学号: 16430121

常州大学

大 作 业 报 告

课程名称：移动终端软件开发

题　　目：电子书阅读器

学生姓名：林锦雄

学　　院：信息数理学院　专 业 班 级：计算机 162

指导教师：高晋树

二〇一九年六月

课程大作业任务书

班级 计算机 162　　姓名 林锦雄

一、设计题目

电子书阅读器

二、设计背景

科技的发展改变了人们的生活方式，使人们的生活日趋方便和快捷。伴随着移动网络的日益发展和智能手机的日趋普及，基于智能手机的应用也逐渐增多，开发需求也猛增。

随着计算机技术的发展，电子书籍已经深入人们的生活学习中，手机上如果有一款阅读器，可以随时随身看电子书。

三、设计内容及要求

本课题主要设计一种基于 Android 平台的电子书阅读器，系统主要的设计目标和功能如下：

1、实现基本的电子书浏览功能，兼容多种格式；

2、能实现调节阅读字体大小，设置阅读标签，自动翻页等功能；

3、实现电子书下载，管理等功能；

4、具备良好的操作界面，操作简单方便，系统运行稳定。

四、进度安排

序号	工 作 内 容	时间安排（天）
1	任务安排和选题	1
2	前期准备	1
3	需求分析	2
4	系统设计	3
5	编码和调试	4
6	测试和完善	1
7	撰写报告	2
8	答辩并且验收报告	1

五、设计时间： 2019 年 3 月 27 日 ～ 2019 年 6 月 6 日

计算机科学与工程 **系**　　**指导教师** 高晋树

目录

1.前言

安卓系统已近成为当今主流的手机操作系统，可以为用户带来良好的移动互联网体验。现在越来越多人通过电子设备来阅读书籍，手机阅读软件应该给用户一个很自然的很人性化的操作体验，如类似如阅读纸质书籍的翻书效果，记录阅读书签的功能。本软件可以在 Android 智能手机上安装运行进行阅读 txt 电子书，可以提供目录检索、调整字体大小、调整阅读模式、调整亮度大小、记录书签等功能。

2.系统的需求分析

2.1 可行性分析

2.1.1 技术可行性分析

电子书阅读器主要体现在使用电子设备直观的展示出对电子书的可视化，所以使用 Android studio 建立 UI 界面，并选择其作为开发软件。

2.1.2 操作可行性分析

本系统的可行性主要通过几个属性来进行评定，分别是正确性、可靠性、健壮性、清晰性和兼容性。针对这些属性提出了如下的要求：

（1）正确性：功能的正常实现

（2）可靠性：可视化结果与实际内容一致

（3）健壮性：进行各种调试系统不奔溃

（4）清晰性：用户清晰的了解界面的功能

（5）兼容性：在不同的系统上都可运行

系统的目的是将复杂的操作可视化，所以采用图形用户界面，将功能操作直观的展示出来，通过一系列监听控件，用户能够通过简单的点击和滑动来完成对电子书的阅读，如同对实体书的操作。

