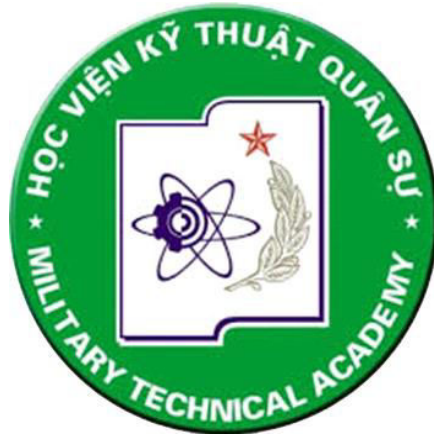


HỌC VIỆN KỸ THUẬT QUÂN SỰ

KHOA CÔNG NGHỆ THÔNG TIN



Báo Cáo Lập Trình Nâng Cao

Đề tài: Xây dựng Web quản lý đặt phòng khách sạn

Nhóm: 12 - Lớp Sáng Thứ 6 - H5512

Thành viên:

Nguyễn Ngọc Khánh - CNTT15

Lý Văn Chấn - CNTT15

Bùi Đình Thủy - CNTT15

1. Phân công công việc

Thành viên	Chức năng
Nguyễn Ngọc Khánh	Xây dựng trang quản trị Admin, thêm sửa xóa Thành phố, Khách sạn, Tài khoản...
Lý Văn Chấn	Xây dựng các trang liên quan đến khách sạn như Trang chủ, Danh sách khách sạn, Chi tiết khách sạn
Bùi Đình Thủy	Xây dựng các trang về người dùng như Đăng ký, Đăng nhập, Trang cá nhân

2. Các công nghệ sử dụng: JSF - Primefaces - JDBC

JSF là gì?

- JavaServer Faces (JSF) là một web framework MVC. Nó tập trung vào việc đơn giản hóa xây dựng giao diện người dùng (User Interface -UI) (với hơn 100 tags UI) cho ứng dụng web và làm cho phép việc sử dụng lại các thành phần UI được thực hiện một cách dễ dàng.

Lợi ích của JSF

- JSF giảm chi phí trong việc tạo và bảo trì ứng dụng, nó sẽ chạy trên một máy chủ ứng dụng Java và hiển thị các thành phần UI cho khách hàng. JSF tạo điều kiện cho phát triển ứng dụng web nhờ các tính năng sau:
- Cung cấp các thành phần giao diện (UI) sử dụng lại được Chuyển dữ liệu dễ dàng giữa các thành phần UI Quản lý trạng thái UI giữa các request từ server

- Cho phép cài đặt các thành phần UI tùy chỉnh Lấy sự kiện ở phía client tới phía server

Các tính năng của JSF

- Các thành phần dựa trên framework (Component Based Framework): JSF là 1 framework server-side dựa trên các component. Nó cung cấp các component sẵn có để xây dựng ứng dụng web. Bạn có thể sử dụng HTML5, các Facelets tag để tạo trang web.
- Công nghệ Facelets (Facelets Technology): Facelets là một mã nguồn mở về hệ thống web mẫu (Web template system). Nó là công nghệ xử lý view mặc định cho JSF, yêu cầu đầu vào XML hợp lệ để làm việc. Facelets hỗ trợ tất cả các thành phần UI của JSF và tập trung hoàn toàn vào xây dựng view cho ứng dụng JSF
- Expression Language (EL) Expression language cung cấp một cơ chế quan trọng cho việc tạo giao diện người dùng (web pages) để giao tiếp với ứng dụng (các beans).
- Phát triển web dễ dàng và nhanh chóng (Ease and Rapid web Development): JSF cung cấp các tool và các thư viện sẵn có và nhiều tiện ích để bạn có thể phát triển ứng dụng web dễ dàng và nhanh chóng.
- Hỗ trợ đa ngôn ngữ (Support Internationalization): JSF hỗ trợ đa ngôn ngữ cho việc tạo ứng dụng web. Bạn có thể tạo một ứng dụng cho nhiều ngôn ngữ khác nhau.
- Bean Annotations: JSF cung cấp các annotations tiện ích, thứ giúp bạn có thể thực hiện kiểm tra tính hợp lệ của data.
- Xử lý exception (Exception Handling): JSF cung cấp cách xử lý exception mặc định.

- Templating: trong ứng dụng JSF bạn có thể tạo các template mới, sử dụng lại chúng và coi nó như một component.
- Hỗ trợ AJAX (AJAX Support): Bạn có thể hiển thị các request tới server side mà không cần tải lại trang. JSF cũng hỗ trợ việc hiển thị lại từng thành phần sử dụng AJAX. (Ví dụ: bạn gõ vào ô search, nó sẽ hiển thị ra kết quả mà không cần tải lại trang... JSF khá mạnh trong làm việc với AJAX, gần như bạn không cần phải quan tâm tới việc gửi nhận request, javascript trong JSF)
- Bảo mật (Security): JSF lưu trữ trạng thái của các thành phần view, khi gửi request tới server, các thành phần này phải có các tham số hợp lệ (ví dụ field name của thẻ input với có giá trị giống với giá trị server đang lưu). Các tham số này được tạo ra một cách ngẫu nhiên.

Kiến trúc của JSF

- Faces Servlet đóng vai trò là Controller: nhận các request từ client gửi đến và điều hướng tới View hoặc Model
- Các Managed Beans đóng vai trò là Model: thực hiện các chức năng business và giao tiếp với data
- Phần View gồm JSP Pages và các component JSF.

PrimeFaces là gì?

- Primefaces là một framework mã nguồn mở cho JSF (JavaServer Faces) với hơn 100 component, tối ưu cho mobile, validate phía client, hỗ trợ AJAX mạnh mẽ...

- PrimeFaces cũng được coi là một trong những thư viện UI phổ biến nhất trong hệ sinh thái Java, được sử dụng rộng rãi bởi nhiều công ty, tập đoàn lớn trên thế giới như: Cisco, Ebay, HP, Ford...
- Ngoài PrimeFaces Showcase thì còn có PrimeFacesNG base trên Angular JS, PrimeFacesReact base trên ReactJS...

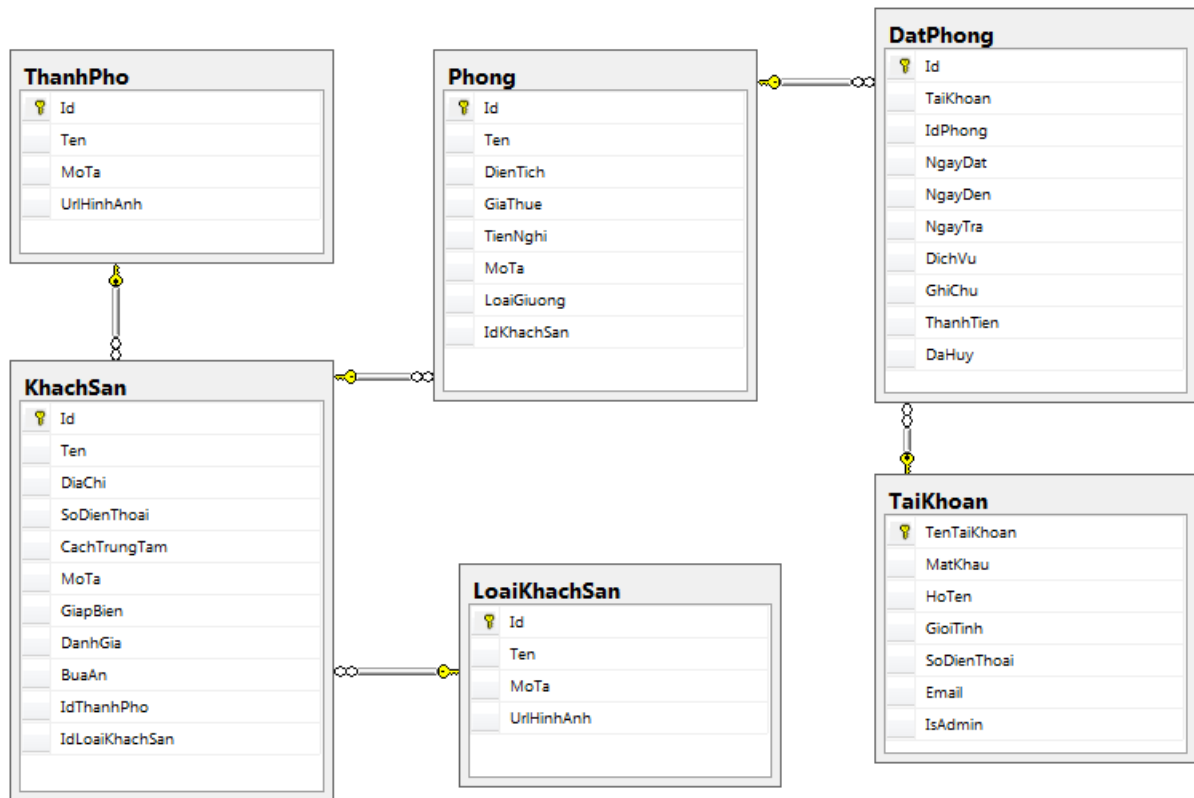
Tại sao dùng PrimeFaces?

- JSF xử lý AJAX rất mạnh, nhưng với PrimeFaces khả năng xử lý ấy còn mạnh mẽ hơn ở mức từng component, tùy chỉnh linh hoạt...
- Đơn giản và hiệu suất: PrimeFaces rất nhẹ, cấu hình đơn giản
- Dễ sử dụng
- Cộng đồng lớn
- Kết hợp với Spring: Được Spring khuyến dùng trong các ứng dụng Spring – JSF

JDBC là gì?

- Java JDBC là một java API được sử dụng để kết nối và thực hiện truy vấn với cơ sở dữ liệu. JDBC API sử dụng trình điều khiển jdbc để kết nối với cơ sở dữ liệu.

