

Học viện Công nghệ Bưu chính Viễn Thông

=====

BÁO CÁO BÀI TẬP LỚN

Môn Lập trình Hướng đối tượng

Đề tài : Hệ thống quản lý thiết bị trong phòng học

Nhóm học phần : NHÓM 1

Họ và tên : Hoàng Đình Lộc

Mã sinh viên : B18DCVT274

HÀ NỘI, 2021

Mục lục

Chương I: Mở đầu.....	3
I. Giới thiệu ứng dụng.....	3
II. Các yêu cầu đối với ứng dụng.....	7
Chương II: Thiết kế.....	8
I. Use case tổng quan toàn hệ thống.....	8
II. Use case chi tiết toàn hệ thống.....	8
III. Biểu đồ lớp thực thể pha thiết kế.....	10
IV. Biểu đồ tuần tự pha thiết kế.....	11
V. Cấu trúc dữ liệu khi lưu vào File.....	14
Chương III: Kết thúc ứng dụng.....	14
I. Các bước cài đặt.....	14
II. Điểm hạn chế của ứng dụng.....	27
III. Tài liệu tham khảo.....	27

Chương I: Mở đầu

I. Giới thiệu ứng dụng

1. Phạm vi phần mềm

- Hệ thống quản lý thiết bị trong phòng học.
- Là Desktop-based application và chỉ được sử dụng trong nội bộ.
- Chỉ có nhân viên quản lý có thể sử dụng ứng dụng này.
- Quản lý các hoạt động:
 - Tạo phòng học mới.
 - Tạo thiết bị mới.
 - Thêm thiết bị cho phòng học.
 - Sắp xếp và tìm kiếm thiết bị trong phòng học.

2. Ai? Vào hệ thống làm gì?

- Nhân viên quản lý:
 - Thêm thông tin về các thiết bị trong phòng học.
 - Tìm kiếm thông tin về thiết bị hoặc phòng học.
 - Kiểm tra cơ sở vật chất của phòng học.

3. Các chức năng hoạt động như nào?

- Nhân viên quản lý:
 - Thêm/Xóa thiết bị.
 - Thêm/Sửa phòng học.
 - Quản lý các thiết bị trong phòng học.
- Chức năng thêm thiết bị:
 - Tại giao diện chính, nhân viên chọn tab “Thông tin thiết bị”.Giao diện hiện lên như hình 1.1:

Hình 1.1: Giao diện tab thông tin thiết bị.

- Nhân viên quản lý click vào nút “Thêm”, con trỏ nhập văn bản xuất hiện ở mục “Ten thiết bị”, đồng thời nút “Cập nhật” và “Bỏ qua” được kích hoạt có thể sử dụng được.
- Nhân viên nhập tên thiết bị, chọn xuất xứ (“Nhập khẩu, Liên doanh, Nội địa”) và nhập năm sản xuất.
- Nếu nhân viên muốn lưu thông tin mới, click vào nút “Cập nhật”, bảng danh sách sẽ xuất hiện thông tin đó, sau đó nhân viên click vào nút “Lưu”, một thông báo “Lưu thành công” xuất hiện kết thúc quá trình thêm thông tin cho thiết bị mới.
- Nếu nhân viên nhấn vào nút “Bỏ qua” thì quá trình nhập thông tin sẽ kết thúc và thông tin đang nhập sẽ không được lưu.
- Chức năng xóa thiết bị:
 - Tại giao diện “Thông tin thiết bị”, nhân viên nhấn vào nút “Hiện thị”, danh sách các thiết bị tồn tại sẽ xuất hiện ở bảng bên tay phải.
 - Nhân viên click vào dòng chứa thông tin thiết bị muốn xóa.
 - Nhân viên click vào nút “Xóa”, sau đó nhấn vào nút “Lưu” để kết thúc chức năng xóa thiết bị.
- Chức năng thêm phòng học:

- Tại giao diện chính, nhân viên chọn tab “Thông tin phong học”.
Giao diện hiện lên như hình 1.2:

He Thong Quan Ly Thiet Bi Trong Phong Hoc

Thông tin thiết bị | **Thông tin phong học** | Quản lý thiết bị phong học

Ma phong học:

Ten phong học:

So ghe:

Ma phong học	Ten phong học	So ghe
--------------	---------------	--------

Them | Sua | Cap nhat | Bo qua | Luu | Hien thi

Hình 1.2: Giao diện tab thông tin thiết bị.

- Nhân viên quản lý click vào nút “Them”, con trỏ nhập văn bản xuất hiện ở mục “Ten phong học”, đồng thời nút “Cap nhat” và “Bo qua” được kích hoạt có thể sử dụng được.
 - Nhân viên nhập tên phòng học và số ghế.
 - Nếu nhân viên muốn lưu thông tin mới, click vào nút “Cap nhat”, bảng danh sách sẽ xuất hiện thông tin đó, sau đó nhân viên click vào nút “Luu”, một thông báo “Luu thanh cong” xuất hiện kết thúc quá trình thêm thông tin cho phòng học mới.
 - Nếu nhân viên nhấn vào nút “Bo qua” thì quá trình nhập thông tin sẽ kết thúc và thông tin đang nhập sẽ không được lưu.
- Chức năng sửa phòng học:
- Tại giao diện “Thông tin phong học”, nhân viên nhấn vào nút “Hien thi”, danh sách các thiết bị tồn tại sẽ xuất hiện ở bảng bên tay phải.

