

项目总结

学号	2021329600006	姓名	陈昊天	手机号码	13456982338
班级	21 计科 4 班	实验室	10-414	邮箱	admin@vayki.com
项目 成果 摘要	项目名称:	网页转 PDF 转换器应用程序			
	本项目旨在开发一款 Android 应用程序，使用户能够将网页内容快速转换成 PDF 格式，便于长期保存和分享特定的网页内容。应用通过 Android 原生开发技术实现，数据存储采用 SQLite 数据库，确保了信息的本地化和安全性。				
	开发过程中，我们重点关注了用户体验设计，提供了清晰直观的界面，并实现了基本功能如网页转 PDF、历史记录查看和删除等。我们采用了 Android Studio 作为开发环境，主要使用 Java 语言编写代码，同时利用 Git 进行版本控制。				
	从技术角度来看，我们利用了 Android 的 WebView 组件来加载和展示网页内容，并通过 PrintDocumentAdapter 和 PrintManager 实现了从网页到 PDF 的转换。我们确保网页的原始布局和格式被准确保留，提供了与网页相似的 PDF 阅读体验。应用程序还能够自动记录用户的转换历史，包括网页 URL、转换时间和 PDF 文件大小等信息。在软件测试方面，我们对关键功能如网页加载、转换质量、历史记录功能等进行了详细的测试，以确保应用的稳定性和可靠性。				
	本项目成功地实现了一个功能全面、操作直观、性能优异的网页转 PDF 转换器应用程序。它不仅满足了用户对于在线内容长期保存和分享的需求，而且提供了一个高效的工具，帮助不同领域的专业人士更好地管理和利用网络资源。				
2023 年 12 月 27 日					
教师评 分	评分要点:				
	1. 设计报告清晰表达了软件全周期开发流程、基本设计开发方法(技术)、设计开发的成果，内容详实；并重视手机应用软件用户体验设计。				
	2. 软件项目具有一定的复杂度，设计报告表述清晰，格式规范。				
成绩: 1. () 2. ()					
年 月 日					

网页转 PDF 转换器应用程序项目总结报告

学号：2021329600006 姓名：陈昊天 班级：21 计科 4 班

1 项目开发的背景和意义

1.1 项目开发的背景

随着互联网信息量的迅速增长，人们越来越依赖网络获取知识与信息。网络信息的临时性与易变性使得长期保存和分享特定网页内容成为一项挑战。因此，开发一款能够将网页内容快速转换成 PDF 格式的应用程序，对于保留和分享网络信息具有重要意义。

本项目的核心是开发一个网页转 PDF 转换器应用程序，该程序允许用户将网页内容方便地转换成 PDF 格式。这不仅有助于用户保存重要信息，也方便了文档的打印和离线阅读。在教育、研究、商业等领域，这种能力尤为重要，因为它有助于保留关键的在线资源和参考资料。

1.2 社会意义与经济价值

社会意义上，这款应用程序有助于信息的长期保存和分享，特别是对于那些需要离线阅读或打印文档的用户。它还可以作为教学和研究工具，帮助学生和研究人员收集和组织网络资料。经济价值方面，该应用程序可以吸引广泛的用户群体，包括学生、教师、研究人员和商业专业人士，为开发者带来潜在的盈利机会。

1.3 可行性分析

本项目选择 Android 平台作为开发环境，考虑到其广泛的用户基础和成熟的开发生态。项目核心技术包括利用 Android 的 WebView 组件来加载和展示网页内容，以及使用 PrintDocumentAdapter 和 PrintManager 实现网页到 PDF 的转换。这些技术在 Android 5.0 及以上版本中得到了广泛支持和验证。另外，项目还采用 SQLite 数据库来存储用户的历史记录，SQLite 以其轻量级和高效性在移动设备上应用广泛。

在开发工具方面，项目选用了 Android Studio，这是一款官方推荐且功能全面的开发环境，特别适合 Android 应用的开发^{[1][2]}。项目主要使用 Java 语言进行编码，Java 的成熟和稳定为项目提供了强大的技术支持。项目还利用 Git 进行版本控制，这有助于提高代码的管理效率和团队间的协作。