3. Cơ sở dữ liệu của web



Các câu lệnh tạo bảng được đính kèm trong QuanLyKhachSanDb.sql

3. Các Thành Phần Chính

Trong Project:

- Package bean: Lưu các Bean:
 - BeanKhachSan, BeanLoaiKhachSan, BeanThanhPho, BeanPhong, BeanTaiKhoan để lưu các thông tin hiển thị của Khách Sạn, Loại Khách Sạn... Trên view và trang Admin

- BeanNguoiDung để lưu các thông tin đăng nhập của người dùng và các hàm chuyển hướng liên quan đến người dùng như Đăng Ký, Đăng Nhập, xem Lịch Sử, Cá Nhân...
- BeanNavigation chủ yếu để lưu các thông tin chuyển hướng trang như Tìm Kiếm, sang trang danh sách khách sạn, sang trang thông tin khách sạn, đặt phòng, lọc khách sạn
- Package dao: Các class hỗ trợ kết nối CSDL bằng JDBC
- Package filter: Class MyFilter để kiểm tra Cookie và kiểm tra bảo mật
- Package model: Các Model chứa thông tin của dữ liệu
- Package msg: Class Message chứa các hàm phục vụ hiển thị message
- Package util: Các Class tiện ích như so sánh ngày, chuyển chữ có dấu về không dấu phục vụ tìm kiếm

ApplicationScoped: Tất cả các bean này sẽ được khởi tạo khi ứng dụng web chạy hoặc khi có request đầu tiên được gửi đến, nó đi theo server, và chỉ bị mất đi khi ứng dụng web bị shutdown

SessionScoped: Đi theo session, mỗi khi một trình duyệt nào đó gửi request lần đầu thì tất cả các bean này được khởi tạo đi theo phiên làm việc đó, mất đi khi trình duyệt đó tắt hoặc hết thời gian, nghĩa là mỗi phiên làm việc có một bean riêng

Các bean như BeanKhachSan, BeanLoaiKhachSan, BeanThanhPho chỉ để hiển thị cho người dùng, ai cũng thấy như nhau nên để ApplicationScoped, mục đích chính để tiết kiệm

BeanNavigation, BeanNguoiDung lưu các thông tin người dùng, mỗi người một khác nên để SessionScoped

Mỗi thuộc tính trong Bean cần có hàm get, set tương ứng để có thể truy cập từ các View, trong view có thể sử dụng `#{tên bean. tên thành phần}` để truy cập đến các thành phần trong bean

VD:

Trong view `dsKhachSan`, để hiển thị toàn bộ khách sạn thì cần một `BeanNavigation` lưu lại danh sách các khách sạn thỏa mãn từ khóa tìm kiếm, sau đó trong view chỉ cần dùng một `p:dataList` để show các khách sạn đó ra:

```
<p:dataList value="#{beanNavigation.listKhachSan}" var="item">
    <p:outputLabel value="#{item.ten}"/>
</p:dataList>
```

Để có thể lưu lại thông tin một tài khoản sau khi chọn cập nhật trong view `caNhan`, cần sử dụng các `p:inputText` kết hợp với `p:commandButton` lồng vào trong một `h:form`

```
<h:form>
    <p:inputText value="#{beanNguoiDung.taiKhoanDangNhap.hoTen}" />
    <p:inputText value="#{beanNguoiDung.taiKhoanDangNhap.matKhau}" />
    <p:commandButton action="#{beanNguoiDung.capNhatThongTin()}" />
</h:form>
```

Như vậy, sau khi bấm vào Button, biến `taiKhoanDangNhap` sẽ được lưu lại các thông tin trong `inputText` đồng thời hàm `capNhatThongTin()` trong `beanNguoiDung` sẽ được gọi

Minh họa sử dụng JSF - BeanTaiKhoan

```
package bean;

import java.io.Serializable;
import model.*;
import java.util.ArrayList;
import javax.faces.bean.ApplicationScoped;
import javax.faces.bean.ManagedBean;
import org.primefaces.PrimeFaces;

@ManagedBean(name = "beanTaiKhoan", eager = true)
@ApplicationScoped
public class BeanTaiKhoan implements Serializable {

    private static final long serialVersionUID = 45438764L;

    private TaiKhoan taiKhoan;
    private ArrayList<TaiKhoan> listTaiKhoan;

    public BeanTaiKhoan() {
        taiKhoan = new TaiKhoan();
        listTaiKhoan = dao.DAOTaiKhoan.getAll();
    }

    public void reset() {
        taiKhoan = new TaiKhoan();
    }

    public void insert(TaiKhoan tmp) {
        if (tmp.getTenTaiKhoan().isEmpty() || tmp.getMatKhau().isEmpty()) {
            msg.Message.errorMessage("Thất Bại", "Không được để trống Tên tài
```

Minh họa sử dụng Primefaces - Trong view dsKhachSan, có sử dụng các p:selectBooleanCheckbox kèm theo ajax cập nhật lại form:ds-khachsan để làm bộ lọc

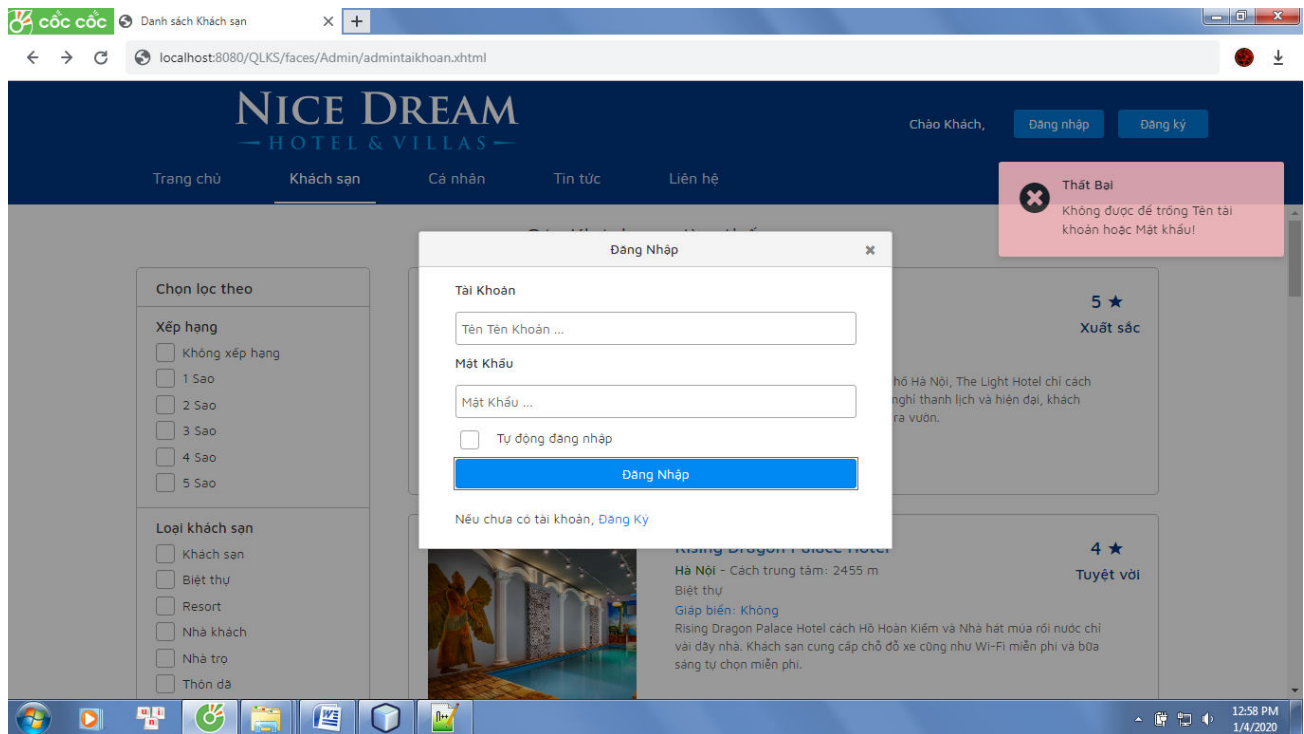
```
<div style="padding: 10px 0 8px 20px">
    <span style="font-size: 16px; font-weight: bold">Chọn lọc theo</span>
</div>
<div class="khoi-loc">
    <div class="head">Xếp hạng</div>
    <c:forEach items="#{beanNavigation.listXepHang}" var="item">
        <div>
            <p:selectBooleanCheckbox value="#{item.checked}" itemLabel="#{it
                <p:ajax update="form:ds-khachsan" listener="#{beanNavigation
            </p:selectBooleanCheckbox>
        </div>
    </c:forEach>
</div>
<div class="khoi-loc">
    <div class="head">Loại khách sạn</div>
    <c:forEach items="#{beanNavigation.listLoaiKhachSan}" var="item">
        <div>
            <p:selectBooleanCheckbox value="#{item.checked}" itemLabel="#{it
                <p:ajax update="form:ds-khachsan" listener="#{beanNavigation
            </p:selectBooleanCheckbox>
        </div>
    </c:forEach>
</div>
```

Minh họa sử dụng JDBC - Các class trong package dao sử dụng JDBC để kết nối CSDL

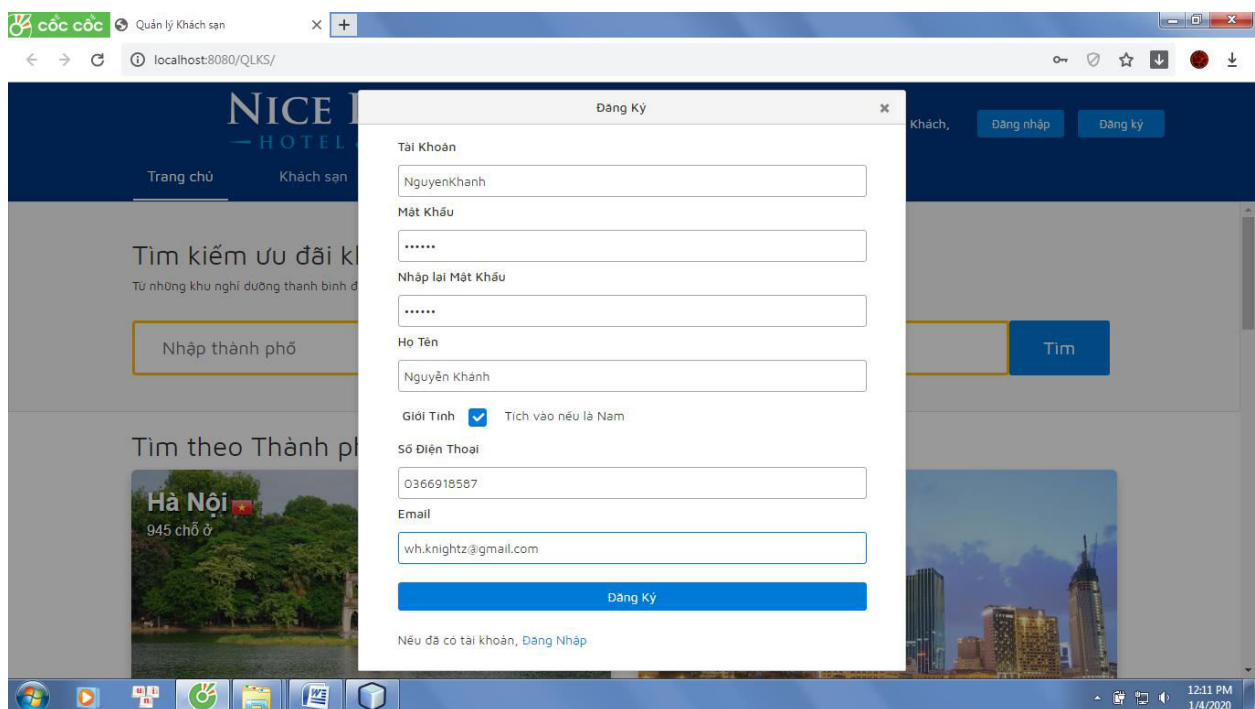
```
public class DAOTaiKhoan {  
  
    private static Connection con;  
  
    public static ArrayList<TaiKhoan> getAll() {  
        ArrayList<TaiKhoan> list = new ArrayList();  
        try {  
            con = SqlConnection.getConnection();  
            Statement stmt = con.createStatement();  
            ResultSet rs = stmt.executeQuery("select * from TaiKhoan");  
            while (rs.next()) {  
                TaiKhoan tmp = new TaiKhoan();  
                tmp.setTenTaiKhoan(rs.getString(1));  
                tmp.setMatKhau(rs.getString(2));  
                tmp.setHoTen(rs.getString(3));  
                tmp.setGioiTinh(rs.getBoolean(4));  
                tmp.setSoDienThoai(rs.getString(5));  
                tmp.setEmail(rs.getString(6));  
                tmp.setIsAdmin(rs.getBoolean(7));  
                list.add(tmp);  
            }  
            con.close();  
        } catch (Exception e) {  
        }  
        return list;  
    }  
}
```

