**Trường Đại Học Bách Khoa Hà Nội**

**2020-2021**

***Báo cáo Lập trình hướng đối tượng***

***Đề tài số 12***

***Quản lý cán bộ bệnh viên***

***(Java swing + MySql)***

Môn học: **Lập trình hướng đối tượng**

Giảng viên:

Mã lớp:

**Nhóm thực hiện**

1. **Hoàng Văn Nguyên - 20180146**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Name | Student ID | Email | Work assignment | Completion | Note |
| 01 | Nguyễn Văn A | 01200210 |  | - Overview OOP in Java + Mysql  - Coding the first function Model  - Write Report  - Discuss and divide work for all members - Report for OOP, java + Mysql | - 100% |  |
| 02 | Nguyễn văn B | 354603134 |  | - Research for java swing, - Coding the second function Controller  - Report for The functions of the system | -100% | IDE:NetBean |
| 03 | Nguyễn văn C | 354603134 |  | - Research for MySql, - Coding the third function View - Report for the system's interface | -100% | CSDL:MySql |

**MỤC LỤC**

[**I.Kiến thức nền tảng 3**](#_Toc18293)

[**II. Hospital Staff Management System**](#_Toc18294) **6**

[**1. System Introduce**](#_Toc18295) **6**

[**2. Model**](#_Toc18296) **8**

[**3. Controller**](#_Toc18298) **10**

[**4. View**](#_Toc18300) **11**

[**III.** **Kết quả**](#_Toc18306) **14**

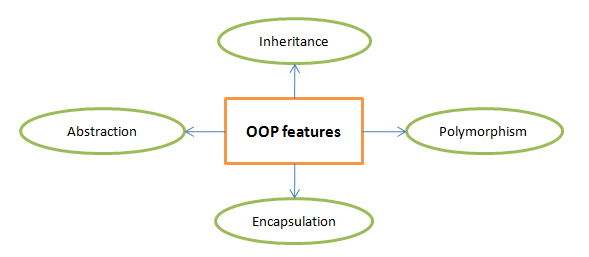
[**IV. Lời kết**](#_Toc18318) **15**

[**V. Tài liệu tham khảo**](#_Toc18319) **15**

**I.Kiến thức nền tảng**

**1. Java:**

**Lập trình hướng đối tượng (OOP-Object-Oriented Programming)** là một phương pháp hay mô hình giúp tăng năng suất, đơn giản hóa việc bảo trì, dễ dàng mở rộng trong thiết kế phần mềm bởi việc cung cấp một vài khái niệm như:

* Đối tượng (Object)
* Lớp (Class)
* Kế thừa (Inheritance)
* Đa hình (Polymorphism)
* Trừu tượng (Abstraction)
* Đóng gói (Encapsulation)
* **Lập trình hướng đối tượng có 4 tính chất:**
* 

### **Giao diện người dùng (Graphical User Interface – GUI)**

Một giao diện người dùng (gọi tắt là GUI) là một cửa sổ (window) chứa các điều khiển (controls) như button, text box, combo box, v.v. Người dùng tương tác với giao diện bằng cách dùng chuột, con trỏ hay bàn phím.

Java cho phép chúng ta tạo giao diện người dùng bằng cách sử dụng một trong hai gói là **AWT** hay **Swing**. **AWT** (Asbtract Window Toolkit) là tập con của **Swing** nên **Swing** được dùng chủ yếu nhưng trong chương trình **Java** phải **import** cả hai gói **Swing** và **AWT**.