- Nhân viên quản lí click vào một dòng cần sửa, sau đó nhấn vào nút “Sua”. Con trỏ nhập văn bản trở về mục nhập “Ten phong học”.
- Nhân viên sửa lại thông tin cần thiết, sau đó nhấn vào nút “Cap nhât” và nút “Luu” lần lượt, kết thúc chức năng sửa thông tin phòng học.
- Chức năng quản lý thiết bị phòng học:
 - Tại giao diện chính, nhân viên chọn tab “Quan ly thiet bi phong hoc”. Giao diện hiện lên như hình 1.3:

Hình 1.3: Giao diện quản lý thiết bị phòng học.

- Nhân viên nhấn vào nút “Hien thi”, bảng bên tay phải sẽ hiển thị danh sách các thiết bị có trong các phòng học đã tồn tại.
- Nhân viên click vào nút “Them”, mục “Thiet bi” được gợi ý, đồng thời nút “Cap nhât” và “Bo qua” được kích hoạt có thể sử dụng được.
- Nhân viên chọn thông tin cho Thiết bị, phòng học, sau đó nhập số lượng, tình trạng thiết bị.
- Nhân viên click vào nút “Cap nhât” và nút “Luu” lần lượt, kết thúc chức năng thêm mới thiết bị cho một phòng.

- Nếu nhân viên nhấn vào nút “Bo qua” thì quá trình nhập thông tin sẽ kết thúc và thông tin đang nhập sẽ không được lưu.
- Nhân viên có thể sắp xếp danh sách theo tên phòng học hoặc số lượng thiết bị bằng cách nhấn vào nút “Ten thiet bi” hoặc “So luong thiet bi”.
- Nhân viên có thể tìm kiếm thiết bị hoặc phòng học bằng cách điền thông tin vào mục “Nhap ten thiet bi” hoặc “Nhap ten phong hoc”, sau đó nhấn vào nút tìm kiếm, bảng đánh sách bên tay phải sẽ xuất hiện thông tin liên quan.