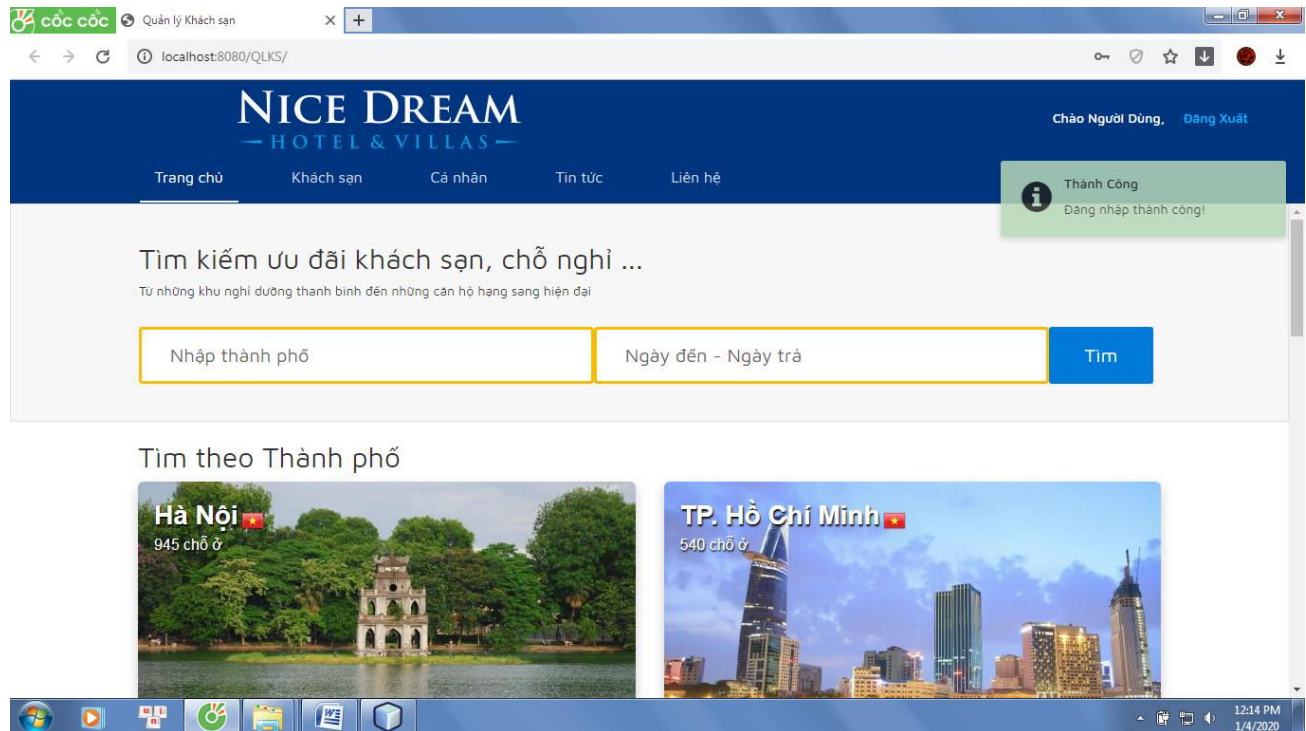
5. Các Chức Năng

- Tại tất cả các View đều có thể bấm vào nút Đăng Nhập/ Đăng Ký (Primefaces - p:commandButton) ở góc phải của web khi chưa đăng nhập, nếu đăng nhập xong thì chuyển thành nút Đăng xuất (Primefaces - p:commandLink)

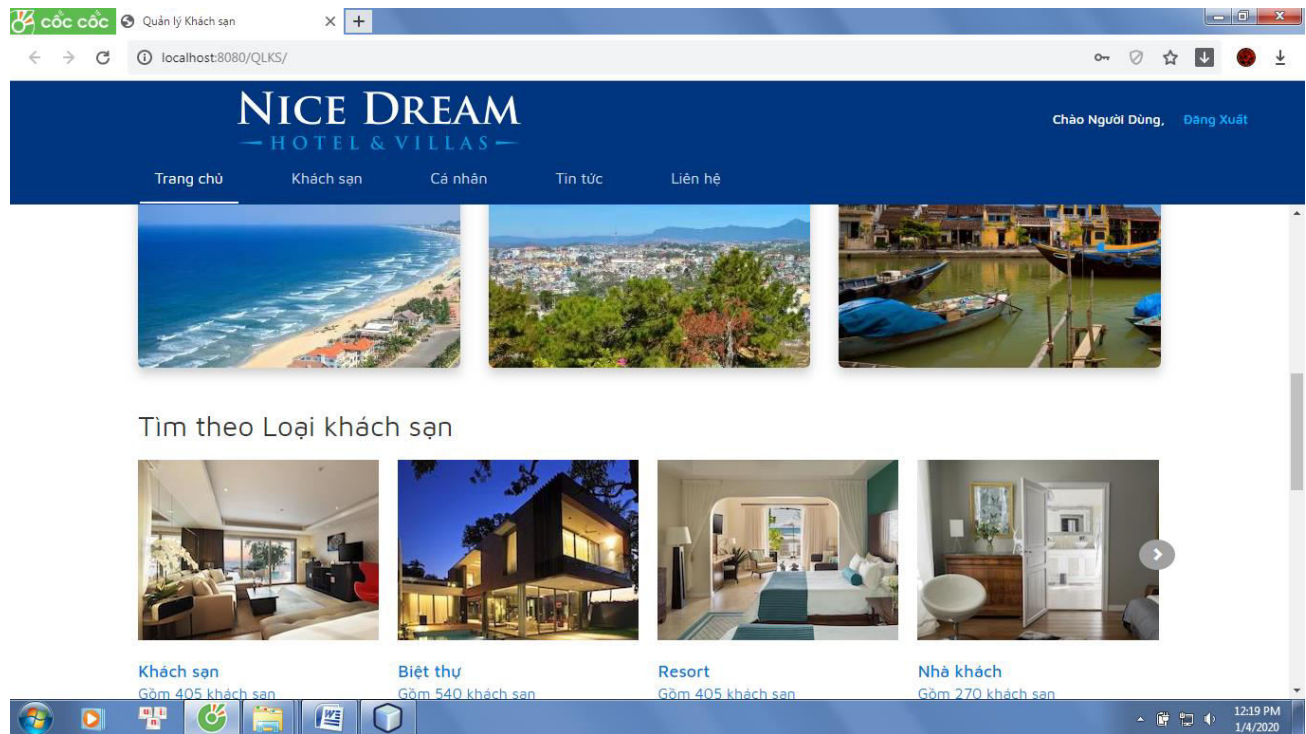


- Khi Chọn Đăng Nhập/Đăng Ký sẽ có một Dialog hiện lên (Primefaces - p:dialog), yêu cầu nhập đủ thông tin, nếu nhập thiếu hoặc sai thông tin sẽ có một Message thông báo ở góc phải (Primefaces - p:growl)