## MySQL :

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. [**RDBMS**](https://en.wikipedia.org/wiki/Relational_database_management_system) là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

MySQL là một trong số các phần mềm RDBMS. RDBMS và MySQL thường được cho là một vì độ phổ biến quá lớn của MySQL. [**Các ứng dụng web lớn nhất**](https://stackshare.io/mysql) như Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Kể cả khi ban đầu nó chỉ được dùng rất hạn chế nhưng giờ nó đã tương thích với nhiều hạ tầng máy tính quan trọng như Linux, macOS, Microsoft Windows, và Ubuntu

# Kết nối java với mysql bằng JDBC

JDBC (Java Database Connectivity) là một chuẩn API (Application Program Interface) cho phép kết nối các chương trình viết bởi Java với các hệ quản trị cơ sở dữ liệu (MySQL, MS SQL, Postgre SQL, Oracle, DB2…)

JDBC chỉ là một tập các interface, các định nghĩa, thông báo lỗi, đặc tả chứ không phải là thư viện. Với mỗi hệ quản trị cơ sở dữ liệu ta sẽ có một cài đặt JDBC riêng cho nó, ví dụ JDBC cho MySQL, JDBC cho MS SQL …

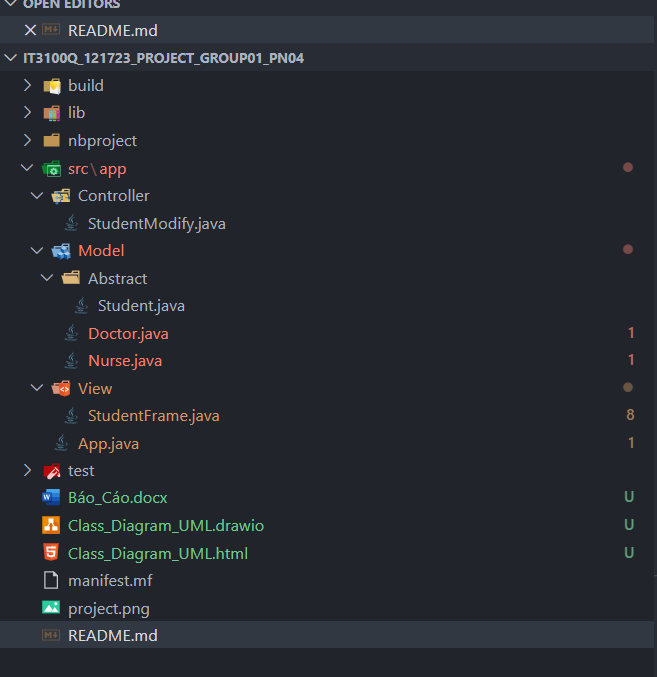
**II.** **Hospital Staff Management System**

**1. System introduce**

**Link source code:** <https://github.com/nguyenhv01012000/IT3100Q_121723_Project_Group01_PN04>

Hệ thống quản lý cán bộ của bệnh viện với các chức năng:  
- Thêm, Bớt, Sửa, Xóa một cán bộ  
- Tìm kiếm cán bộ theo tên, ngày sinh, hoặc đơn vị  
- Tìm kiếm, hiển thị thông tin cán bộ có lương hàng tháng lớn hơn một số nhập  
vào.  
- Thống kê danh sách cán bộ và lương hàng tháng của họ

Cấu trúc file trong project:



* Hệ thống được làm theo mô hình MVC.

**MVC** được viết tắt bởi 3 từ: **M**odel – **V**iew – **C**ontroller, là một mẫu thiết kế nhằm mục tiêu chia tách phần giao diện và code để dễ quản lý, phát triển và bảo trì.  
  
[**Mô hình MVC**](https://niithanoi.edu.vn/mo-hinh-mvc-la-gi.html) chia một ứng dụng thành ba phần tương tác được với nhau, mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

* Model: quản lý, xử lý dữ liệu
* View: giao diện, hiển thị dữ liệu cho người dùng
* Controller: điều khiển sự tương tác giữa Model và View

### Technologies

* **JAVA SWING**: xây dựng giao diện, tham khảo: [docs](https://www.javatpoint.com/java-swing)
* **MYSQL**: cung cấp cơ chế lưu trữ và quản lý cơ sở dữ liệu quan hệ(SQL) một cách thuận tiện và dễ dàng, tham khảo: [docs](https://github.com/nguyenhv01012000/IT3100Q_121723_Project_Group01_PN04/blob/main/w3schools.com/sql)

