

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO PROJECT 2

Đề tài: “Tìm hiểu và xây dựng ứng dụng đọc tin tức với RSS”

Giáo viên hướng dẫn: TS. Lê Bá Vui

Sinh viên thực hiện: Nguyễn Duy Thành

Mã số sinh viên: 20144081

---Hà Nội, tháng 5 năm 2017---

Mục lục

Mục lục	2
Lời cảm ơn.....	3
Danh mục hình ảnh.....	3
Danh mục phần mềm thực hiện	3
1.Phân tích yêu cầu, chức năng:	4
2.Phân tích thiết kế hệ thống và giao diện:.....	4
2.1.Biểu đồ trình tự:	4
2.2 Biểu đồ lớp:.....	5
2.3. Thiết kế giao diện:	6
3.Quá trình thực hiện chi tiết	7
3.1. Xây dựng lớp tin tức:	7
3.2. Đọc nội dung từ URL:	7
3.3. Kỹ thuật DOM trong xử lý XML:.....	7
3.4. Sử dụng kỹ thuật DOM trong phân tích RSS tin tức:	8
3.5. Hiển thị tin tức lên giao diện của chương trình:	9
3.6. Xử lý sự kiện chọn đọc một tin tức:.....	10
3.7.Một số xử lý khác:	10
4.Một số hình ảnh thực tế	11

Lời cảm ơn

Để thực hiện project 2, em xin cảm ơn TS.Lê Bá Vui đã giúp em lựa chọn và hoàn thiện đề tài này. Với khả năng và thời gian hạn chế, sản phẩm môn học của em còn nhiều thiếu sót. Em sẽ tiếp tục trau dồi và hoàn thiện sản phẩm hơn sau ôn học này.

Em xin chân thành cảm ơn!

Danh mục hình ảnh

Hình ảnh	Mô tả
2.1	Biểu đồ trình tự mở trình đọc News
2.2	Biểu đồ trình tự đọc tin tức
2.3	Biểu đồ lớp toàn hệ thống
2.4	Thiết kế giao diện của ứng dụng
3.1	Các thuộc tính của tin tức
3.2	Hàm đọc nội dung từ đường dẫn URL
3.3	Kỹ thuật XMLDomParser
3.4	Kỹ thuật XMLDomParser
3.5	Ứng dụng DOM trong xử lý tệp tin tức định dạng RSS
3.6	NewsAdapter - hiển thị nội dung lên trang chủ
3.7	Xử lý sự kiện chọn đọc tin tức
3.8	Xử lý hiển thị tin tức yêu cầu
4.1	Màn hình Android
4.2	Giao diện chính
4.3	Giao diện đọc tin tức

Danh mục phần mềm thực hiện

1. Android Studio : IDE lập trình Android.
2. Geny motion : công cụ giả lập máy ảo Android.
3. Astah community : công cụ vẽ UML.
4. Balsamiq Mockups : công cụ vẽ giao diện mẫu.

1. Phân tích yêu cầu, chức năng:

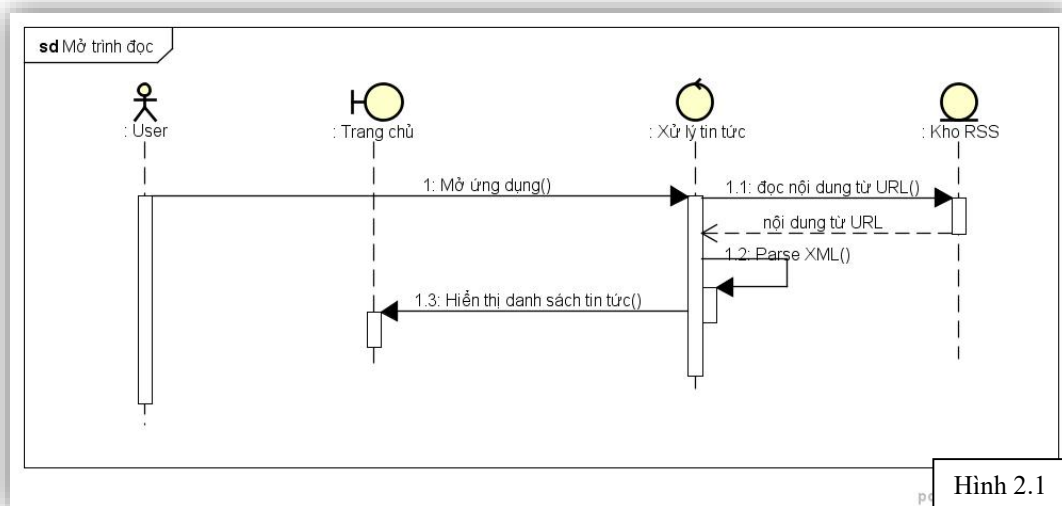
Theo yêu cầu đề tài, cần xây dựng một ứng dụng đọc tin tức với nguồn cấp là các trang RSS sẵn có của các trang báo trực tuyến. RSS là một tiêu chuẩn định dạng tài liệu dựa trên XML nhằm giúp người sử dụng dễ dàng cập nhật và tra cứu thông tin một cách nhanh chóng và thuận tiện nhất bằng cách tóm lược thông tin vào trong một đoạn dữ liệu ngắn gọn, hợp chuẩn. Từ đó, ứng dụng sẽ được xây dựng trên nền tảng Android với các chức năng như sau:

- Lấy về và phân tích trang RSS từ website.
- Hiển thị các tin tức theo nguồn cấp.
- Hiển thị nội dung bài viết khi người dùng chọn xem một tin tức.

2. Phân tích thiết kế hệ thống và giao diện:

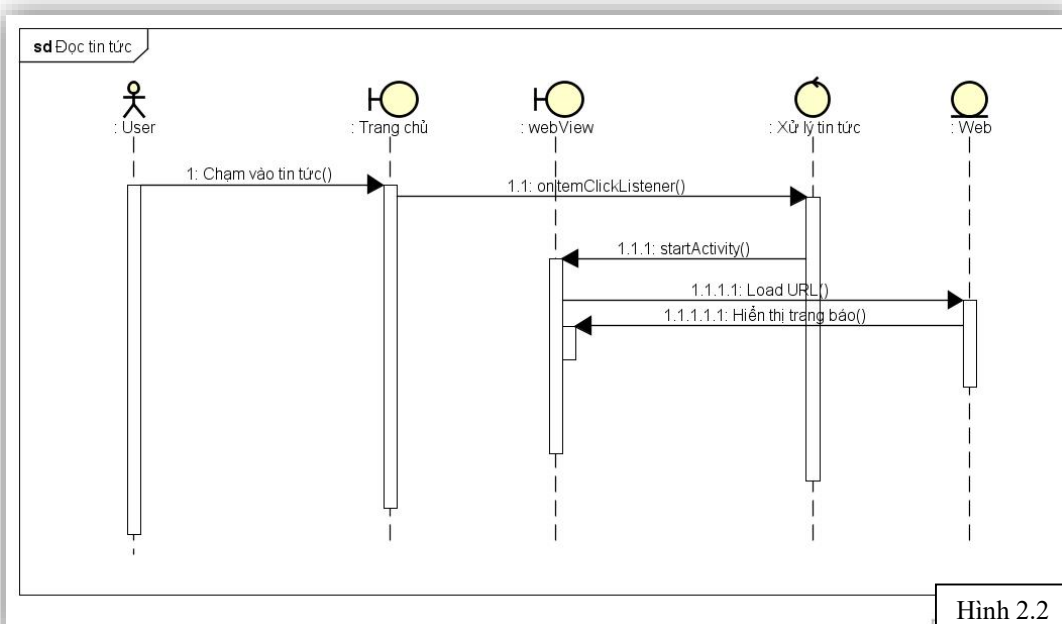
2.1. Biểu đồ trình tự:

a) Trình tự mở trình đọc



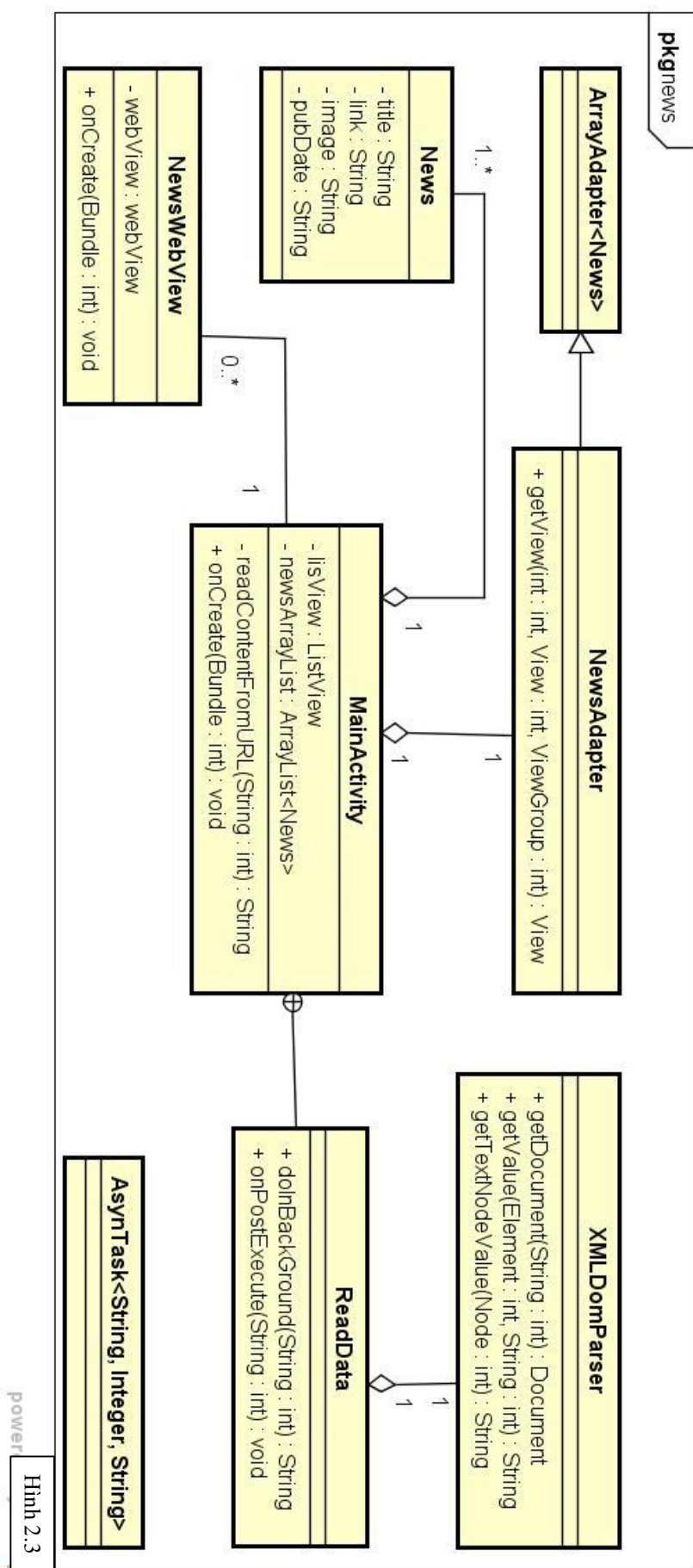
Hình 2.1

b) Trình tự đọc tin tức



Hình 2.2

2.2 Biểu đồ lớp:



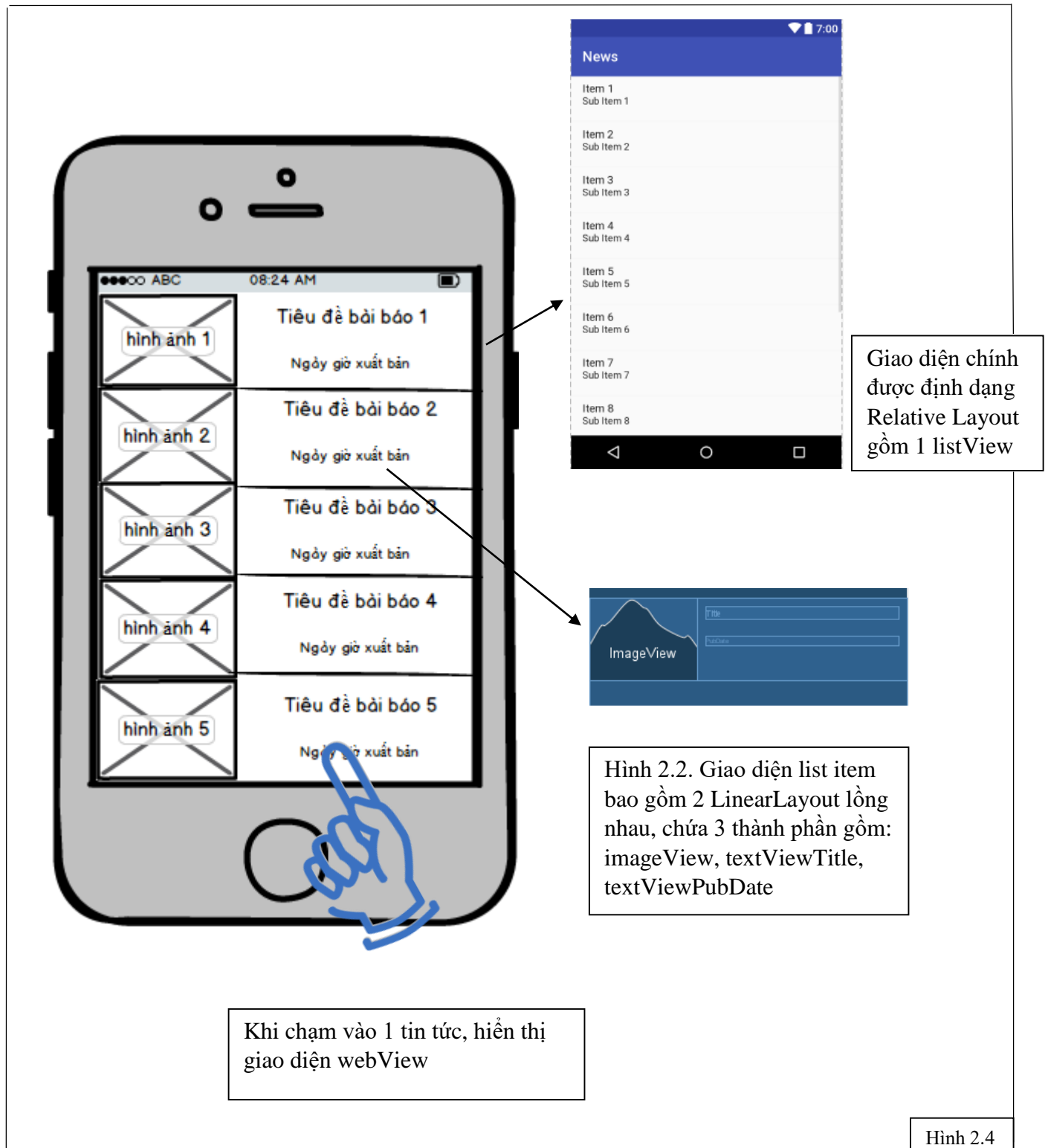
Hình 2.3

Biểu đồ lớp của ứng dụng :

- Lớp Main Activity đóng vai trò xử lý chính.
- Lớp News dùng để khởi tạo đối tượng tin tức.
- Lớp ReadData sử dụng đối tượng XMLDomParser để parse dữ liệu và chuyển sang đối tượng NewsAdapter xử lý lên giao diện.
- Lớp NewsWebView có chức năng nhận link và tải tin tức sau đó hiển thị.

2.3. Thiết kế giao diện:

Dưới đây là hình ảnh mẫu về giao diện chính của ứng dụng đọc tin tức từ trang RSS. Từ mẫu này, ta đi đến các file xml thiết kế giao diện của chương trình.



3. Quá trình thực hiện chi tiết

Phần lập trình chi tiết được gửi trong tệp đính kèm. Sau đây là những bước chính để trong xử lý dữ liệu RSS để xây dựng ứng dụng đọc tin tức.

3.1. Xây dựng lớp tin tức:

Lớp tin tức gồm các thuộc tính của một ô tin tức, bao gồm: tiêu đề, đường dẫn, ảnh, thời gian xuất bản của tin tức và các phương thức khởi tạo có tham số và các getter tương ứng với các thuộc tính.

```
public class News {  
  
    private String title;  
    private String link;  
    private String image;  
    private String pubDate;
```

Hình 3.1

3.2. Đọc nội dung từ URL:

Để xử lý được file RSS, bước đầu tiên cần lấy được nội dung file RSS về. Để thực hiện công việc này, cần xây dựng hàm `readContentFromURL(String theUrl)`. Tham số đầu vào là đường dẫn URL của file RSS. Kết quả trả về của hàm là nội dung file RSS ở dạng chuỗi ký tự được lưu vào một `StringBuilder`. Bởi ở đây xử lý trên một luồng nên dùng `StringBuilder` cho hiệu suất cao hơn so với dùng `String` và `StringBuffer`.