2.2 具体需求分析

本课题主要设计一种基于 Android 平台的电子书阅读器，系统主要的设计目标和功能如下：

（1）基本电子书浏览功能；

（2）具有仿真翻页、覆盖翻页、直接翻页三种翻页风格；

（3）能保存阅读进度，支持目录和书签功能；

（4）支持文字大小设置，亮度调节，能根据光线自动切换夜间模式；

（5）常规、复古、护眼、夜间四种阅读主题；

（6）具备良好的操作界面，操作简单方便，系统运行稳定。

2.2.1 系统功能需求分析

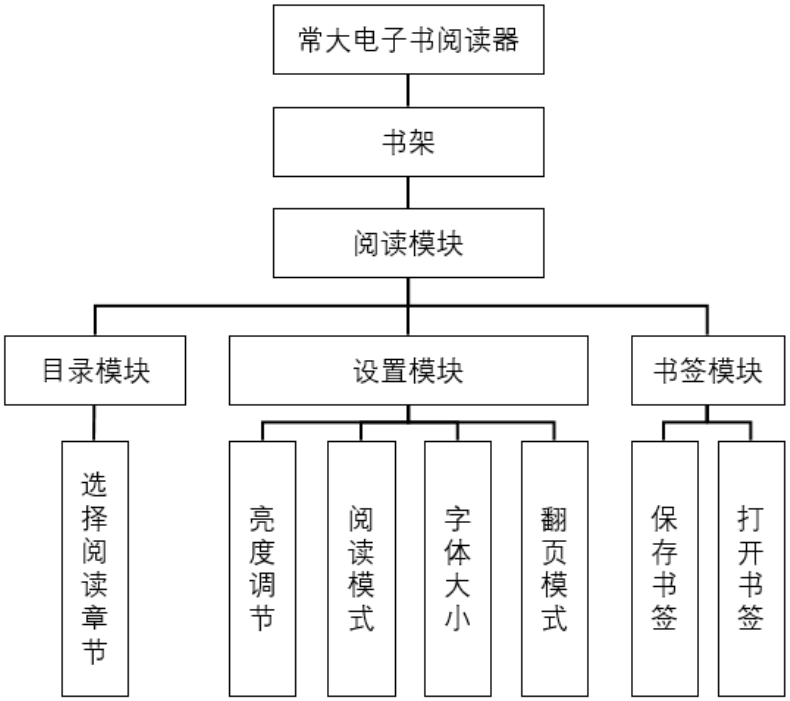

图 1 系统功能示意图

该电子书阅读器主要包含如图 1 所示的五个模块：

- 书架模块：模拟一个书架，就像书店里的书架一样，把书一本一本地排列在书架上，只显示每本书的封面，让阅读者通过点击对应的书的封面来开始阅读指定的书籍。
- 阅读模块：一个电子书最基本的功能就是阅读功能，在手机设备主界面上显示出书的文本内容。其文本内容应按照正常纸质书的文本格式，有良好的分段，清晰的章节、回、卷等名称。一个电子书还应该具有顶部显示当前阅读章节、卷、回等名称，以提示阅读者当前的阅读章节。底部显示当前阅读量占整本书的量的比率，让阅读者可以更好地分配阅读的速度以及知道当前的阅读进度。还应具有显示当前系统时间的功能，以随时提示阅读者当前时间，让阅读者可以更好的分配自己的时间。
- 目录模块：在目录模块中，会显示阅读者当前正在阅读的书籍的全部章节名称，包括章、回、卷、节。在显示出来的章节名称中，阅读者选择想阅读的章节，并点击它，阅读器就会把文本内容翻至对应的章节开始的地方。
- 设置模块：在设置模块中，包含四个小功能，从上到下分别是亮度调节功能、阅读主题功能、字体大小功能以及翻页模式功能。其中亮度调节功能：可以选择手动调节以及根据光线自动调节两种模式，如果选择了根据光线自动调节亮度模式，阅读器就会自动判断外界光线的变化，自动地调整屏幕亮度，而手动调节亮度模式就是用户可以根据自身需求通过亮度调节滚动条自由地选择屏幕指定亮度。阅读主题选择功能：共有

四种阅读主题供阅读者选择，分别是复古主题、常规主题、护眼主题和夜间主题。阅读者点击对应地阅读主题，主屏幕的内容背景色和字体色都会变换到对应的颜色，给阅读者提供更舒适的阅读体验。字体大小调节功能：阅读者可以根据自身的阅读需求以及对应的调节滚动条自由地调整阅读字体的大小。翻页模式调节功能：共有三种翻页模式供阅读者选择，分别是仿真翻页、覆盖翻页和无效果翻页。仿真翻页就是模仿实体书翻页的样子来进行翻页，覆盖翻页则是模仿滑走一整页的样子来进行翻页，而无效果翻页模式就是直接改变阅读页，阅读者不会看到任何变化就完成了翻页。

- 书签模块：书签功能有四种操作，分别是添加书签、显示书签、打开书签以及删除书签。单击工具条弹窗中的书签按钮就是把当前阅读的位置记录并添加为一个书签。长按书签按钮就会弹出新的窗口用于显示添加过的书签。在显示书签中，选择需要打开的书签，并点击它，阅读器就会把文本内容翻至书签对应的地方。在显示书签弹窗的右下角有一个删除书签的按钮，点击该按钮就会删除添加的全部书签。

3 .系统的总体设计

3.1 总体架构设计

3.1.1 界面设计

界面：电子书阅读器一共有 2 个 activity，分别为 ShelfActivity(书架页面)、ReadingActivity (阅读页面)。

弹窗：电子书阅读器拥有 4 个弹窗，分别为阅读页面的 BottomBar（底部工具栏弹窗）、ContentPopup（目录功能弹窗）、SettingPopup（设置功能弹窗）、（LabelPopup）书签功能弹窗。