### Install

* IDE: Netbean,
* CSDL: MySql

### Runs

1. Create name's database in MySql : mydatabase
2. if user = root, password = nguyen in MySql : (StudentModify.java)
3. connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase", "root", "nguyen");
4. else edit respectively ( user = “ …..” , password = “ …” )
5. connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase", "....", "......");
6. Run project: run App.java

**--> App will run**

**App.java**

public class App {

*/\*\**

*\* @param args the command line arguments*

*\*/*

    public static void main(String[] *args*) {

*// TODO code application logic here*

        StudentFrame view = new StudentFrame();

        view.show();

    }

}

**2. Model**

* Có nhiệm vụ quản lý dữ liệu của ứng dụng
* Chứa tất cả các nghiệp vụ logic, đối tượng mô tả dữ liệu
* Thông báo cho view để hiện thị lại kết quả cho người dùng

Trong thư mục Model:

* Trừu tượng (Abstraction)

**Student.java**

public class Student {

    int id;

    String fullname, gender, position, birthday, hospital, phoneNumber, working, surgery, night, salary;

    public Student() {

    }

    public Student(int *id*, String *fullname*, String *gender*, String *position*, String *birthday*, String *hospital*,

            String *phoneNumber*, String *working*, String *surgery*, String *night*, String *salary*) {

        super();

        this.id = *id*;

        this.fullname = *fullname*;

        this.gender = *gender*;

        this.position = *position*;

        this.birthday = *birthday*;

        this.hospital = *hospital*;

        this.phoneNumber = *phoneNumber*;

        this.working = *working*;

        this.surgery = *surgery*;

        this.night = *night*;

        this.salary = *salary*;

    }

……………

* Đóng gói

public int getId() {

        return id;

    }

    public void setId(int *id*) {

        this.id = *id*;

    }

    public String getFullname() {

        return fullname;

    }

    public void setFullname(String *fullname*) {

        this.fullname = *fullname*;

    }

    public String getGender() {

        return gender;

    }

……….

* Kế thừa (Inheritance)

**Doctor.java, Nurse.java**

public class Doctor extends Student{

    public Doctor() {

    }

    public Doctor(int *id*, String *fullname*, String *gender*, String *position*, String *birthday*, String *hospital*, String *phoneNumber*, String *working*, String *surgery*, String *night*, String *salary*) {

        super(*id*, *fullname*, *gender*, *position*, *birthday*, *hospital*, *phoneNumber*, *working*, *surgery*, *night*, *salary*);

    }

    public Doctor(String *fullname*, String *gender*, String *position*, String *birthda*

}

………….

public class Nurse extends Student{

    public Nurse() {

    }

    public Nurse(int *id*, String *fullname*, String *gender*, String *position*, String *birthday*, String *hospital*, String *phoneNumber*, String *working*, String *surgery*, String *night*, String *salary*) {

        super(*id*, *fullname*, *gender*, *position*, *birthday*, *hospital*, *phoneNumber*, *working*, *surgery*, *night*, *salary*);

    }

…………..

* Đa hình (Polymorphism)

    public void setSalary(int *working*, int *surgery*) {

        this.salary = *working*\*500000 + *surgery*\*2000000;

    }

public void setSalary(int *working*, int *night*) {

        this.salary = *working*\*200000 + *night*\*500000;

    }

1. **Controller**

* Định nghĩa các hoạt động, xử lý của hệ thống
* Đối chiếu các hành động của người dùng từ View. Đồng thời tương tác Model để gọi View và hiển thị thông tin tương ứng cho người dùng.

**Modify.java**

public class Modify {

    public static List<Student> findAll() {

        List<Student> studentList = new ArrayList<>();

        Connection connection = null;

        Statement statement = null;

        try {

*// lay tat ca danh sach sinh vien*

            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase", "root", "nguyen");

*// query*

            String sqlsql = "CREATE TABLE IF NOT EXISTS student ( id int primary key auto\_increment,fullname varchar(50) not null, gender varchar(15), position varchar(15),birthday varchar(150), hospital varchar(150),phone\_number varchar(150),working varchar(150),surgery varchar(150),night varchar(150),salary varchar(150))";

            String sql = "select \* from student";

            statement = connection.createStatement();

…………..