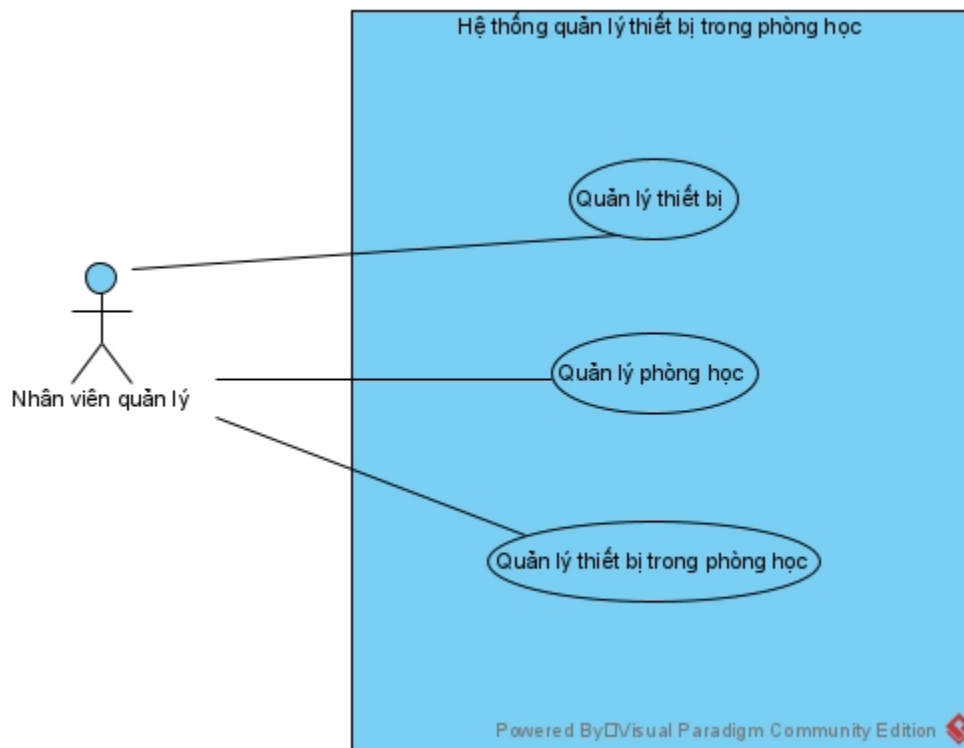
II. Các yêu cầu đối với ứng dụng

- Tên ứng dụng: Quản lý thiết bị trong phòng học.
- Thông tin về thiết bị gồm các thuộc tính (mã thiết bị, tên, xuất xứ, năm sản xuất) – trong đó xuất xứ có thể là: nhập khẩu, nội địa, liên doanh. Mã thiết bị là một số nguyên có 5 chữ số, tự động tăng.
- Thông tin về phòng học (mã phòng học, tên phòng, số ghế) , mã phòng học là số nguyên có 5 chữ số, tự động tăng.
- Bảng Quản lý thiết bị trong phòng học sẽ nhận được nhiều loại thiết bị, mỗi loại có số lượng khác nhau và ghi rõ tình trạng hiện thời.
- Tự xác định các lớp cho phù hợp và viết chương trình có giao diện (có thể dùng kéo thả) thực hiện các chức năng sau:
 - Nhập thêm thiết bị vào file TB.TXT. Có xử lý ngoại lệ. Hiện thị ra bảng danh sách thiết bị đã có trong file.
 - Nhập thêm phòng học vào file PH.TXT . Có xử lý ngoại lệ. Hiện thị ra bảng danh sách phòng học đã có trong file.
 - Lập bảng Quản lý thiết bị cho các phòng học bằng cách nhập các loại thiết bị cho mỗi phòng cùng số lượng và tình trạng tương ứng, lưu vào file QLTB.TXT và in danh sách ra màn hình (chú ý: cùng một loại thiết bị trong phòng thì không thể xuất hiện 2 lần).
- Sắp xếp danh sách Quản lý thiết bị đã lưu trong QLTB.TXT
 - Theo tên thiết bị.
 - Theo số lượng thiết bị mỗi loại (giảm dần).
- Tìm kiếm và hiển thị danh sách quản lý theo tên thiết bị và tên phòng (gần đúng).

Chương II: Thiết kế

I. Use case tổng quan toàn hệ thống

- Actor chính: Nhân viên quản lý.
- Nhân viên quản lý có thể thực hiện các chức năng sau:
 - o Quản lý thông tin thiết bị.
 - o Quản lý thông tin phòng học.
 - o Quản lý thiết bị sử dụng trong phòng học.

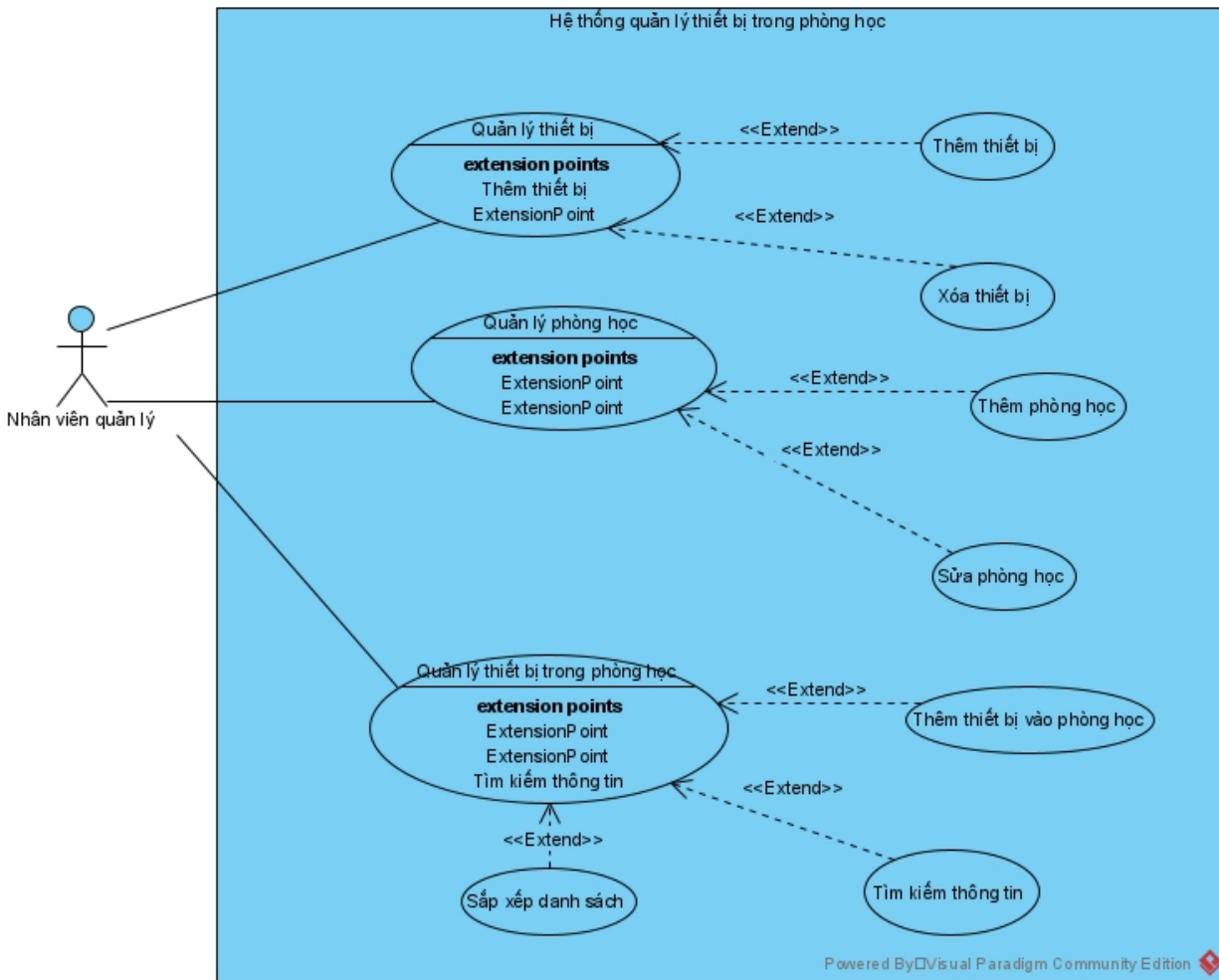


Biểu đồ 2.1: Use case tổng quan toàn hệ thống

Use case mô tả như sau:

- **Quản lý thiết bị:** Cho phép nhân viên quản lý thêm, xóa thông tin thiết bị.
- **Quản lý phòng học:** Cho phép nhân viên quản lý thêm, sửa thông tin của phòng học.
- **Quản lý thiết bị trong phòng học:** Cho phép nhân viên quản lý thêm thiết bị vào trong phòng, sắp xếp danh sách và tra cứu thông tin.

II. Use case chi tiết toàn hệ thống

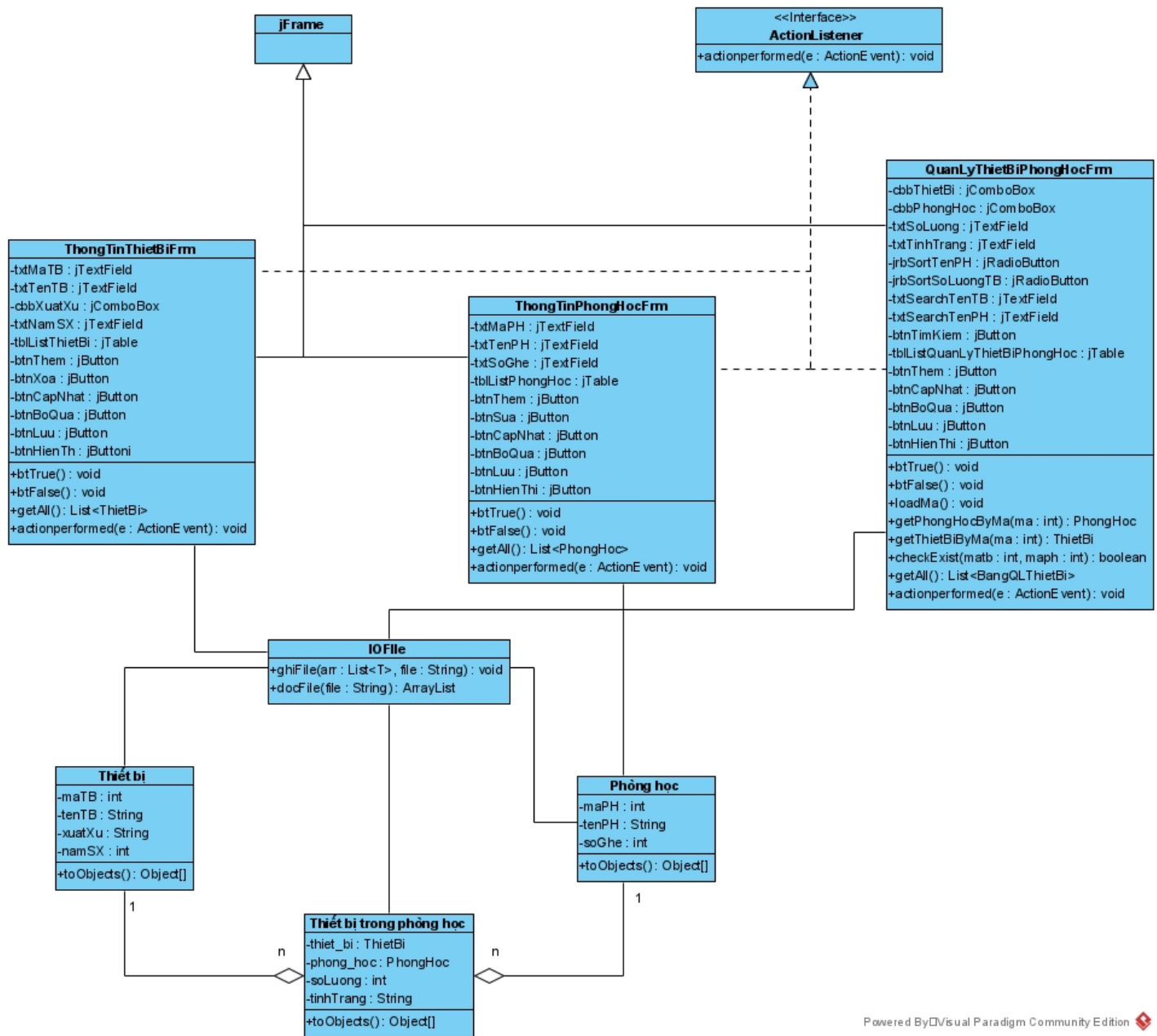


Biểu đồ 2.2: Use case chi tiết toàn hệ thống.