UI：大量使用 Android 原生的组件，该设计规范已经具有足够高的可用性，无需自定义组件。

3.1.2 数据源

电子书阅读器的主要数据就是书的信息，有书名、书的封面这两个关键信息。考虑到显示转换的需要，又增添了格式化文本并将文本以段落为单位保存的属性、目录集合(卷/章/回/集等)属性、目录对应的在段落集合中的索引属性、缩进属性。具体如下代码所示。

```
1. public class Book {
2.
3.     private String mBookTitle;   //书名
4.     private Bitmap mBookCover;   //封面
5.     private List<String> mParagraphList;        //格式化文本,将文本以段落为单位保存
6.     private List<String> mBookContents;          //目录集合(卷/章/回/集等)
7.     private List<Integer> mContentParaIndexs;    //目录对应的在段落集合中的索引
8.     private String mSpace = "\t\t\t\t\t\t";      //首行缩进
9. }
```

3.1.3 业务逻辑

通过在各个 activity 里点击事件的使用，来启动相应的 activity.尤其对 recycleAdapter 的使用来绑定相应的数据在各个 activiy 里展示出来，实现了对电子书阅读器的各种功能的操作。

3.1.4 数据库

数据库采用 Android 原生支持的 SQLite 数据库。数据库主要用于保存添加的阅读书签。

3.2 框架选择

为实现上述的总体设计，相应的选择了下面几个主要框架和库：

'com.android.support:recyclerview-v7:28.0.0',为了使用更为方便的 recycleView 来显示各个 item 的数据，我们引入上述的依赖包，RecyclerView 是 support-v7 包中的新组件，是一个强大的滑动组件，与经典的 ListView 相比，同样拥有 item 回收复用的功能。RecyclerView 有四大组成：

Layout Manager：Item 的布局

Adapter：为 Item 提供数据

Item Decoration：Item 之间的 Divider

Item Animator：添加、删除 Item 动画

4.主要功能模块的详细设计和实现过程

4.1 书架模块

4.1.1 书架页面布局

书架模块的页面主要是 fragment_shelf_layout.xml 文件，并在其中内嵌 item_recycler_view_shelf.xml 文件。

其中，fragment_shelf_layout.xml 布局文件中仅有一个 RecyclerView 控件，它是一个用于大量数据展示的新控件，可以用来代替传统的 ListView，它更加强大和灵活。书架 fragment_shelf_layout.xml 布局文件如图 2 所示：

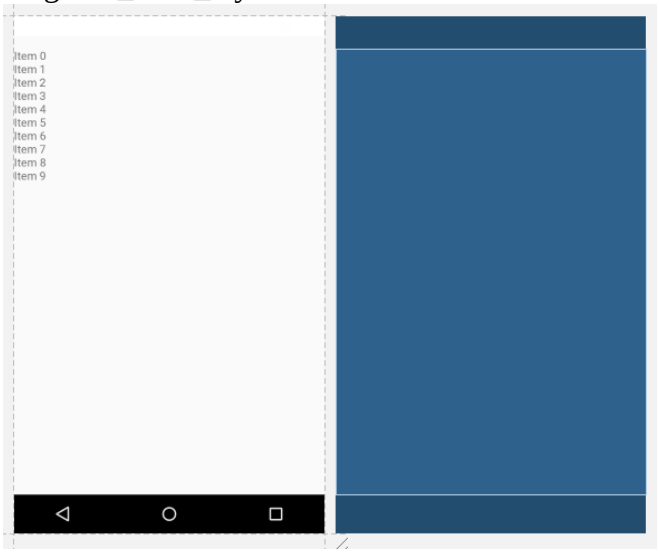

图 2 fragment_shelf_layout.xml 布局文件

item_recycler_view_shelf.xml 布局文件中有一个 ImageView，图像视图，直接继承自 View 类，它的主要功能是用于显示图片，实际上它不仅仅可以用来显示图片，任何 Drawable 对象都可以使用 ImageView 来显示。ImageView 可以适用于任何布局中，并且 Android 为其提供了缩放和着色的一些操作。在这里主要用于在书架上显示书的封面。书架 item_recycler_view_shelf.xml 布局文件如图 3 所示：

