Phần II: Client

1. **Sơ lược về Hệ điều hành Android và lý do sử dụng công nghệ Android.**
2. Sơ lược về Hê điều hành Android

Android là hệ điều hành dành cho các thiết bị di động, được phát triển bởi Google từ năm 2008. Android là một hệ điêu hành mã nguồn mở và miễn phí, được xây dựng trên nhân Linux, sử dụng Java làm ngôn ngữ để phát triển ứng dụng. Nhờ sự hộ trợ từ Google Android liên tục phát triển trong những năm qua, đến nay đã trở thành hệ điều hành trên thiết bị di động phổ biến nhất thể giới.

1. Lý do sử dụng công nghệ Android để phát triển ứng dụng Tìm nhà trọ sinh viên.

* Android là hệ điều hành dành cho thiết bị di động, phù hợp với yêu cầu của Project là phải cài đặt trên smartphone.
* Ứng dụng Android được phát triển bằng ngôn ngữ lập trình Java, đây là ngôn ngữ thân thiện dễ học, phù hợp với sinh viên khi mới tiếp xúc với ngành Công nghệ thông tin.
* Android là hệ điều hành mã ngồn mở, có sự hộ trợ lớn từ Google và cộng đồng phát triển, do đó tài liệu hướng dẫn về Android rất phong phú và đa dạng.
* Android hoàn toàn miễn phí vì vậy phù hợp với sinh viên khi sử dụng để học tập và thực hành.
* Android hỗ trợ rất nhiều nền tảng phần cứng khác nhau (do sử dụng Java) từ những thiết bị di động từ giá rẻ đến cao cấp, điều này khiến cho Android phổ biến và quen thuộc với đa số người dùng đặc biệt là đối tượng người dùng mà ứng dụng đang hướng tới chính là các bạn Sinh viên.

1. **Các công nghệ đã sử dụng**

Android sử dụng rất nhiều công nghệ, từ những công nghệ cơ bản đến những công nghệ tiên tiến hiện đại, tuy nhiên trong giới hạn môn học Project1 chúng em chỉ nêu ra những công nghệ mà mình đã sử dụng để xây dụng ứng dụng mà thôi.

1. Thiết kế giao diện, Activity và Fragment.
2. Thiết kế giao diện

Trong android phần giao diện sử dụng xml để thiết kế sau đó thông qua bộ dịch của hệ điểu hành để vẽ lên giao diện ứng dụng.

Hai thành phần quan trọng nhất tạo nên giao diện đó là *View* và *ViewGroup*. *View* là các thành phần có thể nhìn thấy trực tiếp trên màn hình như nút bấm(button), chữ viết hiển thị trên màn hình, ô text box ... *ViewGroup* là thành phần bao chứa các *View*, nó thường là các layout dùng để chứa các thành phần giao diện khác. Trên màn hình chỉ được phép có duy nhất *ViewGroup* được gọi là gốc, bên trong *ViewGroup* này có thể đặt các *View* hoặc các *ViewGroup* khác.

Để mô tả các thành phần giao diện, trong android sử dụng định dạng file xml với các thẻ được android định nghĩa sẵn.

Dưới đây là source code của một file xml mô tả giao diện:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"**

**android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>

<**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"   
 android:text="Hello world"**/>

</**LinearLayout**>

Trong giao diện này có một *ViewGroup* ngoài cùng là *LinearLayout* chứa trong nó là một *View* là *TextView* dùng để hiển thị ra màn hình dòng chữ “Hello World”.

Trong android có nhiều lớp để mô tả các layout khác nhau như *LinearLayout*, *RelativeLayout*, *TableLayout* ... và các view để mô tả các thành phần giao diện như *TextView*, *EditText*, *Button*, *CheckBox*, *ListView* ...

Các file xml mô tả giao diện của ứng dụng được lưu trữ mặc định trong thư mục app/src/main/res/layout.