Use case mô tả như sau:

- **Thêm thiết bị:** Cho phép nhân viên quản lý thêm mới thiết bị.
- **Xóa thiết bị:** Cho phép nhân viên quản lý xóa thiết bị.
- **Thêm phòng học:** Cho phép nhân viên quản lý tạo phòng học mới.
- **Sửa phòng học:** Cho phép nhân viên sửa thông tin của phòng học.
- **Thêm thiết bị vào phòng học:** Cho phép nhân viên quản lý quản lý thông tin thiết bị trong phòng học.
- **Sắp xếp danh sách:** Cho phép nhân viên quản lý sắp xếp danh sách theo tên phòng học hoặc theo số lượng thiết bị.
- **Tìm kiếm thông tin:** Cho phép nhân viên quản lý tìm kiếm thông tin phòng có thiết bị được sử dụng.

III. Biểu đồ lớp thực thể pha thiết kế

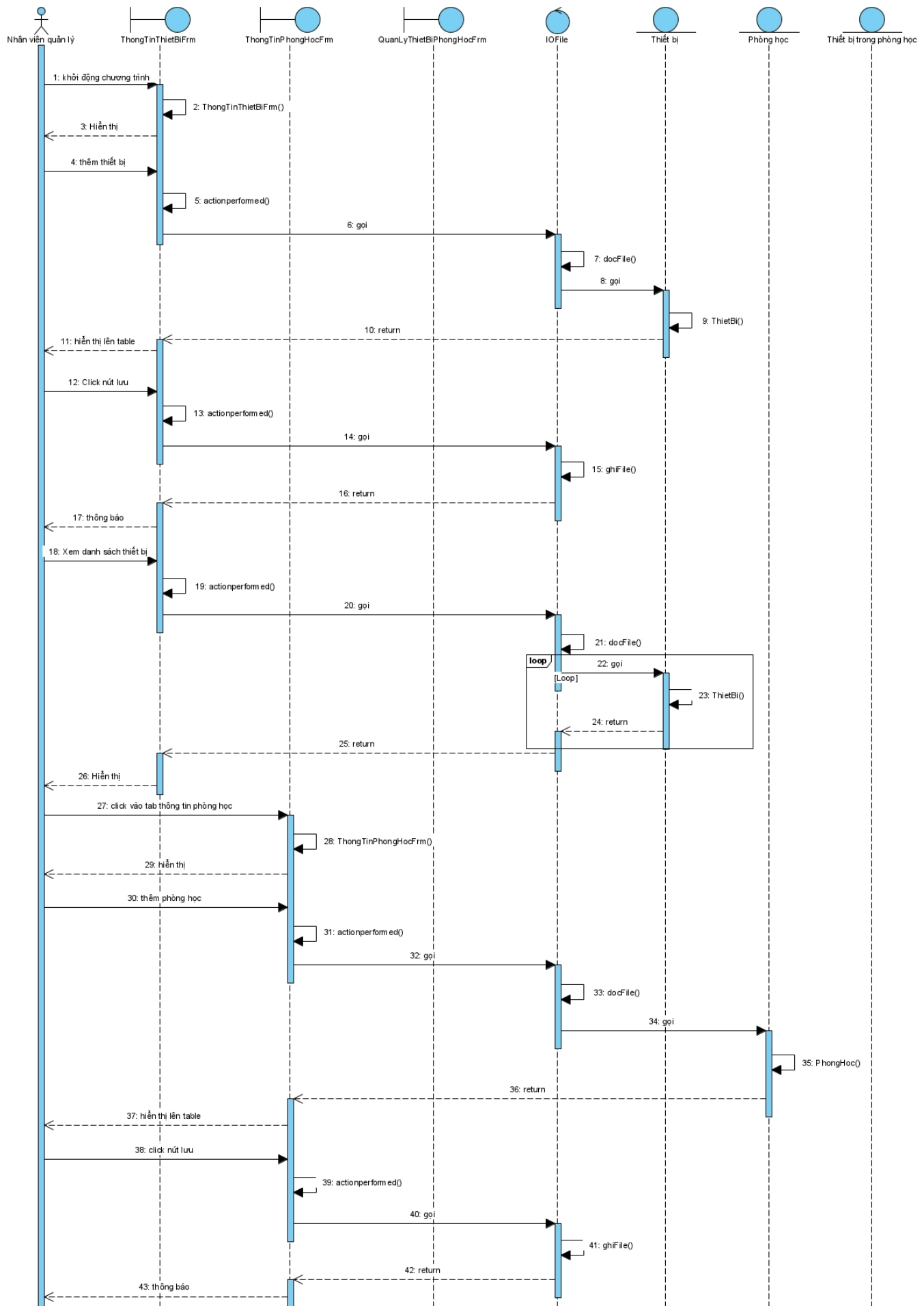


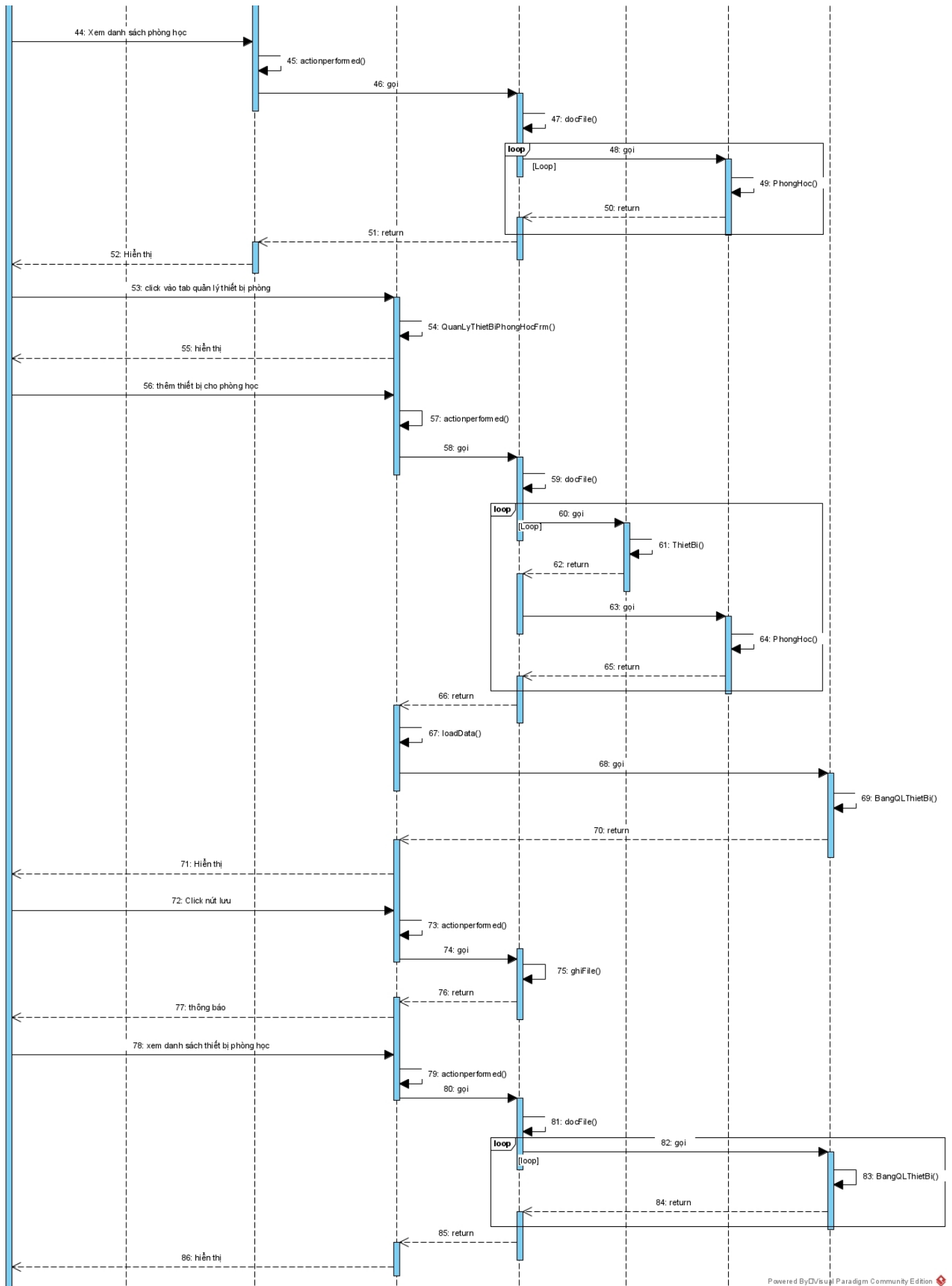
Biểu đồ 2.3: Biểu đồ lớp thực thể pha thiết kế.

- Lớp giao diện:
 - o ThôngTinThietBiFrm là giao diện quản lý thông tin thiết bị của nhân viên quản lý bao gồm các mục nhập tên thiết bị, xuất xứ, năm sản xuất, bảng danh sách thiết bị và 6 nút chức năng thêm, xóa, cập nhật, bỏ qua, lưu và hiển thị.

- ThôngTinPhongHocFrm là giao diện quản lý thông tin phòng học của nhân viên quản lý bao gồm các mục nhập tên phòng học, số ghế, bảng danh sách phòng học và 6 nút chức năng thêm, sửa, cập nhật, bỏ qua, lưu và hiển thị.
- QuanLyThietBiPhongHocFrm là giao diện quản lý thiết bị trong phòng học của nhân viên quản lý bao gồm một danh sách thông tin, mục chọn thiết bị, phòng học cần thêm, mục nhập số lượng và tình trạng hiện tại, mục sắp xếp danh sách, mục nhập tên phòng học hoặc thiết bị tìm kiếm và 5 nút chức năng thêm, cập nhật, bỏ qua, lưu và hiển thị.
- Lớp điều khiển (IOFile): là lớp điều khiển đọc, ghi file giữa lớp thực thể và lớp giao diện.
- Lớp thực thể: Thiết bị, Phòng học. Lớp Thiết bị trong phòng học cần lớp thành phần là lớp Thiết bị và lớp Phòng học.

IV. Biểu đồ tuần tự pha thiết kế





Biểu đồ 2.4 : Biểu đồ tuần tự pha thiết kế.

