Project: Hospital Review

Project report

Group members:

Phạm Thế Duyệt – ITITIU15071

Trần Trọng Tiến – ITITIU15084

Phan Minh Hưng – ITITIU15073

Trần Duy Bảo – ITITIU15076

Nguyễn Phú Vinh – ITITIU15052

Table of Contents:

1. Overview:

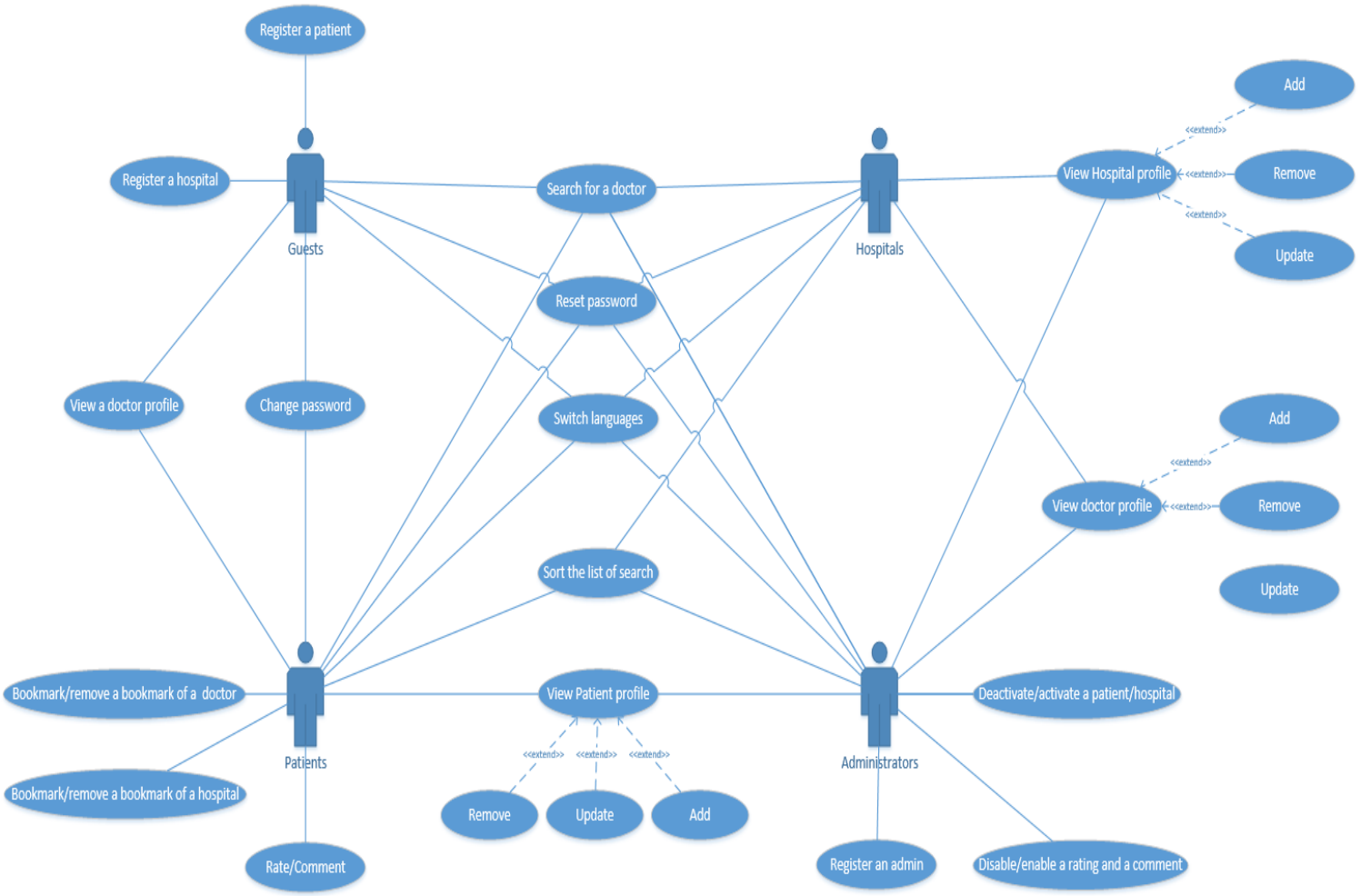
_ Our group are going to develop a hospital review application (website) which uses object-oriented (Java) and scripting programming language (Python, JavaScript) for server-side development. Our project includes some following functions/use cases:

Code	Use Case Name	Category	Stakeholder
UC1	Search for a doctor (without functionalities on rating and commenting on the doctor)	1	Guest, Patient, Hospital, Admin
UC2	Change pasword	2	Guest, Patient,
UC3	Register a patient	3	Guest
UC4	Register a hospital	3	Guest
UC5	Register an admin	3	Admin
UC7	View/Add/Remove/Update Patient profile	4	Admin, Patient
UC8	View/Add/Remove/Update Hospital profile	4	Admin, Hospital
UC9	Rate and Comment a doctor	5	Patient
UC10	Disable/enable a rating and a comment	6	Admin
UC11	Deactivate/activate a patient/hospital	7	Admin
UC12	Switch languages of GUIs	8	Guest, Patient, Hospital, Admin
UC13	Reset password	9	Guest, Patient, Hospital, Admin
UC15	Bookmark/remove a bookmark of a favorite hospital	10	Patient
UC16	Bookmark/remove a bookmark of a favorite doctor	10	Patient
UC17	View a doctor profile	11	Guest/Patient
UC18	Add/Remove/Update a doctor profile	11	Hospital/Admin
UC19	Sort the list of search result by nearest location, doctor name, rating, number of ratings	12	Patient, Hospital and Admin

2. Contribution:

Total	100%
Trần Trọng Tiến	20%
Nguyễn Phú Vinh	20%
Trần Duy Bảo	20%
Phạm Thế Duyệt	20%
Phan Minh Hưng	20%

3. Use case diagram:



4. Evaluate the usage of about data types, control structure, concurrency and exception handling of the selected programming language

- **Java:**

_Data types: In Java, Data type specifies the size and type of values that can be stored in an identifier. The Java language is rich in its data types. Different data types allow us to select the type appropriate to the needs of the application. Data types in Java are classified into two types:

1. Primitive—which include Integer, Character, Boolean, and Floating Point.
2. Non-primitive—which include Classes, Interfaces, and Arrays.

_Control structure: In Java, control statements can be divided into the following three categories: Selection Statements, Iteration Statements, Jump Statements.

+ Selection statements allow you to control the flow of program execution on the basis of the outcome of an expression or state of a variable known during runtime. Selection statements can be divided into the following categories: The if and if-else statements, The if-else statements, The if-else-if statements, The switch statements.

+ Iteration Statements Repeat the same code fragment several times until a specified condition is satisfied is called iteration. Iteration statements execute the same set of instructions until a termination condition is met. Java provides the following loop for iteration statements: The while loop, The for loop, The do-while loop, The for each loop.

+ Jump Statements are used to unconditionally transfer the program control to another part of the program. Java provides the following jump statements: break statement, continue statement, return statement.

_Exception handling: In java, exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime. Exception Handling is a mechanism to handle runtime errors such as ClassNotFound, IO, SQL, Remote etc. The core advantage of exception handling is to maintain the normal flow of the application. Exception normally disrupts the normal flow of the application that is

why we use exception handling. There are mainly two types of exceptions: checked and unchecked where error is considered as unchecked exception. The sun microsystem says there are three types of exceptions: Checked Exception, Unchecked Exception, Error.

- **Python**

_Data types: Python has five standard Data Types: Numbers, String, List, Tuple, Dictionary. Python sets the variable type based on the value that is assigned to it. Unlike more riggers languages, Python will change the variable type if the variable value is set to another value.

+ Numbers: Python numbers variables are created by the standard Python method. Most of the time using the standard Python number type is fine. Python will automatically convert a number from one type to another if it needs. But, under certain circumstances that a specific number type is needed (ie. complex, hexidecimal), the format can be forced into a format by using additional syntax in the table below:

Type	Format	Description
int	a = 10	Signed Integer
long	a = 345L	(L) Long integers, they can also be represented in octal and hexadecimal
float	a = 45.67	(.) Floating point real values
complex	a = 3.14J	(J) Contains integer in the range 0 to 255.