从市场需求的角度来看，随着数字化时代的到来，人们对于在线内容的长期保存和分享有着迫切需求，尤其是在教育、研究和商业领域。本项目旨在满足这一需求，通过将网页内容转换为 PDF 格式，方便用户存档和离线阅读。目标用户群体包括学生、教师、研究人员和商业专业人士，他们需要将网络信息用于学术研究、教学或商业报告。

在项目开发过程中，技术风险主要涉及网页兼容性和 PDF 转换的稳定性问题。这些风险可以通过持续测试和优化来控制 and 降低。市场风险方面，尽管市场对此类应用的需求强烈，但面临的挑战在于如何在众多竞争产品中脱颖而出。通过提供独特的功能和卓越的用户体验，本项目有望在目标市场中占据一席之地。

2 需求分析与数据库设计

2.1 需求分析

(1) 基本功能需求

网页转 PDF：用户能够通过输入或搜索网页 URL，并将其转换为 PDF 格式。转换过程应保持网页的布局和内容。

历史记录查看：应用应提供历史记录功能，允许用户查看之前转换的网页和生成的 PDF 文件。历史记录应包括 URL、转换日期和文件大小等信息。

删除历史记录：用户应能够删除历史记录中的特定项或全部记录。

(2) 用户界面需求

简洁直观的用户界面：应用应具备易于导航和操作的界面。主要功能如网页加载、PDF 转换应一目了然。

进度指示：在网页加载或转换为 PDF 过程中，应显示进度条或其他形式的进度指示，

以提高用户体验。

(3) 性能需求

响应速度: 应用程序应能快速响应用户操作, 包括网页加载和 PDF 转换。

转换质量: 网页到 PDF 的转换应准确无误, 尽可能保留原始网页的格式和内容。

(4) 数据存储需求

使用 SQLite 数据库存储用户的转换历史, 包括网页 URL、域名、转换日期等信息。

(5) 安全性和隐私需求

用户数据保护: 确保用户数据的安全性, 不泄露用户的浏览和转换记录。

数据加密: 对敏感信息如 URL 进行加密处理, 以保护用户隐私。

2.2 数据库设计

(1) 数据库表设计

android_metadata: 存储 Android 系统相关的元数据。

pdf_data: 存储 PDF 文件的具体信息。

domain_data: 存储每个域名相关的统计信息。

(2) pdf_data 表结构

pdf_id (INTEGER): 唯一标识 PDF 记录的 ID。

URL (TEXT): 转换为 PDF 的网页 URL。

domain (TEXT): 网页所属的域名。

file_size (INTEGER): 生成的 PDF 文件大小。

creation_date (TEXT): PDF 文件的创建日期。

(3) domain_data 表结构

domain (TEXT): 域名。

pdf_count (INTEGER): 该域名下转换的 PDF 数量。

total_pdf_size (INTEGER): 该域名下所有 PDF 的总大小。

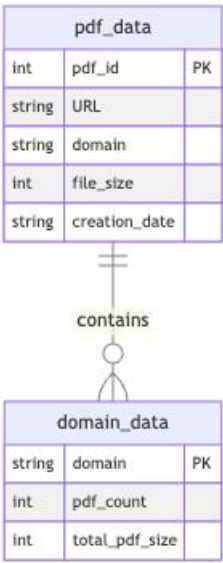


图 2-1 数据库 ER 图

(4) 数据库操作

插入操作：用户每次转换 PDF 时，将记录插入 pdf_data 表。

更新操作：当新的 PDF 生成时，更新 domain_data 表中的统计信息。

删除操作：用户删除历史记录时，相应地从 pdf_data 表中删除记录。

(5) 数据完整性和一致性

确保所有的插入、更新和删除操作都能维持数据的一致性和完整性。

使用事务处理来保证操作的原子性。

3 软件功能与架构设计

3.1 功能分析

(1) 网页浏览与搜索功能

用户可以通过输入 URL 或使用内置搜索功能访问网页。集成的 WebView 组件用于加载和显示网页，提供了与浏览器类似的体验。支持常见网页格式和多媒体内容，确保用户在应用内能够访问各类网站。