V. Cấu trúc dữ liệu khi lưu vào File

```
public class IOFile {
    public static <T> void ghiFile(List<T> arr, String file){
        try {
            ObjectOutputStream output = new ObjectOutputStream(new FileOutputStream(file));
            output.writeObject(arr);
            output.close();
        } catch (FileNotFoundException ex) {
            System.out.println(ex);
        } catch (IOException ex) {
            System.out.println(ex);
        }
    }

    public static <T> List<T> docFile(String file) {
        List<T> list = new ArrayList<>();
        try {
            ObjectInputStream input = new ObjectInputStream(new FileInputStream(file));
            list = (List<T>) input.readObject();
            input.close();
        } catch (FileNotFoundException ex) {
            System.out.println(ex);
        } catch (IOException | ClassNotFoundException ex) {
            System.out.println(ex);
        }
        return list;
    }
}
```

Chương III: Kết thúc ứng dụng

I. Các bước cài đặt

1. Các lớp thực thể
 - a. ThietBi.java

```
package Model;
```

```
import java.io.Serializable;
```

```
/**
```

```
 *
```

```
 * @author Hoang Loc
```

```
 */
```

```
public class ThietBi implements Serializable {
```

```
    private int maTB;
```

```
    private String tenTB;
```

```
    private String XuatXu;
```

```
    private int NamSX;
```

```
    private static int currentID = 10000;
```

```
    public ThietBi() {
```

```
        maTB = currentID++;
```

```
    }
```

```
    public ThietBi(int maTB, String tenTB, String XuatXu, int NamSX) {
```

```
        this.maTB = maTB;
```

```
        this.tenTB = tenTB;
```

```
        this.XuatXu = XuatXu;
```

```
        this.NamSX = NamSX;
```

```
    }
```

```
public int getMaTB() {  
    return maTB;  
}
```

```
public void setMaTB(int maTB) {  
    this.maTB = maTB;  
}
```

```
public String getTenTB() {  
    return tenTB;  
}
```

```
public void setTenTB(String tenTB) {  
    this.tenTB = tenTB;  
}
```

```
public String getXuatXu() {  
    return XuatXu;  
}
```

```
public void setXuatXu(String XuatXu) {  
    this.XuatXu = XuatXu;  
}
```



```
public int getNamSX() {  
    return NamSX;  
}
```

```
public void setNamSX(int NamSX) {  
    this.NamSX = NamSX;  
}
```

```
public static int getCurrentID() {  
    return currentID;  
}
```

```
public static void setCurrentID(int currentID) {  
    ThietBi.currentID = currentID;  
}
```

```
public Object[] toObjects() {  
    return new Object[] {  
        maTB, tenTB, XuatXu, NamSX  
    };  
}
```

b. PhongHoc.java

```
package Model;
```

```
import java.io.Serializable;
```

```
/**
```

```
 *
```

```
 * @author Hoang Loc
```

```
 */
```

```
public class PhongHoc implements Serializable {
```

```
    private int maPH;
```

```
    private String tenPH;
```

```
    private int soGhe;
```

```
    private static int currentID = 10000;
```

```
    public PhongHoc() {
```

```
        maPH = currentID++;
```

```
    }
```

```
    public PhongHoc(int maPH, String tenPH, int soGhe) {
```

```
        this.maPH = maPH;
```

```
        this.tenPH = tenPH;
```

```
        this.soGhe = soGhe;
```

```
    }
```

```
public int getMaPH() {  
    return maPH;  
}
```

```
public void setMaPH(int maPH) {  
    this.maPH = maPH;  
}
```

```
public String getTenPH() {  
    return tenPH;  
}
```

```
public void setTenPH(String tenPH) {  
    this.tenPH = tenPH;  
}
```

```
public int getSoGhe() {  
    return soGhe;  
}
```

```
public void setSoGhe(int soGhe) {  
    this.soGhe = soGhe;  
}
```

```
public static int getCurrentID() {
```

```

        return currentID;
    }

    public static void setCurrentID(int currentID) {
        PhongHoc.currentID = currentID;
    }

    public Object[] toObjects() {
        return new Object[] {
            maPH, tenPH, soGhe
        };
    }
}

```

c. BangQLThietBi.java

```

package Model;

import java.io.Serializable;

/**
 *
 * @author Hoang Loc
 */
public class BangQLThietBi implements Serializable {
    private ThietBi thiet_bi;