1. Activity

Activity được coi là thành phần quan trọng bậc nhất trên android, nó tạo ra giao diện tương tác với người sử dụng, người dùng thông qua Activity có thể tương tác với ứng dụng.

|  |  |
| --- | --- |
| Mỗi một Activity có một giao diện được mô tả bằng file xml lưu trong thư mục layout.  Hình bên mô tả vòng đời của một Activity.  Vòng đời của Activity bắt đầu khi hàm onCreat() được gọi, Activity sẽ vẽ lên giao diện ứng dụng từ những mô tả có trong file xml tương ứng nằm trong thư mục layout, đồng thời tạo các đối tượng nghe để lắng nghe các sự kiện khi người dùng tương tác với Activity.  Activity kết thúc khi hàm onDestroy() được gọi. |  |

1. Fragment

|  |  |
| --- | --- |
| Fragment là một thành phần trong giao diện người dùng hoặc hành vi của một ứng dụng.  Fragment cũng có giao diện riêng của nó, cũng có các hành vi và vòng đời riêng(giống với Activity).  Fragment có thể được đặt trong Activity nó cho phép tạo ra Activity có nhiều modul. Thay vì chỉ thiển thị được 1 Activity trên màn hình, ta có thể chia màn hình thành nhiều phần nhỏ, mỗi phần nhỏ đó được quản lý bởi một fragment.  Fragment thường được sử dụng khi kích thước màn hình lớn (kích cỡ tablet) để tận đụng được tối đa kích thước màn hình, hoặc khi xoay ngang màn hình.  Một Fragment có thể sử dụng trong nhiều Activity làm tăng khả năng tái sử dụng mã nguồn cho chương trình.  Trong ứng dụng *Tìm nhà trọ sinh viên* Fragment được sử dụng cùng với *DrawerLayout* để tạo ra phần Menu và Tìm kiếm. |  |

1. Xml
2. Sơ lược về Xml

Xml (*eXtensible Markup Language*, “Ngôn ngữ đánh dấu mở rộng”) được đề xuất với mục đích chung do W3C đề nghị. Mục đích của Xml là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau. Đặc biệt là các hệ thống chia sẻ dữ liệu qua internet.

Xml cung cấp một phương tiện dùng văn bản (*text*) để mô tả thông tin và áp dụng một cấu trúc kiểu cây cho thông tin đó. Tại mức căn bản, mọi thông tin đều thể hiện dưới dạng text, chen giữa là các thẻ đánh dấu (*markup*) với nhiệm vụ ký hiệu sự phân chia thông tin thành một cấu trúc có thứ bậc của các dữ liệu các phần tử dùng để chứa dữ liệu, và các thuộc tính của các phần tử đó.

Do tính độc lập không phụ thuộc hệ điều hành của Xml nên chúng em quyết định sử dụng Xml để truyền tải thông tin từ Server về Client. Server sẽ gửi dữ liệu cho Client dưới đạng Xml, Client sẽ đọc file Xml nhận được để lấy được thông tin cần thiết.

1. Phân tích cú pháp Xml trong Android

Trong Android có nhiều cách để phân tích cú pháp Xml, trong ứng dụng này sử dụng mô hình DOM.

DOM là một tiêu chuẩn được định nghĩa bởi tổ chức W3C, DOM được dùng trên mọi nền tảng và mọi ngôn ngữ, không chỉ Java mà cả C++, C# ... cũng hỗ trợ DOM.

Với mỗi tài liệu Xml mô hình DOM sẽ chuyển nó về dạng cây của các Object. Mỗi Object sẽ tương ứng với một Node trên cây, trong đó Node gốc của cây chính là *Document*.

Dưới đây là phần source code để lấy về thông tin từ file Xml như sau:

<result>

<tenHienThi>[string]</tenHienThi>

<email>[string]</email>

<tinhTrang>[string]</tinhTrang>

</result>

**public** Void layDocument(String linkFileXml) {  
 **try** {  
 DocumentBuilder db = DocumentBuilderFactory.*newInstance*().newDocumentBuilder();  
 Document document = db.parse(linkFileXml);

NodeList listResult = document.getElementsByTagName(**"result"**); // lấy về danh sách các Node có TagName là “result”

Node result = listResult.item(0); // lấy về Node result

NodeList thongTinCaNhan = result.getChildNodes(); // lấy về danh sách các con của result

String tenHienThi = thongTinCaNhan.item(0).getTextContent(); // lấy tên hiển thị.

String email = thongTinCaNhan.item(1).getTextConTent(); lấy email.

String tinhTrang = thongTinCaNhan.item(2).getTextConTent(); // lấy tình trạng.  
 } **catch** (Exception e){  
 **return null**;  
 }  
}

1. **Tài liệu tham khảo**
2. <https://developer.android.com/intl/vi/training/index.html>
3. <http://grail.cba.csuohio.edu/~matos/notes/cis-493/lecture-notes/slides/_showFiles.php>
4. Tài liệu, slide: Lập trình Android – Thầy giáo Nguyễn Hồng Quang.