(2) 网页转 PDF 功能

用户可以将正在浏览的网页一键转换为 PDF 格式。利用 Android 的 PrintDocumentAdapter 和 PrintManager 实现从网页到 PDF 的转换。转换过程中，保留网页的原始布局和格式，确保 PDF 文件的阅读体验与网页相似。提供 PDF 预览功能，用户在保存 PDF 前可以预览最终效果。

(3) 历史记录管理

应用自动记录用户的转换历史，包括网页 URL、转换时间和 PDF 文件大小。用户可以方便地浏览、搜索和管理历史记录，快速找到过往的转换记录。支持一键清除历史记录，保护用户隐私。

(4) 用户界面

提供一个简洁直观的用户界面，易于新用户上手。主界面集成网页浏览、PDF 转换和历史记录查看等功能，操作流畅自然。用户界面适配多种屏幕尺寸和分辨率，保证在不同设备上的显示效果。



图 3-1 功能模块图

3.2 模块划分

(1) 主界面模块 (MainActivity)

处理网页加载、搜索和 PDF 转换的主要逻辑。

包含输入框、搜索按钮、历史记录按钮、保存为 PDF 按钮等。

(2) 历史记录和下载管理模块 (HistoryActivity、PdfFilesActivity)

显示用户转换历史的列表视图。

提供删除历史记录的功能。

(4) 数据库管理模块 (DatabaseHelper)

负责操作 SQLite 数据库，存储和检索历史记录数据。

(5) 关于界面模块 (AboutActivity):

提供应用程序的相关信息。

3.3 业务流程图

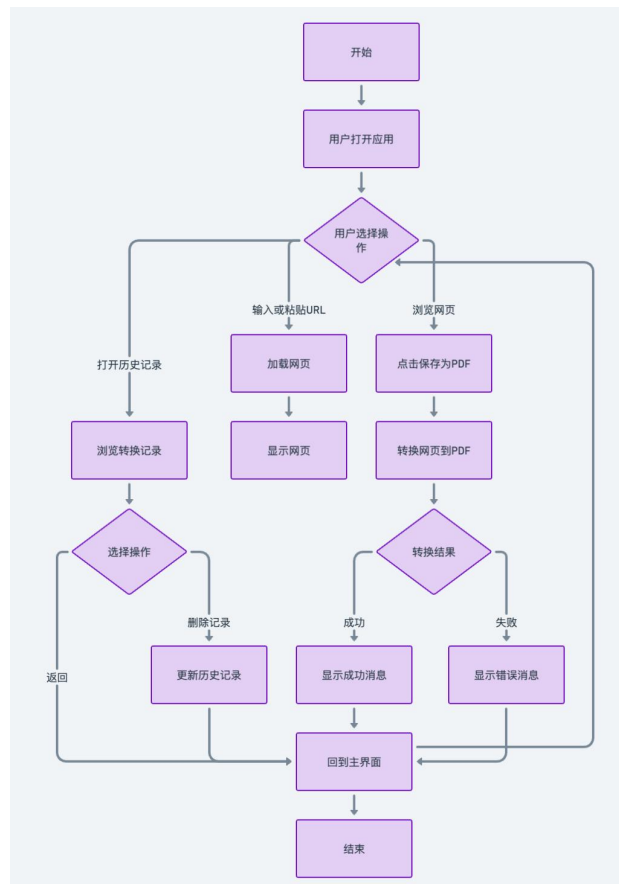


图 3-2 业务流程图

3.4 软件架构设计

(1) MVC 架构

Model: DatabaseHelper 处理数据存储和检索。

View: XML 布局文件定义用户界面。

Controller: Activity 类（如 MainActivity）控制应用逻辑。

(2) 技术栈

前端: Android XML 布局，用于定义 UI 界面。

后端: Java 语言，实现业务逻辑和数据管理。

数据库: SQLite，本地持久化存储。

(3) 接口与服务

WebView: 用于网页加载和显示。

PrintDocumentAdapter 和 PrintManager: 实现 Android 平台的打印功能，将网页内容转换为 PDF。

异常处理: 应用应妥善处理网络错误、打印失败等异常情况，确保用户体验。

4 软件开发与实现

4.1 技术和开发工具

平台	Android
编程语言	Java
开发环境	Android Studio
数据库	SQLite
主要库和组件	WebView
	PrintDocumentAdapter
	PrintManager
	SQLiteOpenHelper
	Snackbar
版本控制	Git

