page. FPT students use this page to **view details of the study tour** such as: **study programs** (English 6 oversea, semester exchange, OJT overseas), school, time, location, fee, …. After that, students are going to come to the IC department to register and will receive an email with the results.

For Example:  Details of the study programs at a foreign school:

0. ID and Name: E6O-English 6 Overseas

1. Time : 6 batches (January, March, May, July, September, November)
2. Day : 30 - 40 days.
3. Location .
1. Help University , Malaysia
2. UOW Malaysia KDU University College, Malaysia
3. Multimedia University , Malaysia
4. UOW Malaysia KDU Penang University College,Malaysia
5. University of Brunei Darussalam, Brunei
6. Swinburne University of Technology , Úc
4. Cost
5. Content  //only the path that link to the file .pdf

All IC's staffs are using some applications such as: Excels, Zalo, … to manage aboard programs, student management, and information of registration. All ways are not effect. So, the IC department needs an application for this case.

## Solution:

Develop a **java console application** to **provide information** about study aboard programs to students. This application provides some features for IC's staffs (**only staffs**)

1. Manage aboard programs
2. Manage students
3. Register a program for a student
4. Report

The program is implemented based on the terminology of the Object-Oriented Programming (OOP) paradigm.

## Function specifications

### Function 1. Manage aboard programs – 100 LOC

After **chosen the option 1** (manage aboard programs): a **sub menu** will be **displayed**:

1. **Displays** all aboard programs
2. **Add** a new aboard program
3. **Edit** information a program by **id**
4. **Search** and **display** a program by **id**
5. **Back** to main menu

**Displays all aboard programs:** display information of a program (**see the example above**) [25 LOC]

**Add a new aboard program**: The user enters detail of the program and then **the system adds** it to the list of programs. Data must validate as the following: [25 LOC]

| Fields | Constraints |
|---|---|
| Id | Is **not duplicate, not empty** |
| name | **Not** empty |
| time | **Only accept** as: January, March, May, July, September, November |
| From Registration Date | **Begin** date to register |
| End Registration Date | **End** date to register |
| days | From **30 to 40 days** |
| location | **Not** empty |
| cost | **Not** empty |
| content | **Not** empty, only enter the **path of the file**( **.doc** or **.pdf**) |

**Edit information a program by id:** The user **enters** the program's **id**, the system will allow he/she **edit** its **detail** (except its id). [25 LOC]

**Hint**: **must validate** data as the table above

**Search and view a program by id**: the staff **enters** an **id**, the system **will display information** of this program. [25 LOC]

### Function 2. Manage students – 100 LOC

After **chosen the option 2** (manage students): a sub menu will be displayed:

1. **Displays** all students
2. **Add** a new student
3. **Edit** information a **student** by **id**
4. **Back** to main menu

**Display all students:** display **detail** of students as the following: [20 LOC]

```
ID      NAME   MAJOR   EMAIL                  PHONE  PASSPORT  ADDRESS
SE123  abc     SE      abcSE123@fpt.edu.vn   123        VN12345    1/1 abc
```

**Add a new student:** The staff **enters detail** of a **student**, and **then** the application will **add** he/she to the student **list** [40 LOC]

**Edit information by id**: The staff **enters** the student's **id** and then user can **edit** all fields of this student **except** the student's **id**. Data **must validate** as the following: [40 LOC]

| Fields | Constraints |
|---|---|
| Id | Is **not duplicate, not empty** |
| name | **Not** empty |
| major | Only **accept** as:  SE,SB,GD,MC |
| email | Contains **“@fpt.edu.vn”** |
| phone | **Not** empty |
| passport | **Not** empty |
| address | **Not** empty |

### Function 3. Register a program for students-100 LOC

There is the folder named “**RegistrationForm” in the HDD.** It is used to **save all text files** that are the result of the function registration.

After **chosen the option 3**. The **staff** will do:

- **Enters** the **programs’ id**, the **student’s id**.
- If they **exist**, the **user** will **enter**:

| | |
|---|---|
| **Registration date**: | <<enter from the user>> |
| **Email** of the **parents**: | <<enter from the user>> |
| **Phone** of the **parents**: | <<enter from the user>> |

**Validation: registration date must be between begin and end of the program’s registration date; email, phone is not empty, if the error occurs, the message will display.**

- If typed data is **valid**, the result of the registration will **save** to the **text file(.doc or .txt) that** is named format as  “**StudentID_ProgramID.doc”**. It looks like as the following:

**Aboard Program Registration Form**

**Information Student:**

Student id:  SE123        Student name: Thi No

Major: SE        Email: NotSE123@fpt.edu.vn        Phone:12345        Passport:VN123

Address: 1/1 ABC        Email of the parents:   ChiPheo@gmail.com    Phone of the parents:123

**Information of the aboard program:**

Program’s id: E60        Program’s name: English 6 oversea

Time: March        Days: 30    Location:   Malaysia     Cost:1000$

**Information of the registration:**

**registration date:**        12/4/2023

*<< SE123_E60.doc>>*

### Function 4.  Report – 100 LOC

After **chosen the option 4** (report): a **sub menu** will be displayed:

**1.Print out** the registration by **student's id**
**2.Print out** the students that registered **more than 2 programs**
3.**Count** students that registered the program
4.**Back** to main menu

**Print out the registration by student's id:** The user **enters** the student's **id**. The system will **display** the **content of the text file** whose **filename** contains the **entered id**. **For example**: if the entered id is SE123, the system print out the content of the file "SE123_xxx.doc or SE123_xxx.txt). [50 LOC]

**Print out the students that registered more than 2 programs**: **display information** of the **students** that have **more than 2 the text file**s in the "**RegistrationForm**" .folder [30 LOC]

**Count students that registered the program:** the user **enters** the program's **id**. The system will **print out count of students** that **registered** (**count filenames** that contains the entered id). [20 LOC]

**Hint**: you can use the File class and the method list() to take the filenames of any path.

The above specifications are only **basic information**; you must perform a **requirements analysis step and build the application according to real requirements.**

The lecturer will explain the requirement only once in the first slot of the assignment.

The end