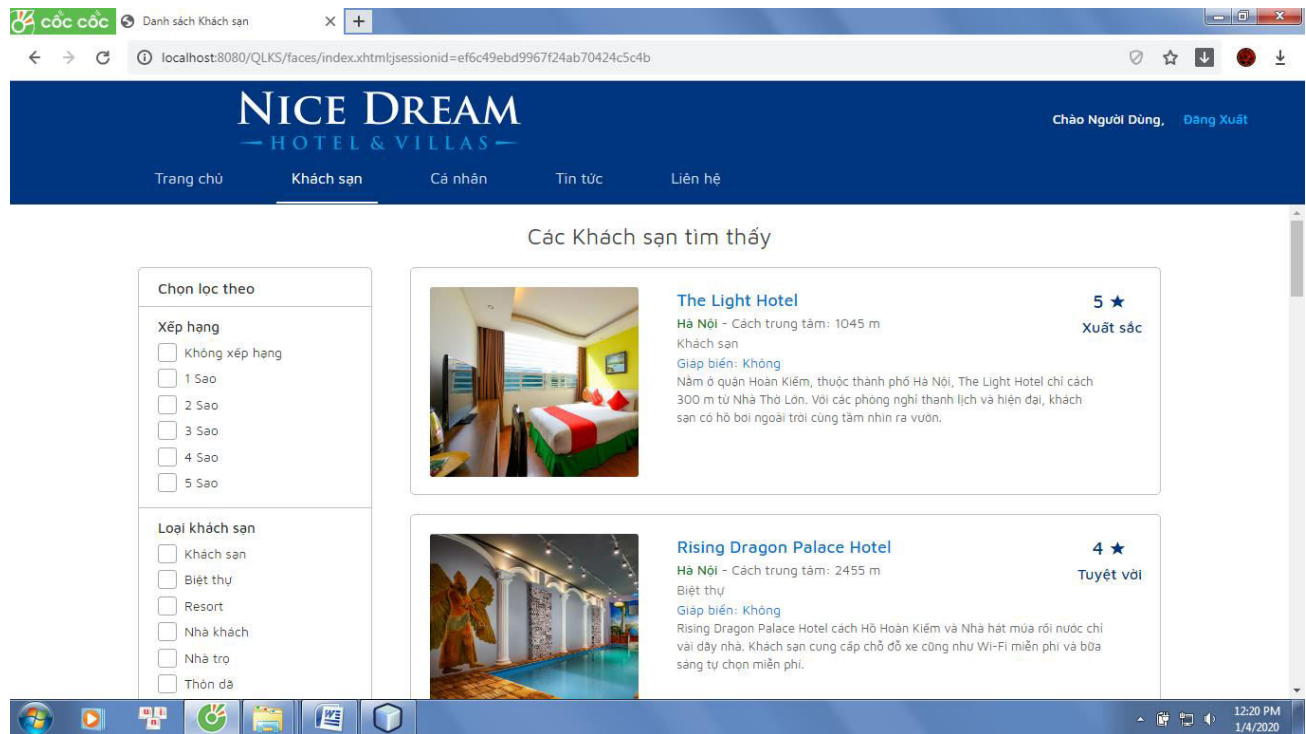




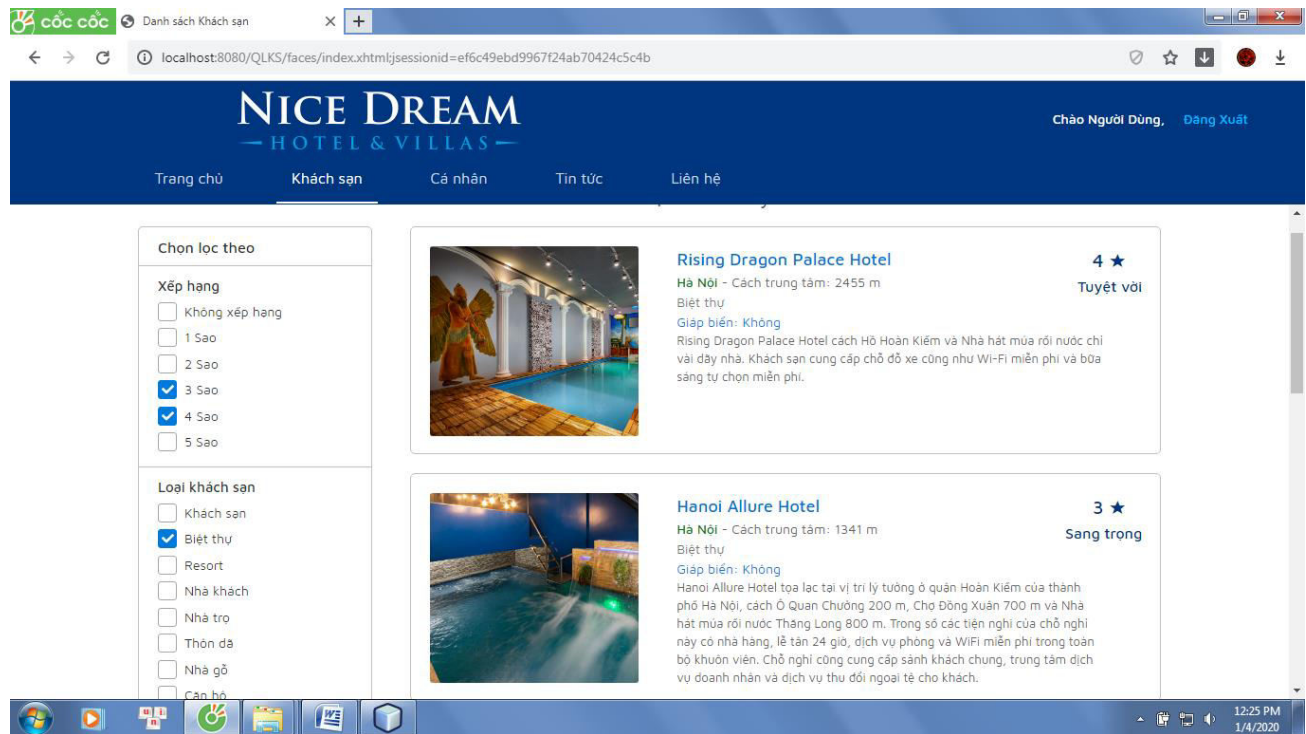
- Đăng nhập thành công nếu là Tài khoản Người dùng sẽ có thông báo đăng nhập thành công, không cần load lại trang (Primefaces - p:commandButton có ajax), Nếu là Admin sẽ chuyển hướng sang trang quản trị
- Tại trang chủ, Người dùng có các lựa chọn:
 - Tìm kiếm theo tên thành phố: Nhập tên thành phố, chọn ngày đến, ngày trả phòng, và bấm Tìm, các khách sạn thỏa mãn sẽ được hiển thị



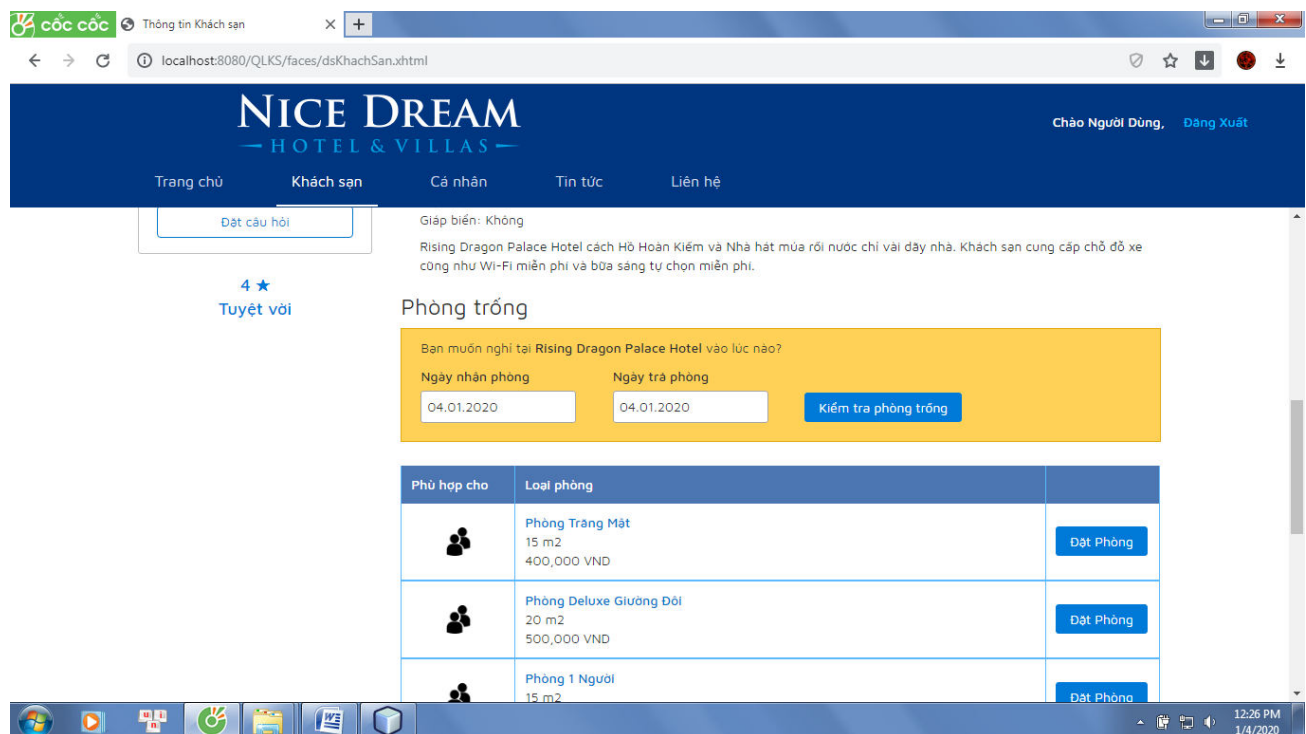
- Ngoài ra người dùng còn có thể có các lựa chọn khác như hiển thị toàn bộ khách sạn thuộc một thành phố nào đó hay nhiều loại chỗ ở như Biệt thự, Resort, Nhà khách ...



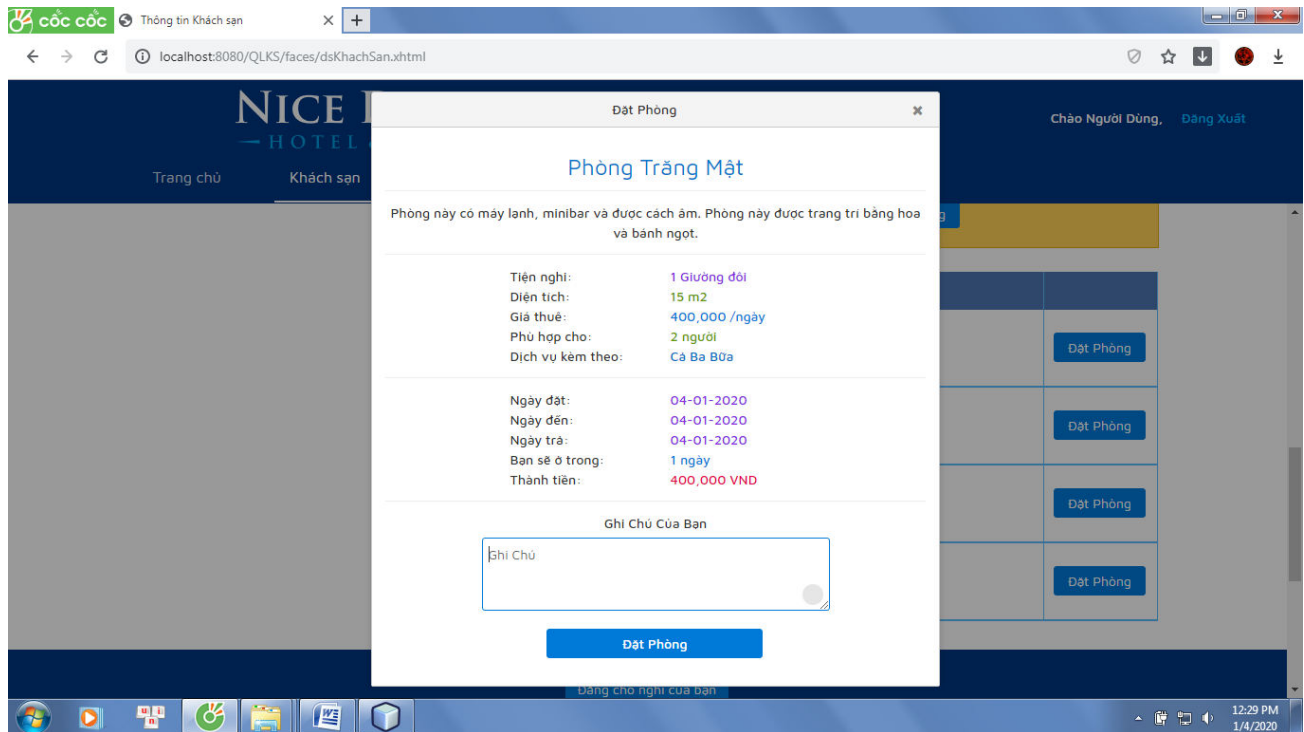
- Sau khi tìm kiếm xong, danh sách các khách sạn thỏa mãn sẽ hiện ra (sử dụng Primefaces - `p:dataList` để hiển thị danh sách), tại đây có thể sử dụng bộ lọc bên trái để lọc ra các khách sạn mong muốn theo các tiêu chí như Xếp hạng, Loại khách sạn, Bữa ăn mà khách sạn hỗ trợ, Cách trung tâm thành phố bao nhiêu, Có giáp biển không (Sử dụng Primefaces - `p:selectBooleanCheckbox` kết hợp ajax để lọc, không cần load lại trang)