Trong hàm, cần tạo một kết nối `URLConnection` tới file và dùng `bufferedReader` để đọc nội dung từ kết nối đó. Dùng `bufferedReader` ở đây vì `bufferedReader` là một lớp gọn nhẹ, cung cấp hàm `readLine()` để đọc từ file một cách dễ dàng.

```
private static String readContentFromURL(String theUrl) {  
    StringBuilder content = new StringBuilder();  
    try {  
  
        URL url = new URL(theUrl);  
        URLConnection urlConnection = url.openConnection();  
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));  
        String line;  
        while ((line = bufferedReader.readLine()) != null) {  
            content.append(line).append("\n");  
        }  
        bufferedReader.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return content.toString();  
}
```

Hình 3.2

3.3. Kỹ thuật DOM trong xử lý XML:

Về bản chất, RSS là một file XML. Để làm việc với XML, chúng ta có 2 phương pháp: SAX và DOM. Ứng dụng này sử dụng kỹ thuật DOM:

- Ưu điểm : Truy cập file XML như cấu trúc cây, các thành phần được lưu trữ như các đối tượng nên dễ sử dụng.
- Nhược điểm : Toàn bộ file XML được load vào bộ nhớ nên gây nặng nề trong lưu trữ.

Ta xây dựng lớp XMLDomParser gồm 3 phương thức :

- **public Document** getDocument(String xml): phương thức này dùng đối tượng tạo từ DocumentBuilder() để parse() chuỗi XML đã đọc được sang dạng Document.

```
DocumentBuilder db = factory.newDocumentBuilder();
InputSource is = new InputSource();
is.setCharacterStream(new StringReader(xml));
is.setEncoding("UTF-8");
document = db.parse(is);
```

Hình 3.3

- **public String** getValue(Element item, String name)
 - **private String** getTextNodeValue(Node element)
- Hai phương thức này dùng để đọc nội dung các thành phần theo tên thẻ trong XML.

```
public String getValue(Element item, String name) {
    NodeList nodes = item.getElementsByTagName(name);
    return this.getTextNodeValue(nodes.item(0));
}

private String getTextNodeValue(Node element) {
    Node child;
    if (element != null) {
        if (element.hasChildNodes()) {
            for (child = element.getFirstChild(); child != null; child = child.getNextSibling()) {
                if (child.getNodeType() == Node.TEXT_NODE) {
                    return child.getNodeValue();
                }
            }
        }
    }
    return "";
}
```

Hình 3.4

3.4. Sử dụng kỹ thuật DOM trong phân tích RSS tin tức:

Đầu tiên, ta xây dựng lớp ReadData kế thừa AsyncTask. Vì, MainThread hoạt động theo cơ chế FIFO, nếu để tác vụ ngầm này chạy trong đó sẽ làm ứng dụng bị treo do có thời gian xử lý lâu. Lớp này ghi đề 2 phương thức:

- **protected String** doInBackground(String... params): phương thức này gọi tới phương thức đọc nội dung từ URL đã xây dựng ở mục 3.2.
- **protected void** onPostExecute(String xmlString): phương thức này thực hiện các chức năng: dùng DOM để lấy giá trị; tạo các đối tượng News và thêm vào 1 ArrayList; setAdapter để hiển thị ArrayList này lên màn hình.

Qua phân tích file RSS, ta thấy các thông tin cần lấy như sau:

- Tiêu đề, đường dẫn, ngày xuất bản : nằm lần lượt trong các thẻ <title>, <link>, <pubDate>
- Đường dẫn ảnh nằm trong thẻ <description> với nhiều thẻ bên trong.
Sử dụng lớp XMLDomParser ta lấy được các dữ liệu trên:

```
XMLDomParser parser = new XMLDomParser();
Document document = parser.getDocument(xmlString);
super.onPostExecute(xmlString);

NodeList listOfItem = document.getElementsByTagName("item");
NodeList listOfDescription = document.getElementsByTagName("description");
String title;
String link;
String image = "";
String pubDate;
for (int i = 0; i < listOfItem.getLength(); i++) {
    String cData = listOfDescription.item(i + 1).getTextContent();
    Pattern pattern = Pattern.compile("<img[^>]+src\\s*=\\s*['\"]([^\"]+)['\"][^>]*>");
    Matcher matcher = pattern.matcher(cData);
    if (matcher.find()) {
        image = matcher.group(1);
    }
    Element element = (Element) listOfItem.item(i);
    title = parser.getValue(element, "title");
    link = parser.getValue(element, "link");
    pubDate = parser.getValue(element, "pubDate");
    newsArrayList.add(new News(title, link, image, pubDate));
}
```

