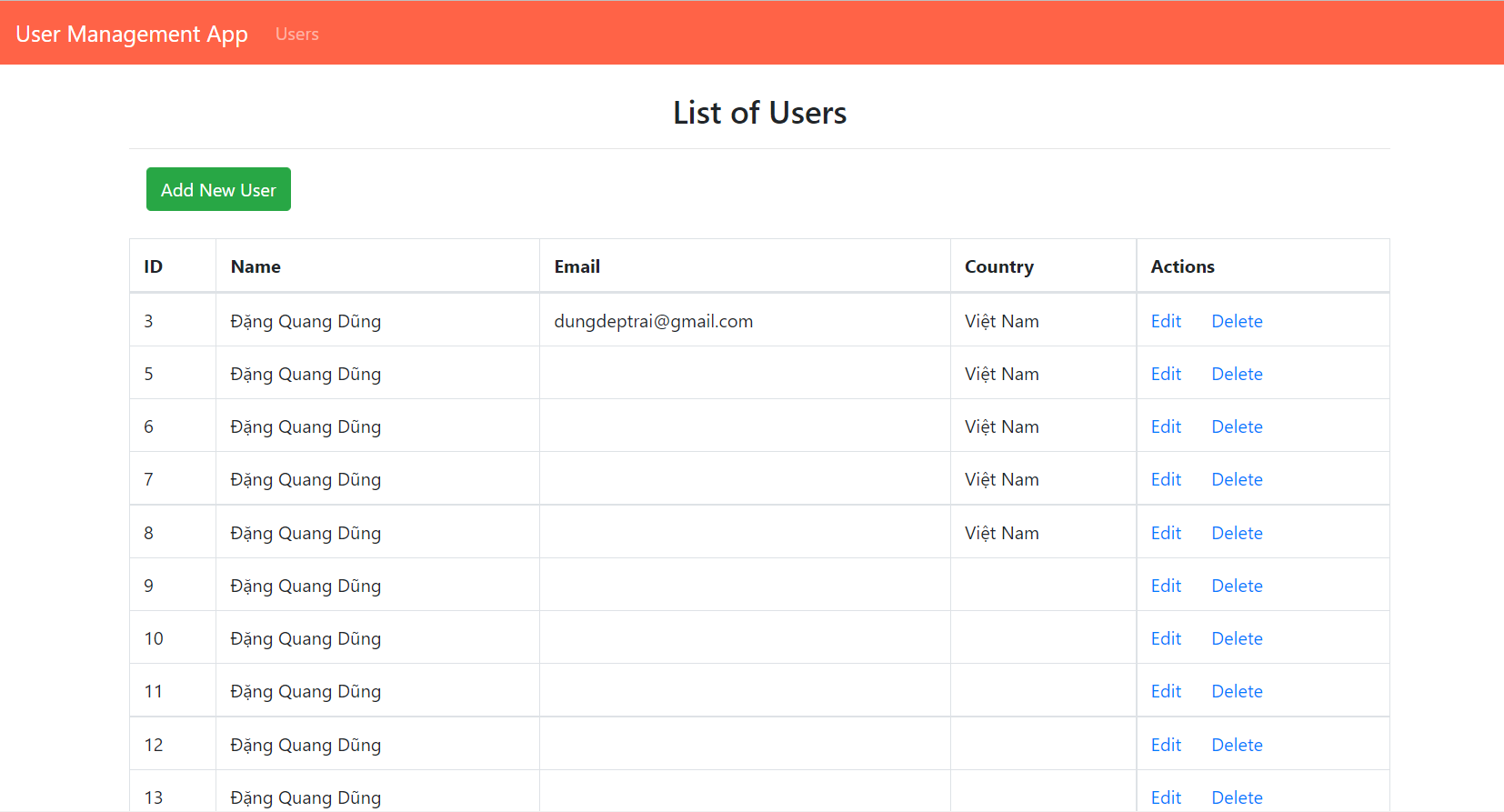
# 1,



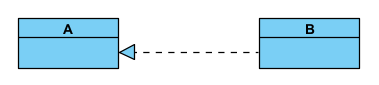
2,

# 3,

* JSP: là viết tắt của JavaServer Pages là một công nghệ để phát triển các trang web động. JSP giúp các nhà phát triển chèn java code vào các trang HTML bằng cách sử dụng các thẻ JSP đặc biệt.
* Servlet có thể được mô tả bằng nhiều cách, tùy thuộc vào ngữ cảnh:
* Servlet là một công nghệ được sử dụng để tạo ra ứng dụng web.
* Servlet là một API cung cấp các interface và lớp bao gồm các tài liệu.
* Servlet là một thành phần web được triển khai trên máy chủ để tạo ra trang web động.
* Có nhiều interface và các lớp trong API servlet như Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, ...
* DAO: Data Access Object (DAO) Pattern là một trong những Pattern thuộc nhóm cấu trúc (Structural Pattern). Mẫu thiết kế DAO được sử dụng để phân tách logic lưu trữ dữ liệu trong một lớp riêng biệt. Theo cách này, các service được che dấu về cách các hoạt động cấp thấp để truy cập cơ sở dữ liệu được thực hiện. Nó còn được gọi là nguyên tắc Tách logic (Separation of Logic). Ý tưởng là thay vì có logic tiếp xúc trực tiếp với cơ sở tài liệu, mạng lưới hệ thống file, dịch vụ web hoặc bất kể chính sách tàng trữ nào mà ứng dụng cần sử dụng, tất cả chúng ta sẽ để logic này sẽ tiếp xúc với lớp trung gian DAO. Lớp DAO này sau đó tiếp xúc với mạng lưới hệ thống tàng trữ, hệ quản trị CSDL như thực thi những việc làm tương quan đến tàng trữ và truy vấn tài liệu ( tìm kiếm, thêm, xóa, sửa, … ).
* JDBC: là viết tắt của Java Database Connectivity, là một Java API chuẩn để kết nối giữa ngôn ngữ lập trình Java và các cơ sở dữ liệu đa dạng. Sử dụng JDBC, bạn có thể thực hiện nhiều tác vụ đa dạng khi làm việc với cơ sở dữ liệu như tạo, xóa cơ sở dữ liệu; tạo và thực thi các lệnh SQL hoặc MySQL; tạo, xóa các bản ghi;