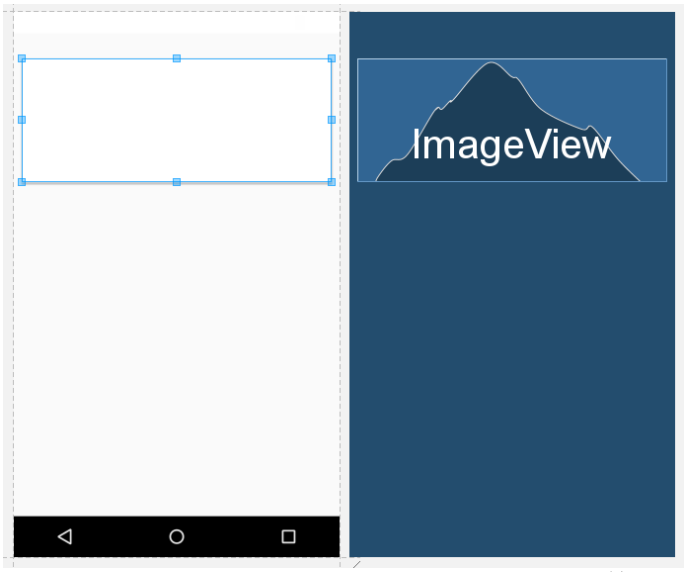

图 3 item_recycler_view_shelf.xml 布局文件

4.1.2 书架功能具体实现

在 ShelfActivity 中，其继承 SingleFragmentActivity 设置书架的基本信息，如屏幕显示、与 layout 布局文件关联等。然后通过继承 SingleFragmentActivity 中重写 createFragment 方法来创建 ShelfFragment，显示书架。

创建的 ShelfFragment，会关联到对应的 layout 布局文件中的容器和控件，然后会做一些页面的初始化操作，如书摆设的布局，书的监听器等。

这里书的摆设像书店里的书架一样，把书一本一本地排列在书架上，摆设布局设置为网格布局，一层最多放置三本书，书都以封面展示。阅读者点击对应的书的封面来开始阅读指定的书籍。

书架页面如图 4 所示。

图 4 书架页面

```
1. @Override
2. public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
3.     View v = inflater.inflate(R.layout.fragment_shelf_layout, container, false);//书架视图
4.     initEvents(v);
5.     return v;
6. }
7.
8. //初始化
9. private void initEvents(View v) {
10.     mContext = getActivity();
11.     mBookList = new BookLab(mContext).getBookList();
12.     RecyclerView recyclerView = (RecyclerView) v.findViewById(R.id.fragment_book_shelf_recycler_view);
13.     recyclerView.setLayoutManager(new  GridLayoutManager(mContext,3));//网格布局
14.     recyclerView.setAdapter(new  BookAdapter(mBookList));//设置 Adapter 适配器
15. }
16.
17. private class BookAdapter extends RecyclerView.Adapter<BookHolder> {
18.     private List<Book> bookList = new ArrayList<>();
19.     public BookAdapter(List<Book> bookList) {
20.         this.bookList = bookList;
21.     }
22.     @Override
23.     public BookHolder onCreateViewHolder(ViewGroup parent, int viewType) {
24.         LayoutInflater inflater = LayoutInflater.from(mContext);
```

```
25.          View view = inflater.inflate(R.layout.item_recycler_view_shelf, parent,
 false);
26.          return new BookHolder(view);
27.      }
28.      @Override
29.      public void onBindViewHolder(BookHolder holder, int position) {
30.          holder.bind(bookList.get(position));
31.      }
32.      @Override
33.      public int getItemCount() {
34.          return bookList.size();
35.      }
36.  }
37.
38.  private class BookHolder extends RecyclerView.ViewHolder implements View.OnClic
kListener {
39.      private ImageView mBookCover;
40.      private Book mBook;
41.      public BookHolder(View itemView) {
42.          super(itemView);
43.          mBookCover = (ImageView) itemView.findViewById(R.id.item_recycler_view_
image_view);
44.          itemView.setOnClickListener(this);
45.      }
46.      public void bind(Book book) {
47.          mBook = book;
48.          mBookCover.setImageBitmap(mBook.getBookCover());
49.      }
50.      @Override
51.      public void onClick(View v) {
52.          Intent intent = ReadingActivity.newIntent(mContext, mBookList.indexOf(m
Book));
53.          startActivity(intent);
54.      }
55.  }
```

4.2 阅读模块

4.2.1 阅读页面布局

阅读模块的页面主要是 fragment_reading_layout.xml 文件，并在其中内嵌 bottom_bar.xml 文件。

其中，fragment_reading_layout.xml 布局文件中有一个 FlipView 控件和包含 bottom_bar.xml 文件作为工具栏在屏幕底部。FlipView 是一个 Android 翻页效果的一个控件，效果类似于 Flipboard，它更加强大和灵活，是目前最好用的翻页效果控件。Bottom_bar.xml 布局文件中就是一个线性布局，三个一排排列的按钮控件，分别是目录、设置和书签按钮。