```

```
private PhongHoc phong_hoc;
```

```
private int soLuong;
```

```
private String tinhTrang;
```

```
public BangQLThietBi() {
```

```
}
```

```
public BangQLThietBi(ThietBi thiet_bi, PhongHoc phong_hoc, int soLuong,  
String tinhTrang) {
```

```
    this.thiet_bi = thiet_bi;
```

```
    this.phong_hoc = phong_hoc;
```

```
    this.soLuong = soLuong;
```

```
    this.tinhTrang = tinhTrang;
```

```
}
```

```
public ThietBi getThiet_bi() {
```

```
    return thiet_bi;
```

```
}
```

```
public void setThiet_bi(ThietBi thiet_bi) {
```

```
    this.thiet_bi = thiet_bi;
```

```
}
```

```
public PhongHoc getPhong_hoc() {
```

```
    return phong_hoc;
```

```
}
```

```
public void setPhong_hoc(PhongHoc phong_hoc) {  
    this.phong_hoc = phong_hoc;  
}
```

```
public int getSoLuong() {  
    return soLuong;  
}
```

```
public void setSoLuong(int soLuong) {  
    this.soLuong = soLuong;  
}
```

```
public String getTinhTrang() {  
    return tinhTrang;  
}
```

```
public void setTinhTrang(String tinhTrang) {  
    this.tinhTrang = tinhTrang;  
}
```

```
public Object[] toObjects() {  
    return new Object[] {  
        phong_hoc.getMaPH(), phong_hoc.getTenPH(), thiet_bi.getTenTB(),  
        thiet_bi.getMaTB(), soLuong, tinhTrang  
    };  
}
```

```
}  
}
```

2. Lớp điều khiển

IOFile.java

```
package Controller;
```

```
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.ObjectInputStream;  
import java.io.ObjectOutputStream;  
import java.util.ArrayList;  
import java.util.List;
```

```
/**
```

```
 *
```

```
 * @author Hoang Loc
```

```
 */
```

```
public class IOFile {
```

```
    public static <T> void ghiFile(List<T> arr, String file){
```

```
        try {
```

```
            ObjectOutputStream output = new ObjectOutputStream(new  
FileOutputStream(file));
```

```
        output.writeObject(arr);
        output.close();
    } catch (FileNotFoundException ex) {
        System.out.println(ex);
    } catch (IOException ex) {
        System.out.println(ex);
    }
}
```

```
public static <T> List<T> docFile(String file) {
    List<T> list = new ArrayList<>();
    try {
        ObjectInputStream input = new ObjectInputStream(new
FileInputStream(file));
        list = (List<T>) input.readObject();
        input.close();
    } catch (FileNotFoundException ex) {
        System.out.println(ex);
    } catch (IOException | ClassNotFoundException ex) {
        System.out.println(ex);
    }
    return list;
}
}
```


3. Lớp giao diện

a. Main.java

```
public class Main extends javax.swing.JFrame {  
  
    /**  
     * Creates new form Main  
     */  
    public Main() {  
        initComponents();  
        setTitle("He Thong Quan Ly Thiet Bi Trong Phong Hoc");  
        setSize(700, 500);  
        FormThietBi p1 = new FormThietBi();  
        jTabbedPane1.add("Thong tin thiet bi", p1);  
        FormPhongHoc p2 = new FormPhongHoc();  
        jTabbedPane1.add("Thong tin phong hoc", p2);  
        FormQLTB p3 = new FormQLTB();  
        jTabbedPane1.add("Quan ly thiet bi phong hoc", p3);  
    }  
}
```

b. FormThietBi.java

He Thong Quan Ly Thiet Bi Trong Phong Hoc

Thong tin thiet bi | Thong tin phong hoc | Quan ly thiet bi phong hoc

Ma thiet bi:

Ten thiet bi:

Xuat xu:

Nam san xuat:

Ma thiet bi | Ten thiet bi | Xuat xu | Nam san x...

Them Xoa Cap nhat Bo qua Luu Hien thi

c. FormPhongHoc.java

He Thong Quan Ly Thiet Bi Trong Phong Hoc

Thong tin thiet bi | **Thong tin phong hoc** | Quan ly thiet bi phong hoc

Ma phong hoc	Ten phong hoc	So ghe
--------------	---------------	--------

Ma phong hoc:

Ten phong hoc:

So ghe:

Them Sua Cap nhat Bo qua Luu Hien thi

d. FormQLTB.java

He Thong Quan Ly Thiet Bi Trong Phong Hoc

Thong tin thiet bi | Thong tin phong hoc | **Quan ly thiet bi phong hoc**

Thiet bi: 10000

Phong hoc: 10000

So luong:

Tinh trang:

Sap xep theo: ☐ Ten phong hoc ☐ So luong thiet bi

Nhap ten thiet bi:

Nhap ten phong hoc:

Timkiem

Ma pho...	Ten pho...	Ten thie...	Ma thiet bi	So luong	Tinh trang
-----------	------------	-------------	-------------	----------	------------

Them Cap nhat Bo qua Luu Hien thi

II. Điểm hạn chế của ứng dụng

Chức năng tìm kiếm của ứng dụng vẫn còn hạn chế do chuỗi string truyền vào chưa được chuẩn hóa, do đó danh sách chỉ hiện ra kết quả chính xác tương ứng với người nhập.

Tính năng tìm kiếm gần đúng là một tính năng khó, đòi hỏi kỹ thuật cao và sự am hiểu, ở thời điểm hiện tại người viết báo cáo này vẫn chưa thể giải được vấn đề đó. Vì vậy, chức năng này mới chỉ dừng ở mức tìm kiếm chính xác. Hy vọng rằng trong tương lai tôi có thể khắc phục được vấn đề này.

III. Tài liệu tham khảo

1. Giáo trình lập trình hướng đối tượng - Nguyễn Thị Vân Anh.
2. Giáo trình lập trình hướng đối tượng – Nguyễn Hoàng Anh.