* 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 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.
* CRUD:
* Create: Thêm mới
* Read: Đọc từ CSDL
* Update: Cập nhật
* Delete: Xóa

# 4,

* Realization (Hiện thực hóa): Nói đơn giản thì là 1 class implement 1 interface. Interface sẽ thể hiện cấu trúc của đối tượng còn class sẽ dựa trên khung xương đó tạo ra các đối tượng cụ thể.



public interface A {

...

} // interface A

public class B implements A {

...

} // class B

* Generalization (khái quát hóa): Các đối tượng cụ thể sẽ thừa hưởng các thuộc tính và phương thức của đối tượng khái quát.



public class A {

...

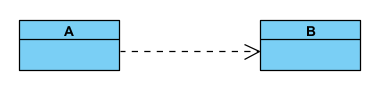
} // class A

public class B extends A {

....

} // class B

* Dependency (phụ thuộc): 2 lớp có quan hệ với nhau mà khi thay đổi ở lớp độc lập sẽ tạo ra thay đổi ở lớp phụ thuộc.

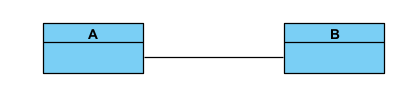


public class A {

public void doSomething(B b) { }

}

* Association (có quan hệ): Biểu thị 1 lớp có quan hệ với 1 lớp khác.

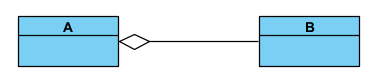


public class A {

public void doSomething(B b) { }

}

* Aggregation (thu nạp): Biểu thị 1 lớp B nằm trong phần thuộc tính của lớp A. Là 1 trong những thuộc tính của lớp A. Nhưng khi lớp A bị xóa bỏ thì lớp B không bị ảnh hưởng. Đối tượng thuộc lớp B có thể nằm trong nhiều đối tượng của lớp khác ngoài A.



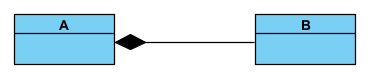
public class A {

private B bb;

public void setB(B b) { bb = b; }

}

* Composition (hợp thành): Biểu thị 1 lớp B nằm trong phần thuộc tính của lớp A. Là 1 trong những thuộc tính của lớp A. Nhưng khi lớp A bị xóa bỏ thì lớp B cũng sẽ bị xóa. Đối tượng thuộc lớp B chỉ có thể nằm trong 1 đối tượng thuộc lớp A.



public class A {

private B bb;

public A() {

bb = new B();

}

# 5,