**5. View**

* Chứa giao diện, tương tác với người dùng, sử dụng Model để hiển thị kết quả cho người dùng
* Đưa ra kết quả từ tầng Controller
* Thu nhận các hoạt động, yêu cầu của người dùng và chuyển cho tầng Controller xử lý

**Frame.java**

public class StudentFrame extends javax.swing.JFrame {

    DefaultTableModel tableModel;

    List<Student> studentList = new ArrayList<>();

*/\*\**

*\* Creates new form StudentFrame*

*\*/*

    public StudentFrame() {

        initComponents();

        tableModel = (DefaultTableModel) tblStudent.getModel();

        showStudent();

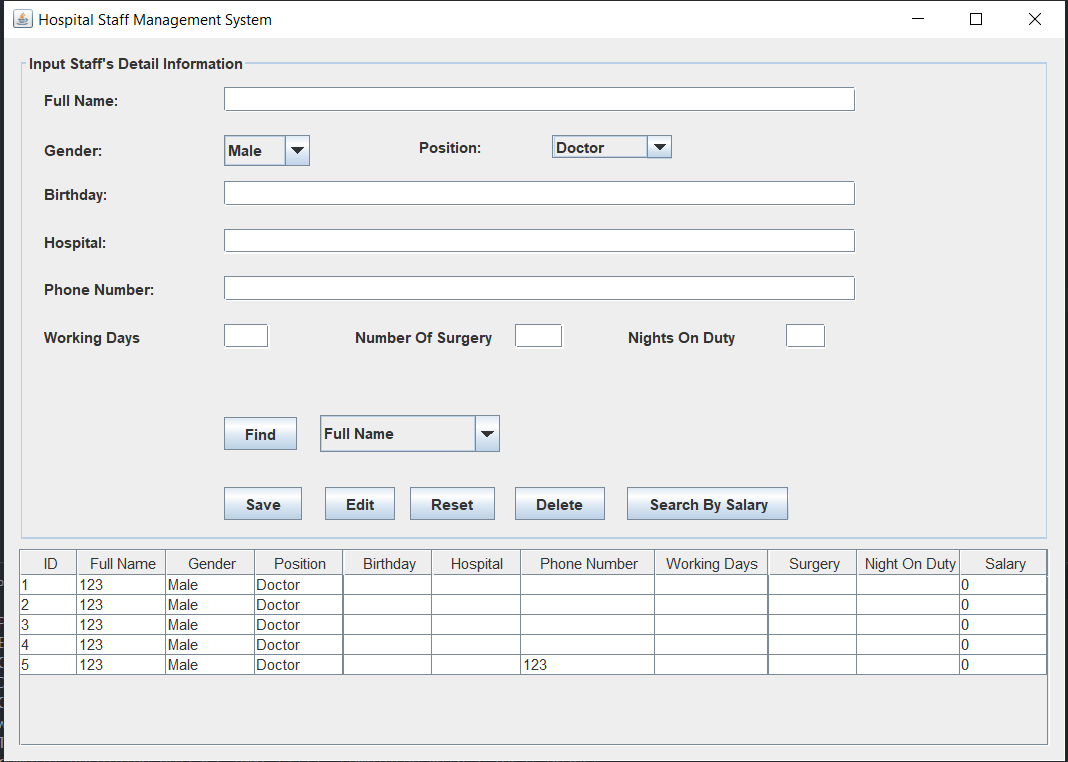
    }

………………….