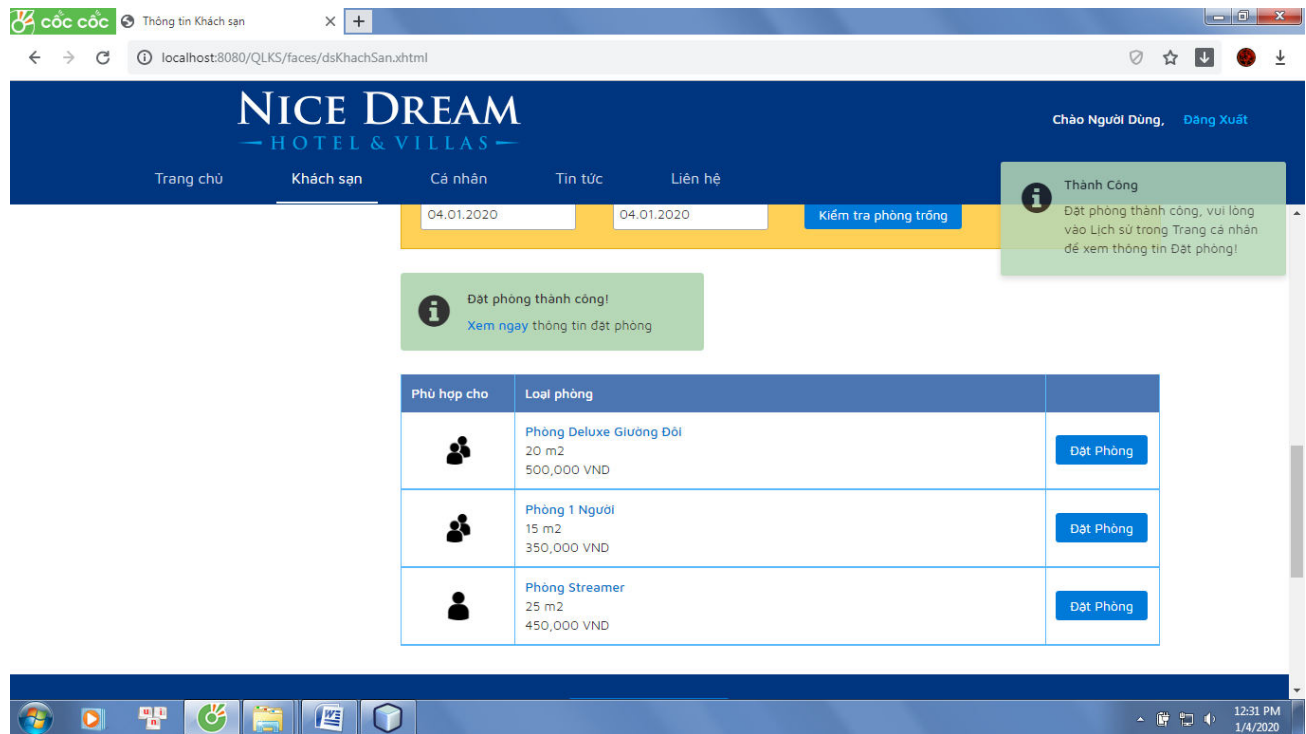
- Sau khi chọn xong khách sạn mong muốn, thông tin khách sạn đó sẽ hiện ra chi tiết bao gồm cả các loại phòng của khách sạn đó



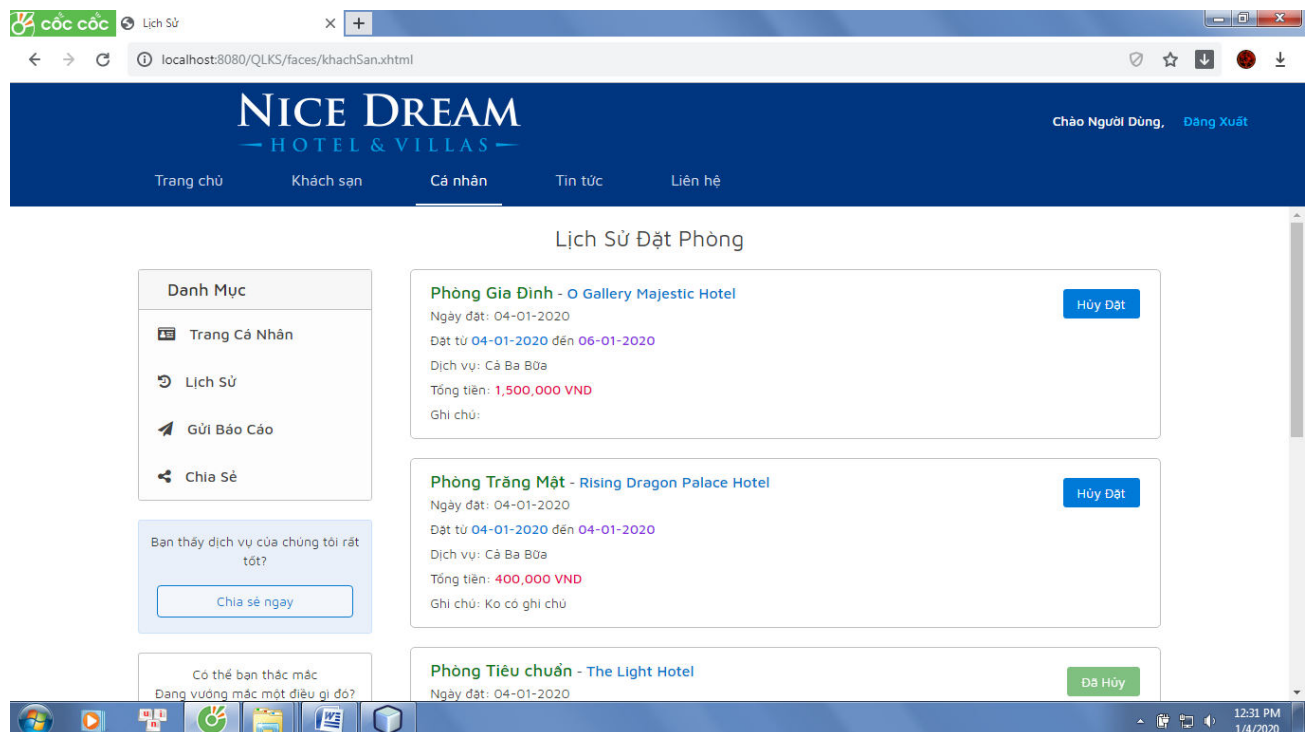
- Tại đây, khách hàng muốn đặt khách sạn trong khoảng thời gian nào thì cần nhập thông tin vào các ô Ngày nhận phòng và Ngày trả phòng (p:datePicker) rồi bấm Kiểm tra phòng trống, các Phòng trống trong khoảng thời gian đó sẽ hiện ra, khách hàng hài lòng với phòng nào thì bấm chọn đặt phòng đó



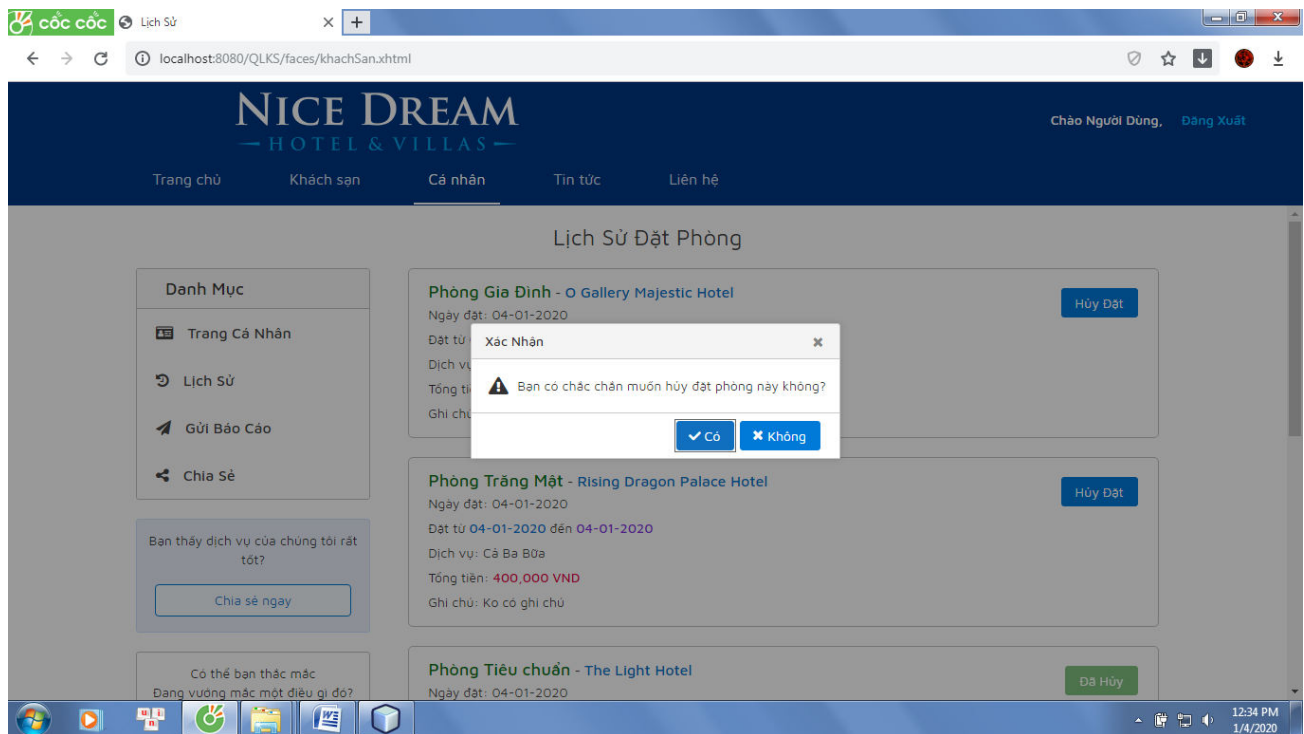
- Nếu khách hàng chưa đăng nhập thì có thể xem thông tin phòng nhưng không thể đặt phòng mà cần đăng nhập trước, sau khi đặt phòng, phòng được đặt đã được sử dụng trong khoảng thời gian đặt phòng nên đương nhiên sẽ biến mất, có thể chuyển sang Lịch sử đặt phòng để xem thông tin các phòng đã đặt bao gồm cả các phòng đã hủy đặt