+ String: Create string variables by enclosing characters in quotes. Python uses single quotes ‘double quotes “and triple quotes “”” to denote literal strings. Only the triple quoted strings “”” also will automatically continue across the end of line statement.

+ List: Lists are a very useful variable type in Python. A list can contain a series of values. List variables are declared by using brackets [] following the variable name. All lists in Python are zero-based indexed. When referencing a member or the length of a list the number of list elements is always the number shown plus one.

+ Tuple: Tuples are a group of values like a list and are manipulated in similar ways. But, tuples are fixed in size once they are assigned. In Python the fixed size is considered immutable as compared to a list that is dynamic and mutable. Tuples are defined by parenthesis ().

+ Dictionary: Dictionaries in Python are lists of Key:value pairs. This is a very powerful datatype to hold a lot of related information that can be associated through keys. The main operation of a dictionary is to extract a value based on the key name. Unlike lists, where index numbers are used, dictionaries allow the use of a key to access its members. Dictionaries can also be used to sort, iterate and compare data. Dictionaries are created by using braces ({}) with pairs separated by a comma (,) and the key values associated with a colon (:). In Dictionaries the Key must be unique.

_Control structure: The statements that allow the computer to select or repeat an action. Control statement selects one option among all options and repeats specified section of program.

+Python if statement: Through Python if statement, a program can branch to section of code or just skip it.

+ Python if-else statement: It is also called a two-way selection statement, because it leads the program to make a choice between two alternative courses of action.

+ Python if-elif-else structure: Occasionally, a program contains several testing conditions that involve more than two alternative courses of action. In multi-way if statements program checks each condition until one evaluates to true or all evaluate to false. When a condition evaluates to True, the corresponding action of condition is taking place. If no condition satisfy means evaluate to true then the corresponding action of trailing else is performed.

+Loop: Programming languages provide control statements with repetition statements known as loops, which repeat an action. Two types of loop: Definite loop, Indefinite loop.

+For loop: A for loop is used for iterating over a sequence (that is either a list, a tuple or a string). This is less like the for keyword in other programming language and works more like an iterator method as found in other object-orientated programming languages. With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.

+While loop: With the while loop we can execute a set of statements as long as a condition is true.

+Break Statement: With the break statement we can stop the loop even if the while condition is true.

_Exception handling: Python provides two very important features to handle any unexpected error in your Python programs and to add debugging capabilities in them.

+ Exception Handling − This would be covered in this tutorial. Here is a list standard Exceptions available in Python: Standard Exceptions.

+ Assertions − This would be covered in Assertions in Python tutorial.

5. Example of data types, control structure, concurrency and exception handling of the selected programming language from the source code:

- **Java:**

_Data types:

+ Primitive—which include Integer, Character, Boolean, and Floating Point are used almost in all classes. The following figure is the example from ControlBookmark class in Controller package:

```
switch (action) {
    case "bookmarkdoctor": {
        String paID = request.getParameter("pID");
        String doID = request.getParameter("dID");
        int p_id = Integer.parseInt(paID);
        int d_id = Integer.parseInt(doID);
        bookmark.bookmarkDoctor(p_id, d_id);

        response.sendRedirect("viewdoctor.jsp");
        break;
    }
    case "removebookmarkdoctor": {
        String paID = request.getParameter("pID");
        String doID = request.getParameter("dID");
        int p_id = Integer.parseInt(paID);
        int d_id = Integer.parseInt(doID);
        bookmark.removeBookmarkDoctor(p_id, d_id);
        response.sendRedirect("profileUser.jsp");
        break;
    }
```