Hình 3.5

3.5. Hiển thị tin tức lên giao diện của chương trình:

Để hiển thị các tin tức lên giao diện của ứng dụng, ta kế thừa từ lớp ArrayAdapter. Trong phương thức getView(), để load ảnh, ta sử dụng thư viện Picasso.

```
public View getView(int position, View convertView, ViewGroup parent) {
    View view = convertView;
    if (view == null) {
        LayoutInflater inflater = LayoutInflater.from(getContext());
        view = inflater.inflate(R.layout.news_list_view, null);
    }
    News p = getItem(position);
    if (p != null) {
        TextView textViewTitle = (TextView) view.findViewById(R.id.textViewTitle);
        textViewTitle.setText(p.getTitle());

        TextView textViewPubDate = (TextView) view.findViewById(R.id.textViewPubDate);
        textViewPubDate.setText(p.getPubDate());

        ImageView imgView = (ImageView) view.findViewById(R.id.imageView);
        Picasso.with(getContext()).load(p.getImage()).into(imgView);
    }
    return view;
}
```

Hình 3.6

3.6. Xử lý sự kiện chọn đọc một tin tức:

Để xử lý sự kiện này, ta sử dụng kỹ thuật Intent gửi yêu cầu đến lớp NewsWebView. Lớp NewsWebView nhận đường link tin tức và hiển thị bài báo lên webView.

Ta được hàm onCreate() trong MainActivity như sau:

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    listView = (ListView) findViewById(R.id.listView);  
    newsArrayList = new ArrayList<>();  
    runOnUiThread(new Runnable() {  
        @Override  
        public void run() {  
            new ReadData().execute("http://vnexpress.net/rss/thoi-su.rss");  
        }  
    });  
    listView.setOnItemClickListener((parent, view, position, id) → {  
        Intent intent = new Intent(MainActivity.this, NewsWebView.class);  
        intent.putExtra("link", newsArrayList.get(position).getLink());  
        startActivity(intent);  
    });  
}
```

Hình 3.7

Tại lớp NewsWebView, đường dẫn tin tức được xử lý như sau:

```
String link = intent.getStringExtra("link");  
webView.loadUrl(link);  
webView.setWebViewClient(new WebViewClient());
```

Hình 3.8

3.7. Một số xử lý khác:

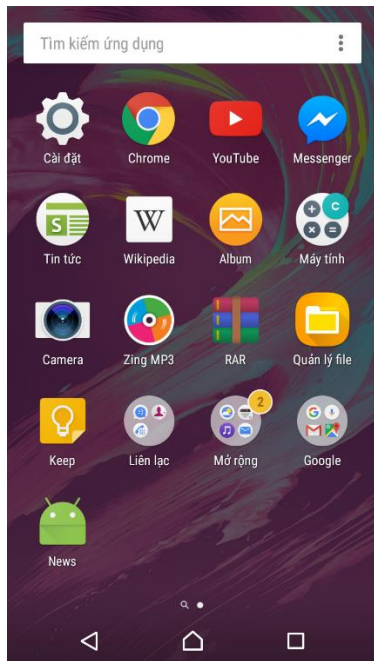
Ứng dụng đọc tin tức từ RSS lấy dữ liệu từ web, vì vậy ta cần yêu cầu quyền sử dụng Internet trong AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
```

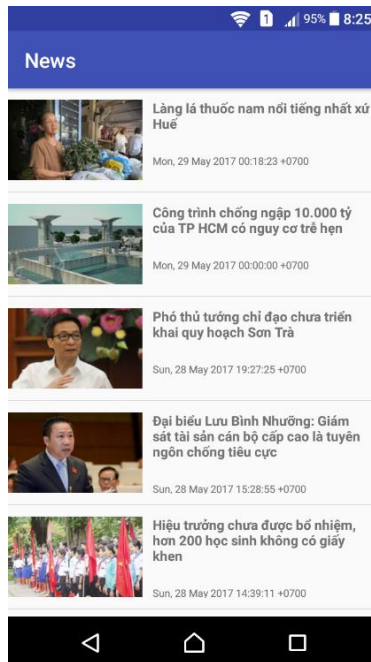
Trong phần hiển thị hình ảnh, ta sử dụng thư viện Picasso để load ảnh từ web, thông tin này cần được khai báo trong file build.gradle

```
compile 'com.squareup.picasso:picasso:2.5.2'
```

4. Một số hình ảnh thực tế



Hình 4.1



Hình 4.2



Hình 4.3

Hình 4.1. Ta thấy icon ứng dụng News

Hình 4.2. Khi click vào icon, ứng dụng kết nối tới internet và load lên toàn bộ tin tức

Hình 4.3. Khi click vào 1 tin tức, giao diện đọc tin tức được thực hiện.