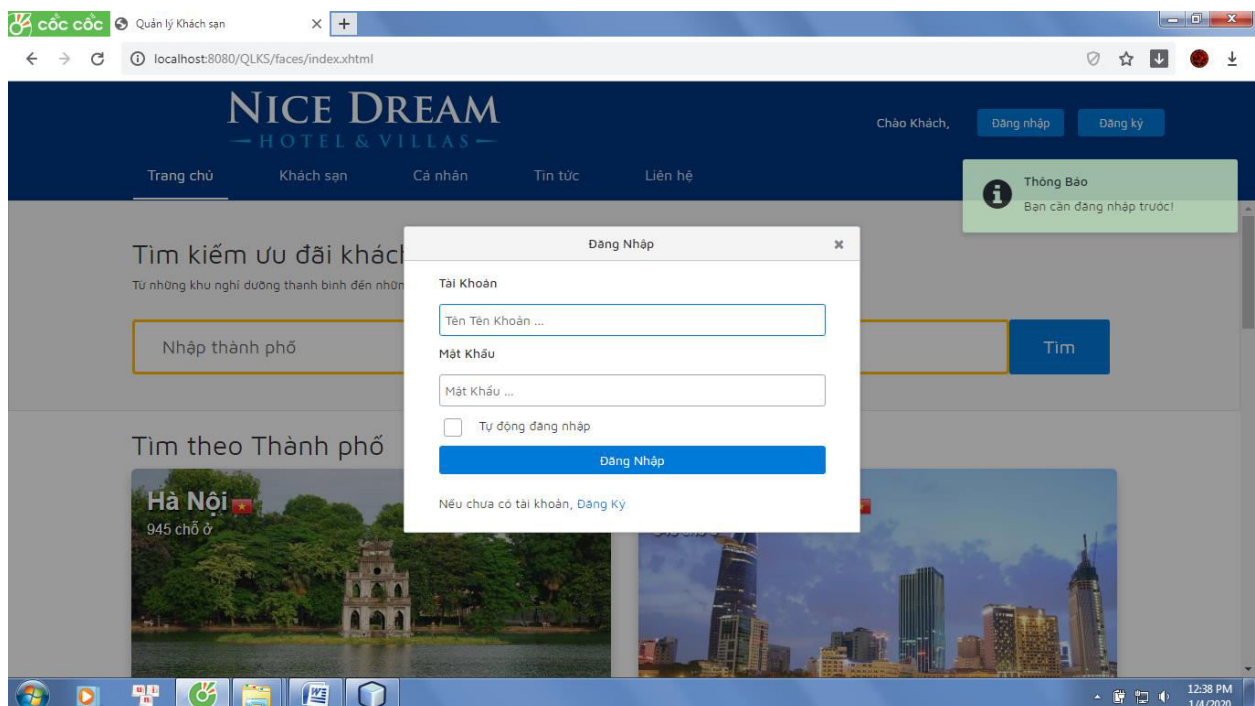
- Trong Lịch sử, các thông tin đặt phòng chi tiết từ trước đến nay sẽ được hiển thị, tại đây, nếu phòng đó chưa quá hạn đặt thì có thể hủy (sử dụng Primefaces - p:dataList kết hợp với p:commandButton để hiển thị)



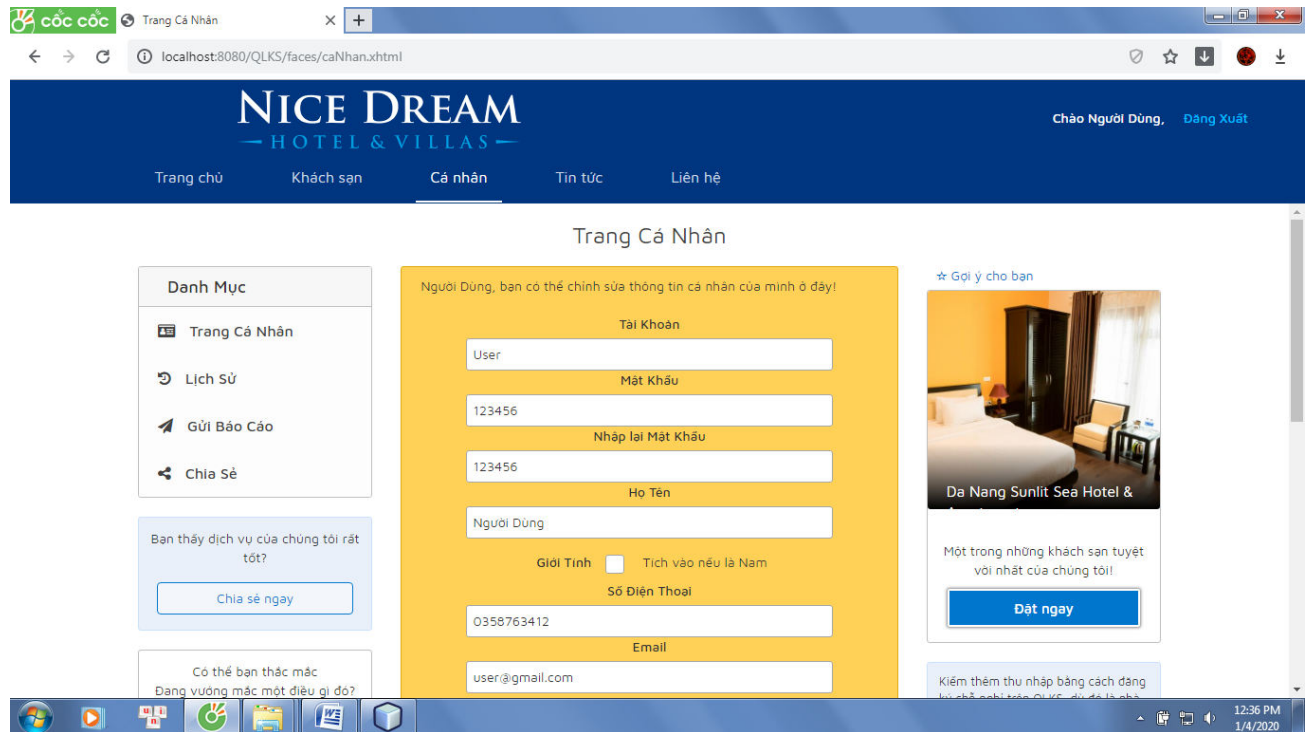
- Khi hủy đặt phòng sẽ có yêu cầu xác nhận trước (p:confirmDialog), chỉ khi bấm có thì phòng mới bị hủy đặt



- Ngoài ra, nếu khách hàng muốn cập nhật thông tin cá nhân của mình thì có thể vào mục Cá Nhân, tuy nhiên nếu chưa đăng nhập mà bấm vào Cá Nhân thì trang web sẽ yêu cầu đăng nhập trước



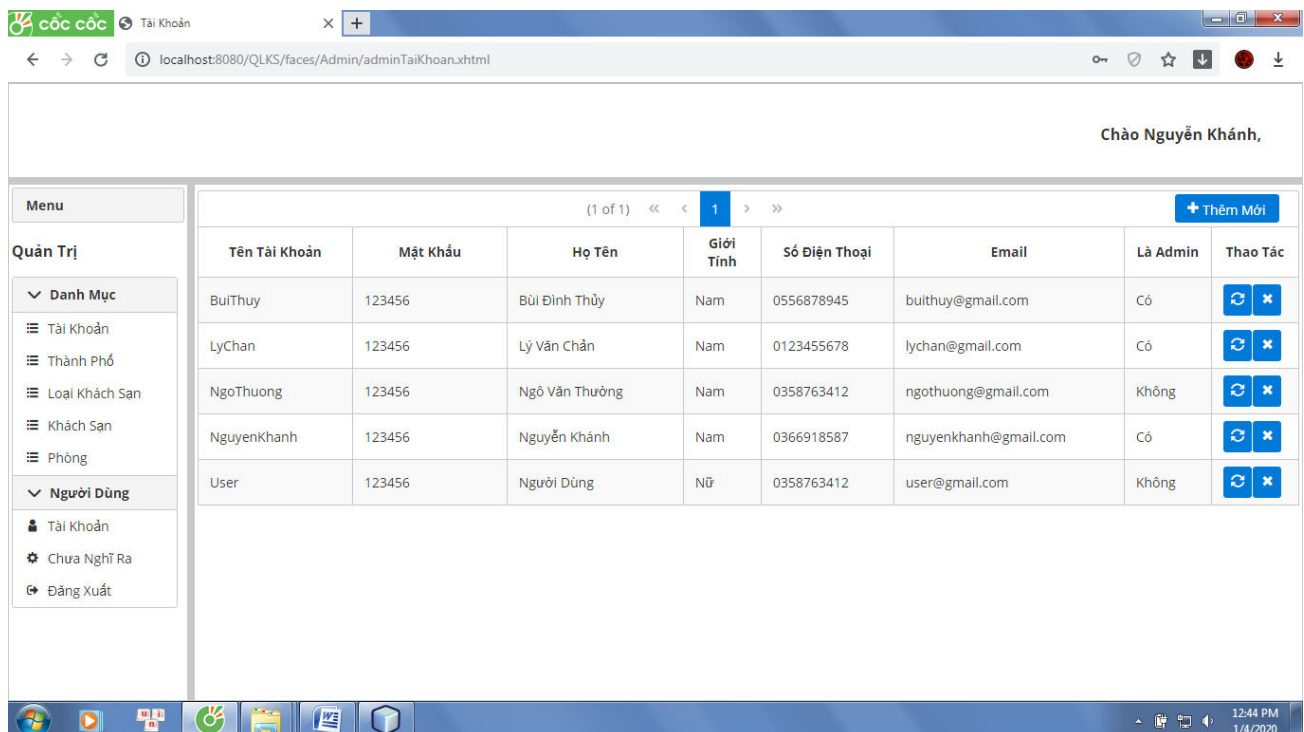
- Khi vào trang cá nhân, có thể cập nhật các thông tin của bản thân bằng cách nhập các thông tin vào các textbox (p:inputText) rồi bấm Cập nhật (p:commandButton có ajax), sau khi cập nhật, session chứa thông tin tài khoản cũng ngay lập tức được cập nhật



- Trên Navigation bar có sử dụng các h:commandLink để chuyển hướng trang
- Ngoài ra còn sử dụng Cookies để lưu thông tin tài khoản trên trình duyệt, sau khi đăng nhập lần sau vào lại hệ thống tự động đăng nhập.