+ Non-primitive—which include Classes, Interfaces, and Arrays are used in some DTO classes (doctor, hospital, admin, patient, …) in DTO package:

```
public class Patient {

    private Integer ID;
    private String fname;
    private String lname;
    private String sex;
    private String email;
    private String pass;
    private String address;
    private String lang;
    private String hashcode;
    private String status;
    private Integer attempt;

    public Patient() {
    }

    public Patient(Integer ID, String fname, String lname, String sex, String email, String pass, String address, String lang) {
        this.ID = ID;
        this.fname = fname;
        this.lname = lname;
        this.sex = sex;
        this.email = email;
        this.pass = pass;
        this.address = address;
        this.lang = lang;
    }

    public Integer getID() {
        return ID;
    }

    public void setID(Integer ID) {
        this.ID = ID;
    }

    public String getFname() {
        return fname;
    }

    public void setFname(String fname) {
        this.fname = fname;
    }
```

```
public class Doctor {

    private int ID;
    private String fname;
    private String lname;
    private String sex;
    private String degree;
    private boolean insurance;
    private String speciality;
    private String hours;
    private String lang;
    private int allowReview;

    public Doctor() {
    }

    public Doctor(String fname, String lname, String sex, String degree, boolean insurance, String speciality, String hours, String lang) {
        this.fname = fname;
        this.lname = lname;
        this.sex = sex;
        this.degree = degree;
        this.insurance = insurance;
        this.speciality = speciality;
        this.hours = hours;
        this.lang = lang;
    }
```

_Control structure:

+ Selection statements: There is an example from Login class in Controller package:

```
if (action == null) {
    rd = sc.getRequestDispatcher("/login.jsp");
    rd.forward(request, response);
} else {
    Message msg = new Message();
    adminLogin = false;

    String email = request.getParameter("email");
    String pass = request.getParameter("password");
    String remember = request.getParameter("remember");

    if (email == null || pass == null) {
        msg.setCode(0);
        msg.setText("Please type your email and password");
    } else {
        Patient patient = PatientDAO.getUserbyEmail(email);

        String error = "";
        if (patient.getPass() == null) {

            admin = AdminDAO.getUserbyEmail(email);

            // Check if user does not exist
            if (admin.getPass() == null) {
                error = "User account does not exist";
            } else if (!BCrypt.checkpw(pass, admin.getPass())) {
                error = "Your password is not correct";
            } else {
                adminLogin = true;
            }

        } else if (!BCrypt.checkpw(pass, patient.getPass())) {
            error = "Your password is not correct";
```

+ Iteration Statements: are used almost in all classes in DAO package to get or insert data from MySQL and some JSP page to show the list of objects:

```
public static Admin getUserbyEmail(String email) {
    Admin admin = new Admin();

    String query = "SELECT * FROM admin WHERE email = ?";

    // Connect to database
    Connection connection = Database.getConnection();

    try {
        PreparedStatement ps = connection.prepareCall(query);
        ps.setString(1, email);
        ResultSet rs = ps.executeQuery();

        while (rs.next()) {
            admin.setID(rs.getInt("id"));
            admin.setEmail(rs.getString("email"));
            admin.setPass(rs.getString("password"));
        }

        connection.close();
        return admin;

    } catch (SQLException ex) {
        ex.printStackTrace();
    }
    return null;
}
```

```
<%
    Patient patient = null;
    Admin admin = null;

    Cookie[] cookies = request.getCookies();
    if (cookies != null) {
        for (Cookie cookie : cookies) {
            if (cookie.getName().equals("u_email")) {
                patient = PatientDAO.getUserbyEmail(cookie.getValue());
            } else if (cookie.getName().equals("a_email")) {
                admin = AdminDAO.getUserbyEmail(cookie.getValue());
            }
        }
    }
    if (session.getAttribute("patient") != null) {
        patient = (Patient) session.getAttribute("patient");
    } else if (session.getAttribute("admin") != null) {
        admin = (Admin) session.getAttribute("admin");
    }
%>
```

+ Jump Statements are used almost in all classes in Controller package. The following figure is an example from activeReview class in Controller package:

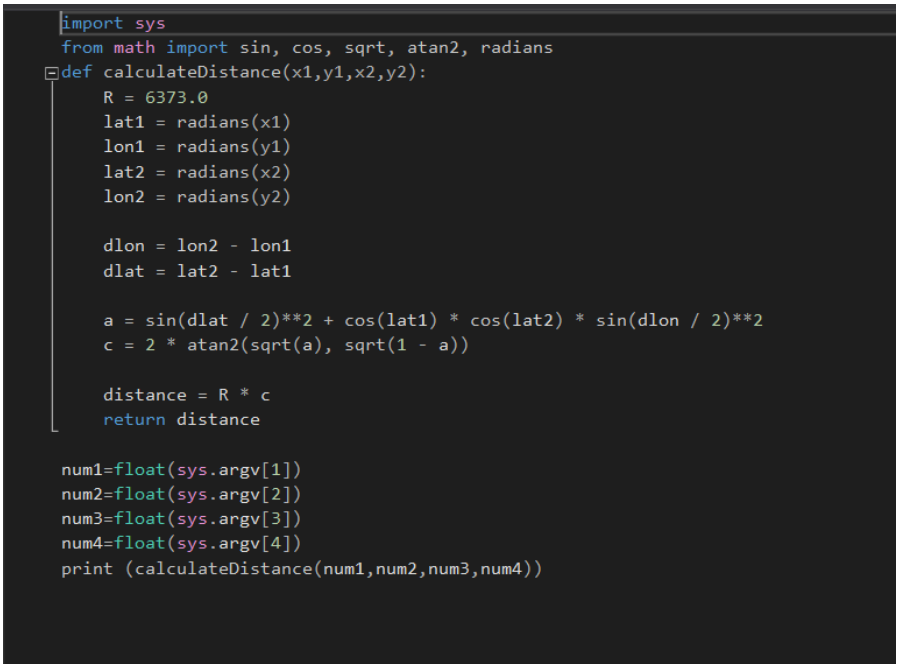
```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    AdminDAO admin = new AdminDAO();
    String action = request.getParameter("action");

    switch (action) {
        case "ActivateReview": {
            String idd = request.getParameter("activateDoctorID");
            int id = Integer.parseInt(idd);
            admin.activateReview(id);
            response.sendRedirect("viewdoctor.jsp");
            break;
        }
        case "DeactivateReview": {
            String idd = request.getParameter("deActivateDoctorID");
            int id = Integer.parseInt(idd);
            admin.deactivateReview(id);
            response.sendRedirect("viewdoctor.jsp");
            break;
        }
        case "removeDoctor": {
            String idd = request.getParameter("removeDoctorID");
            int id = Integer.parseInt(idd);
            admin.removeDoctor(id);
            response.sendRedirect("adminuser.jsp");
            break;
        }

        default:
            break;
    }
}
```

_Exception handling: are used in all DAO class to handle runtime errors of SQL while connecting database:

```
public List<Comment> getAllComment(int id) {
    List<Comment> list = new ArrayList<>();
    String query = "SELECT * FROM comment WHERE d_id = ?";

    // Connect to database
    Connection connection = Database.getConnection();

    try {
        PreparedStatement ps = connection.prepareCall(query);
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();

        while (rs.next()) {
            Comment c = new Comment();
            c.setID(rs.getInt("c_id"));
            c.setComment(rs.getString("c_comment"));
            c.setdID(rs.getInt("d_id"));
            c.setpID(rs.getInt("p_id"));
            list.add(c);
        }
        connection.close();
    } catch (SQLException ex) {
        Logger.getLogger(CommentDAO.class.getName()).log(Level.SEVERE, null, ex);
    }

    return list;
}
```

- **Python**

_Data types: are used in caldistance.py in folder calculate



```python
import sys
from math import sin, cos, sqrt, atan2, radians
def calculateDistance(x1,y1,x2,y2):
    R = 6373.0
    lat1 = radians(x1)
    lon1 = radians(y1)
    lat2 = radians(x2)
    lon2 = radians(y2)

    dlon = lon2 - lon1
    dlat = lat2 - lat1

    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))

    distance = R * c
    return distance

num1=float(sys.argv[1])
num2=float(sys.argv[2])
num3=float(sys.argv[3])
num4=float(sys.argv[4])
print (calculateDistance(num1,num2,num3,num4))
```

6. Language decisions:

_ Python:
+ When code can be shrinked down to couple of scripts. By scripts, they should be merely like bash scripts, it can be full fledged program with classes.
_ Java:
+ If we want to use whole bunch of third party libraries - Java it is! Maven is superior. Pip and setup tools are close though, still sticking with maven build and package distribution mechanism
+ If we have to create hundreds of source files, then Java comes naturally because of its jar packaging to zip them all.