表 4-1 技术和开发工具

4.2 主要功能的编程实现方法

(1) 网页浏览和搜索

实现技术: 使用 WebView 组件加载和显示网页。

实现方法: 通过 WebViewClient 的 onPageStarted 和 onPageFinished 方法来监控网页加载状态, 并显示进度条^{[3][4]}。用户在输入框中输入 URL 后, 点击搜索按钮, 调用 performSearch 方法, 该方法将调用 webView.loadUrl(url)加载用户指定的 URL。

```
webView.setWebViewClient(new WebViewClient() {
    @Override
    public void onPageStarted(WebView view, String url, Bitmap favicon)
    {
        progressBar.setVisibility(View.VISIBLE);
    }
    @Override
    public void onPageFinished(WebView view, String url) {
        super.onPageFinished(view, url);
        printWeb = webView;
        progressBar.setVisibility(View.GONE);
    }
});
private void performSearch() {
    String inputUrl = userInput.getText().toString().trim();
    String url = formatUrl(inputUrl);
    webView.loadUrl(url);
    buttons.setVisibility(View.GONE);
    search.setVisibility(View.GONE);
    webView.setVisibility(View.VISIBLE);
    back.setVisibility(View.VISIBLE);
}
```

(2) 网页转 PDF 功能

实现技术: 利用 Android 的 PrintDocumentAdapter 和 PrintManager 实现网页的打印功能。

实现方法: 用户浏览网页并点击保存为 PDF 按钮后, 应用调用 PrintTheWebPage 方法。此方法首先检查 Android 版本, 然后创建 PrintDocumentAdapter 对象和发起打印任务。在 PrintDocumentAdapter 中处理网页到 PDF 的转换过程。

```
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
private void PrintTheWebPage(WebView webView) {
    printBtnPressed = true;
    PrintManager printManager = (PrintManager)
```

```

        this.getSystemService(Context.PRINT_SERVICE);
        String tempName = "[web2pdf]" + webView.getUrl();
        String jobName = tempName.replace("https://.", " ");
        PrintDocumentAdapter printAdapter =
webView.createPrintDocumentAdapter(jobName);
        assert printManager != null;
        printJob = printManager.print(jobName, printAdapter, new
        PrintAttributes.Builder().build());
        savePdfData(webView.getUrl());
    }

```

(3) 历史记录管理

实现技术：使用 SQLite 数据库进行数据存储。

实现方法：DatabaseHelper 类继承自 SQLiteOpenHelper，负责处理数据库相关操作。在每次用户完成 PDF 转换后，应用调用 savePdfData 方法，将转换记录插入到数据库中。历史记录的查询、显示和删除通过 DatabaseHelper 中的相应方法实现^[5]。

```

public List<PdfData> getAllPdfData() {
    List<PdfData> pdfDataList = new ArrayList<>();
    String selectQuery = "SELECT * FROM " + TABLE_NAME;
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);
    int idIndex = cursor.getColumnIndex(COLUMN_ID);
    int urlIndex = cursor.getColumnIndex(COLUMN_URL);
    int domainIndex = cursor.getColumnIndex(COLUMN_DOMAIN);
    int dateIndex = cursor.getColumnIndex(COLUMN_DATE);
    if (urlIndex != -1 && domainIndex != -1 && dateIndex != -1) {
        if (cursor.moveToFirst()) {
            do {
                PdfData pdfData = new PdfData();
                pdfData.setId(cursor.getInt(idIndex));
                String url = cursor.getString(urlIndex);
                if (url.length() > 120) {
                    url = url.substring(0, 120);
                }
                pdfData.setUrl(url);
                pdfData.setDomain(cursor.getString(domainIndex));
                pdfData.setDate(cursor.getString(dateIndex));
                pdfDataList.add(pdfData);
            } while (cursor.moveToNext());
        }
    } else {
        //...记录不存在...
    }
}