Quản Trị Admin

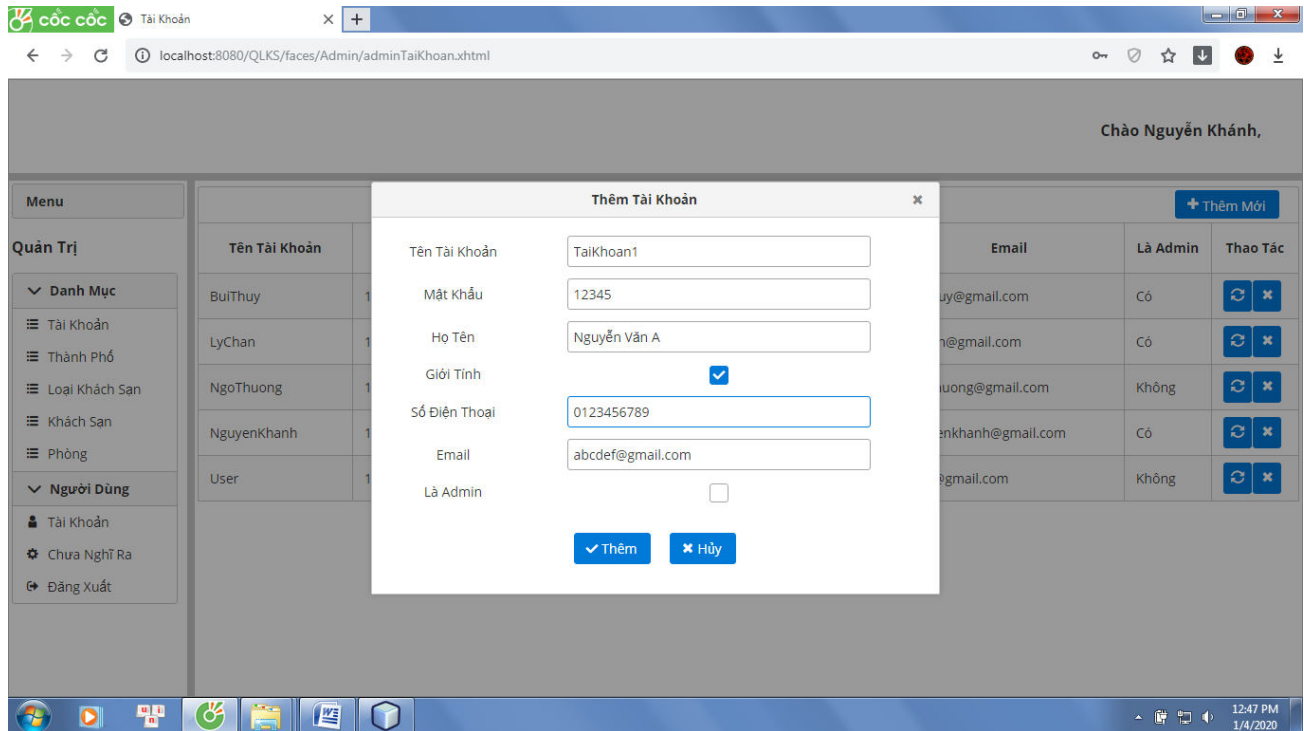
- Ban đầu sẽ có sẵn 3 tài khoản với quyền admin:
 - NguyenKhanh - 123456
 - LyChan - 123456
 - BuiThuy - 123456
- Khi đăng nhập bằng những tài khoản là admin, hệ thống sẽ tự chuyển sang trang Quản trị, Hiển thị Danh sách các Tài Khoản, Thành Phố, Loại Khách Sạn, Khách Sạn và các Phòng. (menu trái sử dụng p:menu, sử dụng p:dataTable để hiển thị dạng bảng)
- Tại đây admin có thể thực hiện 3 chức năng Thêm, Sửa, Xóa đối với Tài Khoản, Thành Phố, Loại Khách Sạn, Khách Sạn và các Phòng
- Giao diện chính trang Quản trị, sử dụng, các bảng sử dụng p:dataTable, có phân trang, khi next trang không cần load lại view



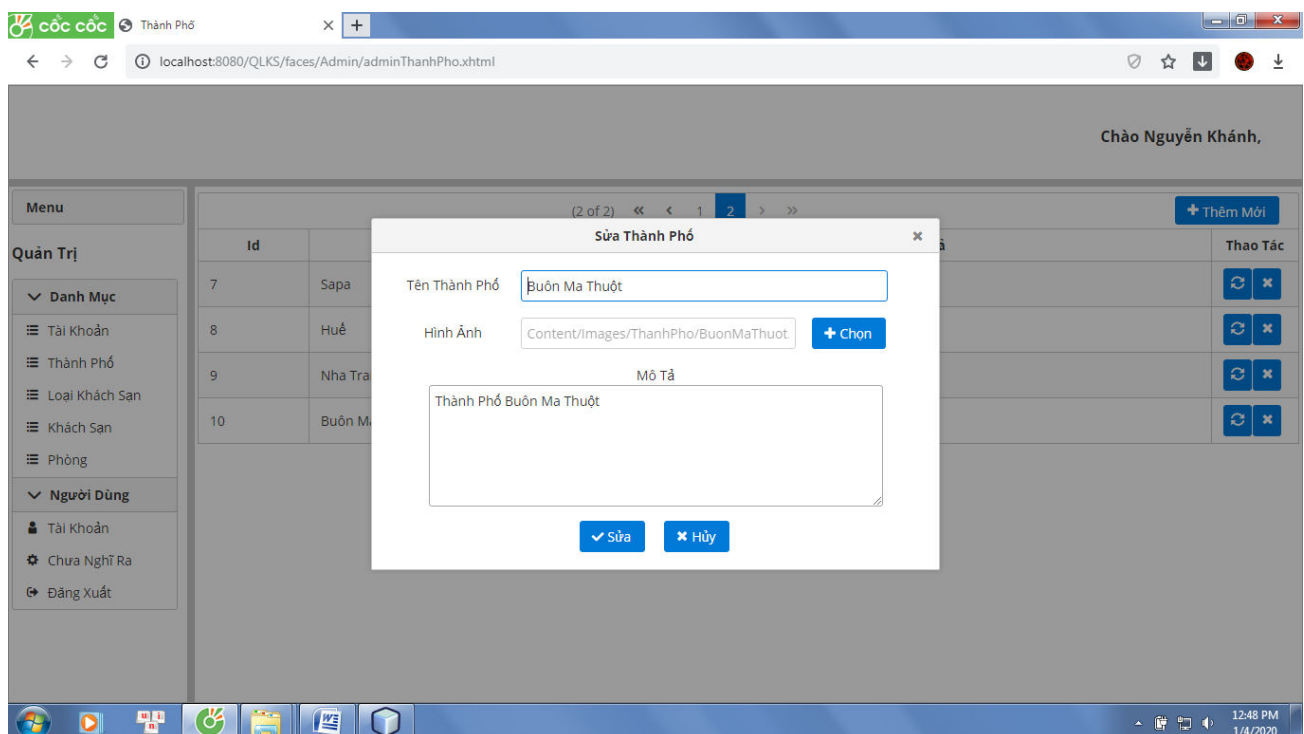
The screenshot shows a web application running in a browser. The address bar indicates the URL is `localhost:8080/QLKS/faces/Admin/adminTaiKhoan.xhtml`. The page title is "Tài Khoản". The main content area displays a table of users with the following columns: Tên Tài Khoản, Mật Khẩu, Họ Tên, Giới Tính, Số Điện Thoại, Email, Là Admin, and Thao Tác. The table contains 6 rows of data. A sidebar menu on the left lists various management options under "Quản Trị", including "Danh Mục" (with sub-items: Tài Khoản, Thành Phố, Loại Khách Sạn, Khách Sạn, Phòng) and "Người Dùng" (with sub-items: Tài Khoản, Chưa Nghĩ Ra, Đăng Xuất). The top right of the page shows a greeting: "Chào Nguyễn Khánh,". The bottom of the page shows a Windows taskbar with the time 12:44 PM on 1/4/2020.