阅读页面 fragment_reading_layout.xml 布局文件以及 bottom_bar.xml 布局文件的效果图，如图 5 所示：

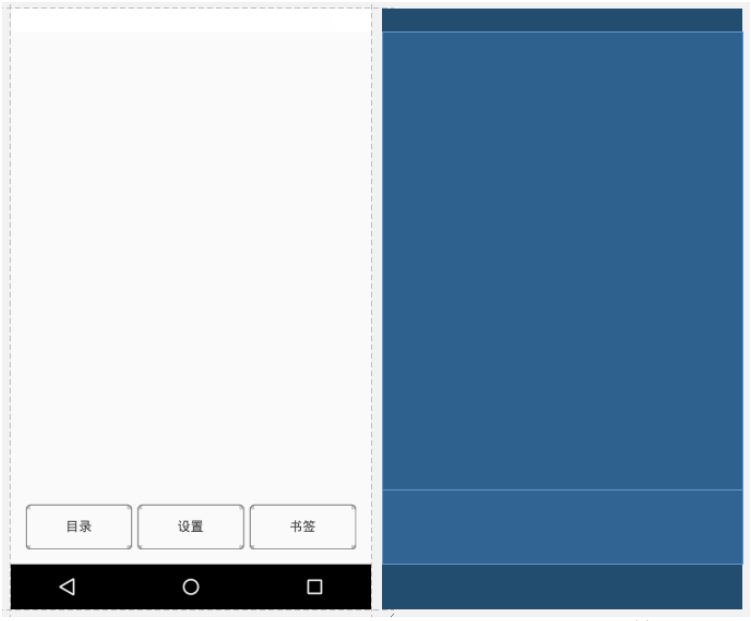

图 5 fragment_reading_layout.xml 布局文件

4.2.2 阅读功能具体实现

阅读主界面 ReadingActivity 继承 SingleFragmentActivity，依然是获得屏幕的基本属性，并初始化一些配置，如屏幕状态、关联到对应的视图页面的容器和控件等等。然后其会根据书架页面传过来的书的信息，找到对应的书籍，并显示书籍的内容供阅读者进行阅读。

阅读主界面如图 6 所示。

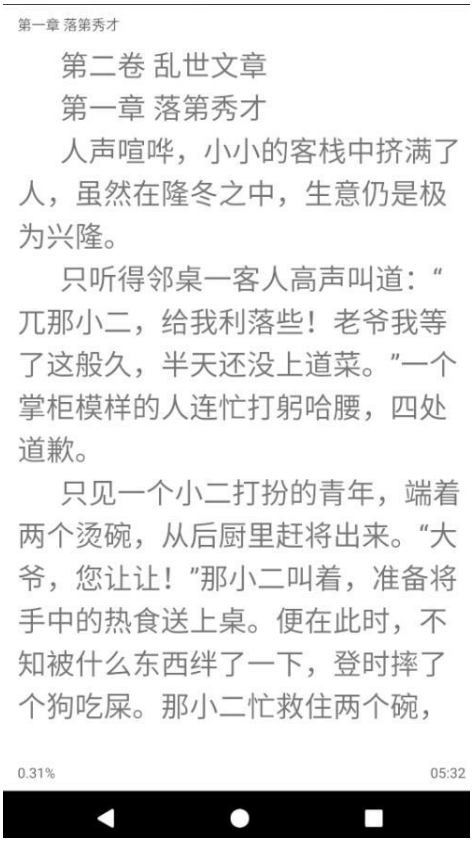

图 6 阅读页面

```
1. //阅读界面
2. public class ReadingActivity extends SingleFragmentActivity {
3.     public static final String EXTRA_BOOK_ID = "EXTRA_BOOK_ID";
4.
5.     public static Intent newIntent(Context context, int bookId) {
6.         Intent intent = new Intent(context, ReadingActivity.class);
7.         intent.putExtra(EXTRA_BOOK_ID,  bookId); //存储 bookId
8.         return intent;
9.     }
10.
11.     @Override
12.     protected void setScreen() {
13.         getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, Window
Manager.LayoutParams.FLAG_FULLSCREEN);    //全屏
14.     }
15.
16.     @Override
17.     protected Fragment createFragment() {
18.         int bookId = getIntent().getIntExtra(EXTRA_BOOK_ID,  0); //取 bookId，0 为
默认值
19.         return ReadingFragment.newInstance(bookId);
20.     }
21. }
```