```

```

    }
    cursor.close();
    db.close();
    return pdfDataList;
}

```

(4) 用户界面交互

实现技术：使用 XML 布局文件定义用户界面。

实现方法：主界面 (MainActivity) 中包含 WebView、输入框、按钮等组件。使用事件监听器响应用户操作，如搜索、转换 PDF、查看历史记录等。

```

<WebView
    android:id="@+id/webViewMain"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/toolbar"
    android:visibility="gone" />
<Button
    android:id="@+id/savePdfBtn"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_alignParentBottom="true"
    android:background="@color/colorPrimary"
    android:text="@string/convert_webpage_to_pdf"
    android:textColor="#ffffff" />
<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:visibility="gone" />

```

(5) 异常处理和用户反馈

实现技术：对关键操作添加异常处理，并使用 Snackbar 提供反馈。

实现方法：在网络请求、数据库操作等关键部分添加 try-catch 块来处理异常。使用 Snackbar 向用户显示操作结果，如“转换成功”或“转换失败”。

```



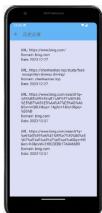
private String getDomainName(String url) {
    try {
        URL uri = new URL(url);
        String domain = uri.getHost();
        return domain.startsWith("www.") ? domain.substring(4) :

```

```
        domain;
    } catch (MalformedURLException e) {
        e.printStackTrace();
        return null;
    }
}
```

4.3 软件测试

软件测试的目的是确保应用程序在关键功能方面表现正常且稳定。包括网页加载、网页到 PDF 的转换、历史记录功能、删除历史记录以及进度指示和转换质量的测试。

测试项目	测试步骤	测试结果	测试记录
网页加载测试	打开应用，输入有效的网页 URL，观察网页是否成功加载。	应用能够正确加载和显示网页	
网页到 PDF 转换测试	在应用中加载一个网页，点击转换为 PDF 按钮，检查 PDF 文件是否生成	应用将加载的网页转换成 PDF	
历史记录功能测试	进行几次网页转 PDF 操作，然后打开历史记录，检查是否记录了所有操作	应用正确记录每次转换操作	

删除历史记录测试	在历史记录中选择一条或多条记录进行删除，确认记录是否被成功移除	应用能够删除指定的历史记录	
进度指示测试	加载一个网页和转换为 PDF，观察是否有进度条或其他进度指示出现	在网页加载和转换过程中有进度指示	
转换质量测试	转换多种网页，比较原网页和 PDF 文件的布局和内容	PDF 质量满足需求	

表 4-2 测试用例与记录

5 项目成果总结

5.1 主用户界面

图 5-1 展示了主用户界面，包含输入框、搜索按钮、历史记录按钮、保存为 PDF 按钮等。用户可以在输入框中输入网址或搜索内容，通过搜索按钮来加载和显示网页。



图 5-1 主用户界面

5.2 网页浏览界面

图 5-2 展示了应用程序使用 Webview 进行网页浏览的界面，用户可以在此界面中查看网页内容。界面底部提供了转换为 PDF 的按钮，便于用户将当前网页保存为 PDF 文件。



图 5-2 网页浏览界面

5.3 PDF 保存界面

图 5-3 展示了使用 PrintManager 和 PrintDocumentAdapter 调用打印服务的 PDF 保存界面，用户可以选择不同的打印选项，并将网页转换成 PDF 格式。