Tên Tài Khoản	Mật Khẩu	Họ Tên	Giới Tính	Số Điện Thoại	Email	Là Admin	Thao Tác
BuiThuy	123456	Bùi Đình Thủy	Nam	0556878945	buihuy@gmail.com	Có	↺ ✖
LyChan	123456	Lý Văn Chân	Nam	0123455678	lychan@gmail.com	Có	↺ ✖
NgoThuong	123456	Ngô Văn Thường	Nam	0358763412	ngothuong@gmail.com	Không	↺ ✖
NguyenKhanh	123456	Nguyễn Khánh	Nam	0366918587	nguyenkhanh@gmail.com	Có	↺ ✖
User	123456	Người Dùng	Nữ	0358763412	user@gmail.com	Không	↺ ✖

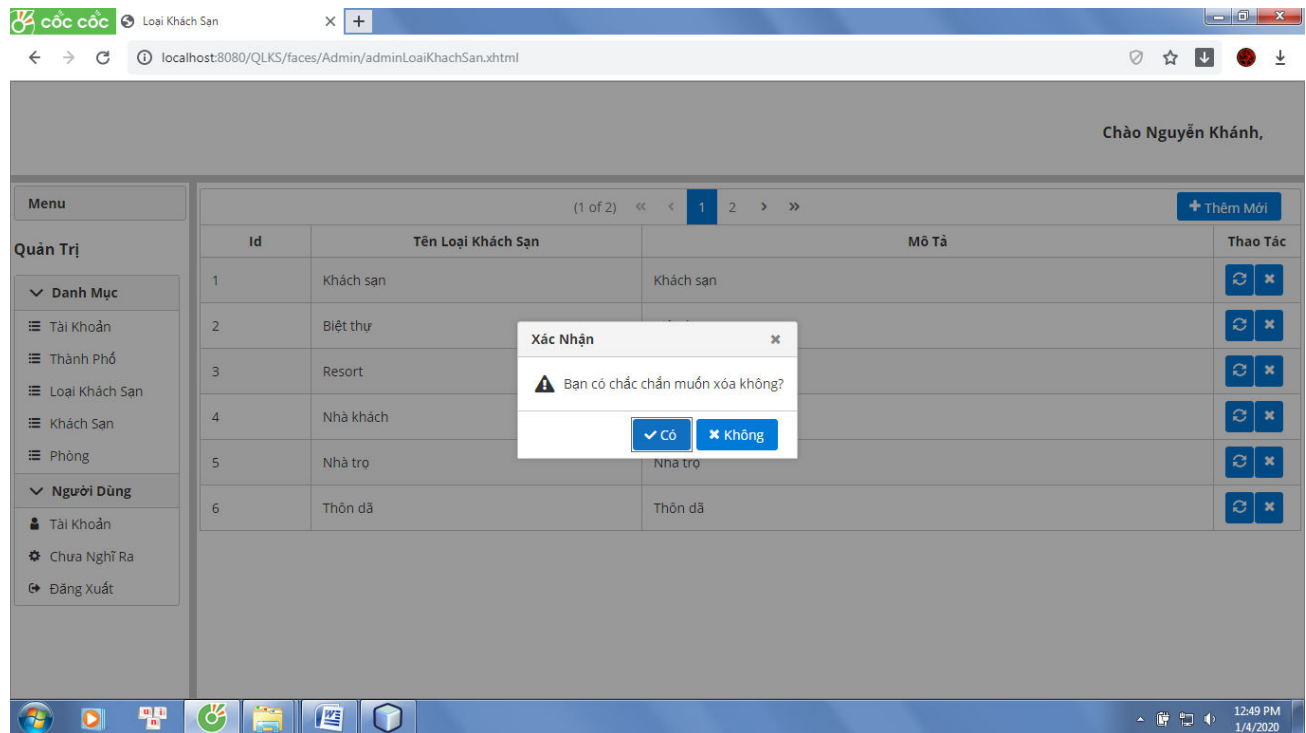
- Thêm một tài khoản, sử dụng một p:dialog để nhập thông tin, nếu để trống trường nào đó hoặc tài khoản đã tồn tại sẽ báo lỗi bằng message p:growl, nếu thêm thành công cũng sẽ có một message thông báo thành công



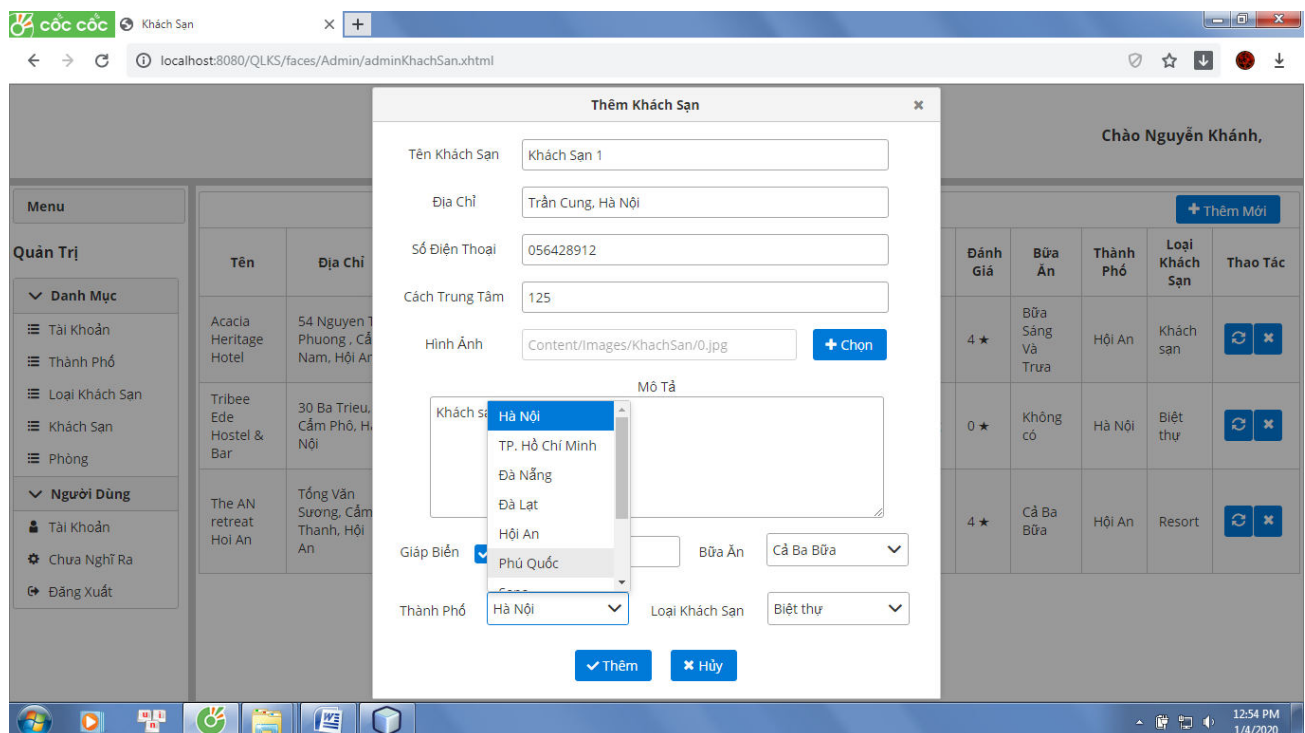
- Sửa một thành phố, upload ảnh có sử dụng p:fileUpload, ảnh được chọn sẽ được đưa vào web server và url trên web server sẽ được lưu vào CSDL



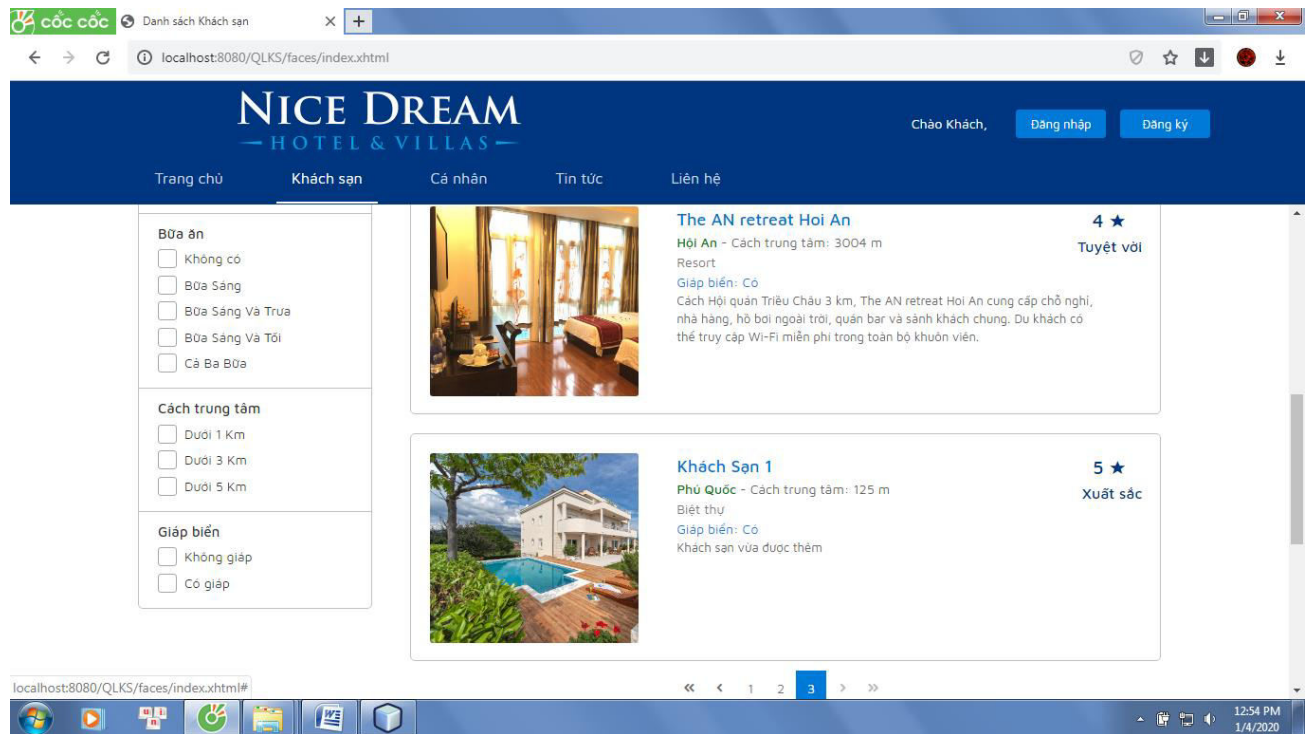
- Xóa một Loại Khách Sạn, yêu cầu xác nhận xóa bằng p:confirmDialog



- Khi thêm mới một khách sạn, có thể chọn Thành phố, Loại khách sạn trong p:selectOneMenu, bao gồm cả các Thành phố, Loại khách sạn vừa mới thêm, các p:selectOneMenu này lấy dữ liệu từ listThanhPho trong bean



- Sau khi đăng xuất và trở lại trang Khách sạn có thể thấy khách sạn vừa thêm



- Ngoài ra để ngăn chặn người ngoài truy cập đến trang quản trị, chẳng hạn người dùng nhập đường dẫn <http://localhost:8080/QLKS/faces/Admin/adminTaiKhoan.xhtml>, nhóm em có sử dụng một bộ lọc (MyFilter.java), nếu người dùng chưa đăng nhập hoặc đã đăng nhập nhưng không phải Admin mà cố truy cập vào trang Admin sẽ bị quay về trang chủ
- Website đã được up lên host, có thể tham khảo tại <http://whknightz.herokuapp.com/>