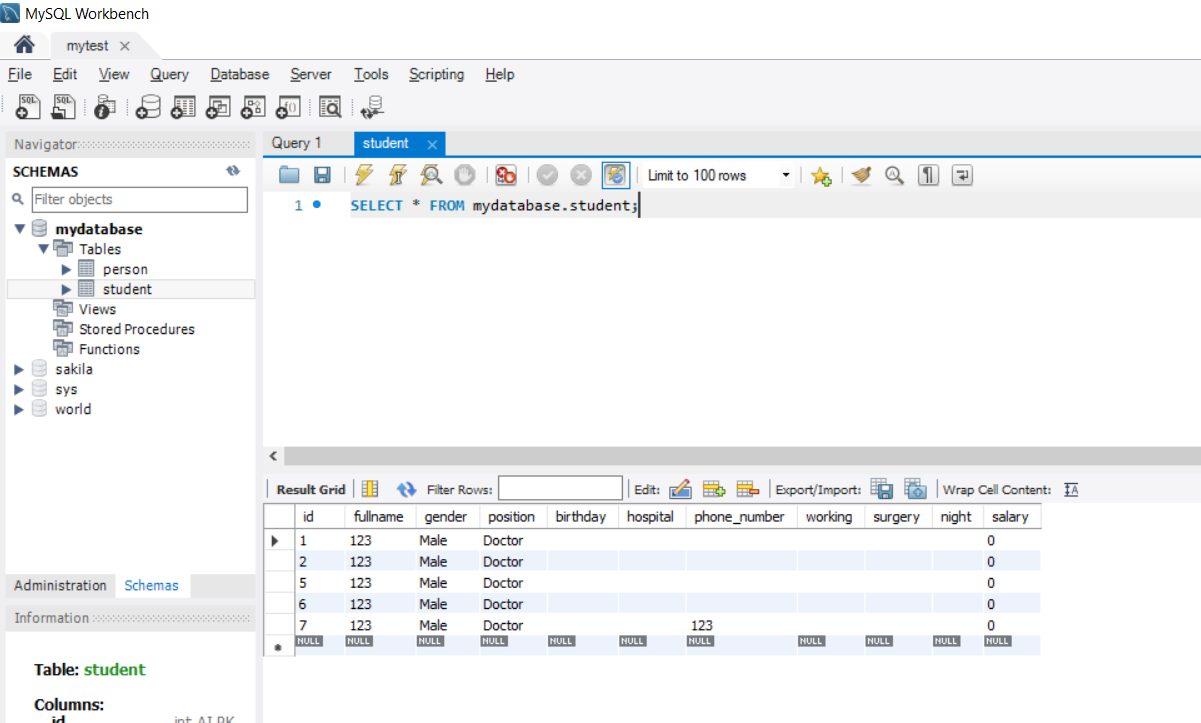
# 

**III. Kết quả**

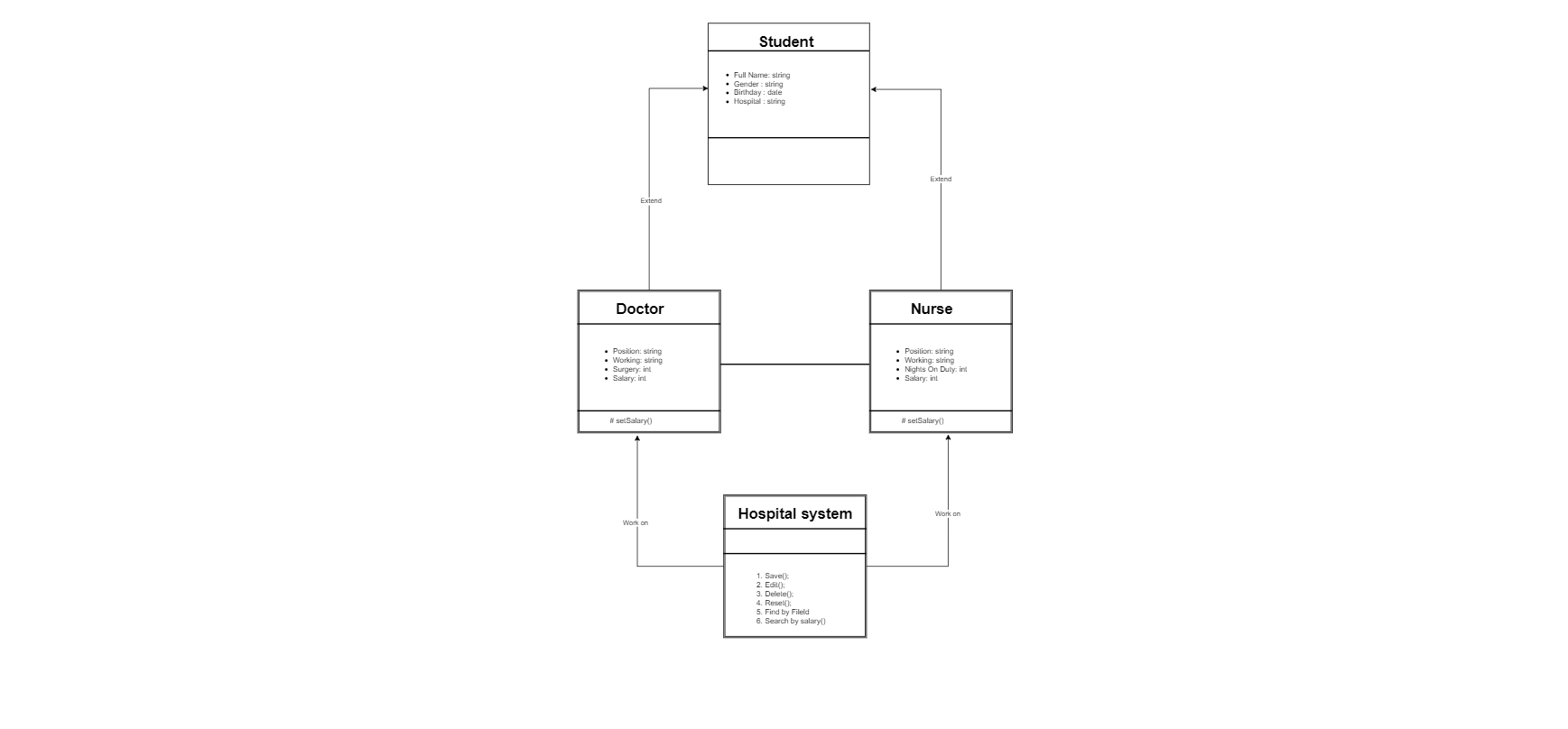
**Khi chạy chương trình sẽ hiển thị ra:**

****

**Dữ liệu nhập vào sẽ được lưu vào cơ sở dữ liệu MySql:**

****

**Class\_Diagram\_UML**

****

**IV. Lời kết**

Với lượng kiến thức vô cùng hạn chế cũng như chưa có nhiều thời gian để tìm hiểu chuyên sâu, báo cáo này không hy vọng mang đến những gì mới mẻ mà đơn giản chỉ là sự tổng hợp, lược dịch và chỉnh sửa lại các bài viết đã có trên internet, sách in và một số tài liệu khác. Sai sót về mặt nội dung cũng như khiếm khuyết trong cách trình bày chắc chắn không thể tránh khỏi, rất mong nhận được sự thông cảm và giúp đỡ của thầy để báo cáo được hoàn thiện hơn!

**V. Tài liệu tham khảo**

* [The JavaTM Tutorials](http://java.sun.com/docs/books/tutorial/index.html) − Loạt bài hướng dẫn về Java giúp lập trình viên tạo các ứng dụng sử dụng ngôn ngữ Java.
* [Sun Developer Network](http://java.sun.com/reference/docs/) − Site chính thức của Sun Microsystem liệt kê tất cả API documentation, Java Technology mới nhất, sách và các nguồn kháchttp://wikipedia.com
* <https://teamvietdev.com/chia-se-bo-tai-lieu-lap-trinh-java-swing/>
* https://www.mysqltutorial.org/
* …