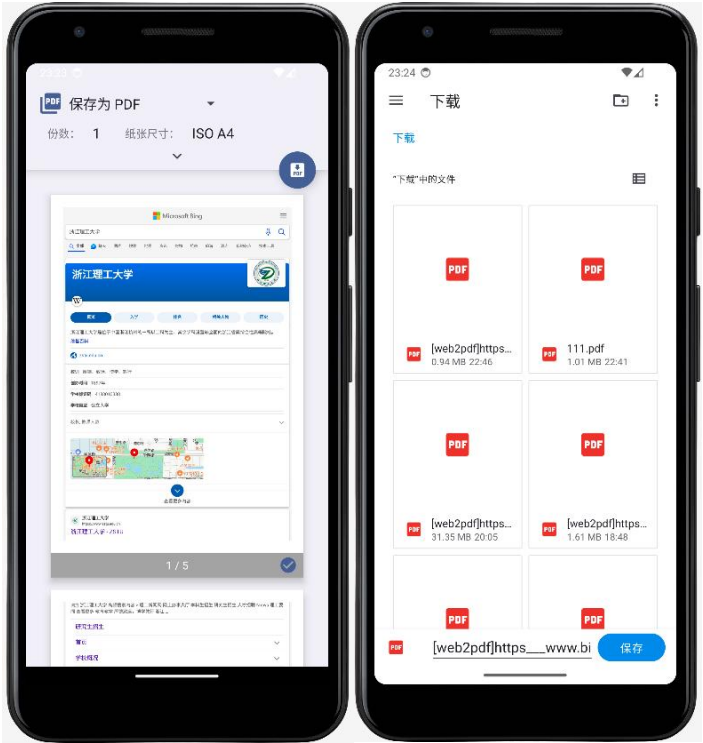


图 5-3 PDF 保存界面

5.4 历史记录和下载管理界面

图 5-4 展示了历史记录和下载管理界面，显示了用户之前浏览并保存为 PDF 的网页列表。每个条目包括已保存的 PDF 的 URL、域名、创建日期等信息。

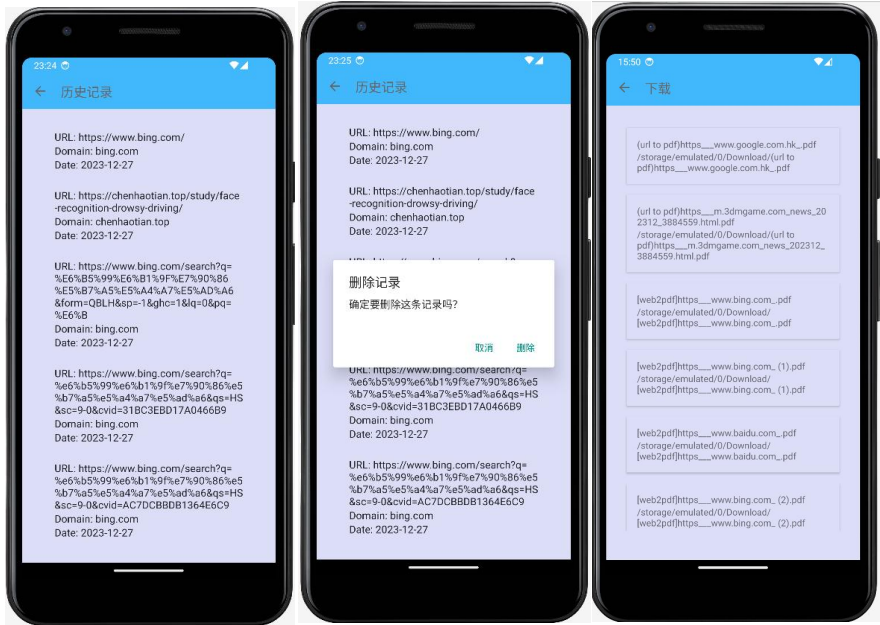


图 5-4 历史记录和下载管理界面

5.5 关于界面

图 5-5 提供应用程序的相关信息。



图 5-5 关于界面

6 参考文献

- [1] DiMarzio J. Beginning android programming with android studio[M]. John Wiley & Sons, 2016.
- [2] 殷涛, 崔佳冬. 基于 Android 软件开发组件化研究[J]. 计算机应用与软件, 2019, 36(9): 16-20.
- [3] 赵光泽, 李晖, 孟杨. Android 平台 WebView 组件安全及应用加固研究[J]. 信息网络安全, 2015 (10): 61-65.
- [4] Hazarika P, CP R R, Tolety S. Recommendations for webview based mobile applications on android[C]. IEEE, 2014: 1589-1592.
- [5] 马获蕾, 汤海凤. Android 系统中 SQLite 数据库的研究[J]. 电脑知识与技术: 学术版, 2013, 9(10): 6243-6245.