Fragment 又称 Activity 片段，使用 Fragment 我们可以把屏幕划分成几块，然后进行分组，进行一个模块化的管理！从而可以更加方便的在 运行过程中动态地更新 Activity 的用户界面！另外 Fragment 并不能单独使用，他需要嵌套在 Activity 中使用，尽管他拥有自己的生命周期，但是还是会受到宿主 Activity 的生命周期的影响。

ReadingFragment 继承 Fragment。用于装配阅读主界面的基本信息，如背景设置、控件装载、书内容的文本装载等。主要就是显示书的内容，让阅读者可以像阅读实体书一样正常阅读，然后提供底部工具栏，供阅读者进行个性化阅读设置，以及书籍目录和阅读书签的使用。

```
1. //阅读界面的 fragment
2. public class ReadingFragment extends Fragment implements View.OnClickListener {
3.
4.     public static ReadingFragment newInstance(int bookId) {
5.         Bundle args = new Bundle();
6.         args.putInt(ARG_FLIP_BOOK_ID,  bookId);   //存储 bookId 到 Bundle
7.         ReadingFragment fragment = new ReadingFragment();
8.         fragment.setArguments(args);   //通过 fragment 传递参数，即 bookId
9.         return fragment;               //返回阅读界面
10.     }
11.
12.     @Override
13.     public void onCreate(Bundle savedInstanceState) {
14.         super.onCreate(savedInstanceState);
15.         initDatas();     //初始化数据
16.     }
17.
18.     private void initDatas() {
19.         mContext = getActivity();
20.         mBookId = getArguments().getInt(ARG_FLIP_BOOK_ID);
```

```
21.         mBook = new BookLab(mContext).getBookList().get(mBookId);
22.         mBookPageFactory = new BookPageFactory(mContext, mBookId);
23.         mBgColors = new int[]{
24.             0xffe7dcbe,  //复古
25.             0xffffffff,  // 常规
26.             0xffcbe1cf,  //护眼
27.             0xff333232  //夜间
28.         };
29.     }
30.
31.     //初始化视图
32.     private void initViews(View v) {
33.         mFlipView = (FlipView) v.findViewById(R.id.flip_view);
34.         mBottomBar = (LinearLayout) v.findViewById(R.id.bottom_bar_layout);
35.         mBottomBtns = new Button[]{
36.             (Button) v.findViewById(R.id.button_content),
37.             (Button) v.findViewById(R.id.button_setting),
38.             (Button) v.findViewById(R.id.button_label)
39.         };
40.         mContentPopup = new ContentPopup(mContext, mBook);
41.         mSettingPopup = new SettingPopup(mContext);
42.         mLabelPopup = new LabelPopup(mContext, mBookId);
43.     }
44. }
```

阅读界面底部提供工具条供阅读者进行阅读设置和使用。如图 7 所示。

目录功能会显示阅读者当前正在阅读的书籍的全部章节名称，包括章、回、卷、节。在显示出来的章节名称中。

设置功能，包含四个小功能，从上到下分别是亮度调节功能、阅读主题功能、字体大小功能以及翻页模式功能。

书签功能有四种操作，分别是添加书签、显示书签、打开书签以及删除书签。

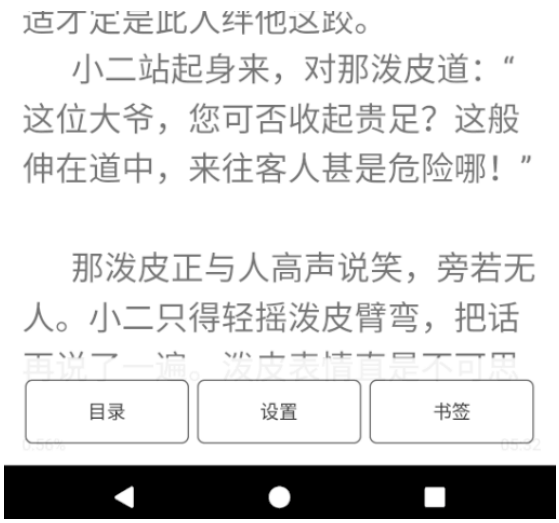

图 7 阅读界面底部工具条

5.系统测试及运行评价

5.1 功能测试

除了基本的阅读功能外，测试阅读的翻页功能、工具栏中的目录功能、设置功能以及书签功能。

翻页功能：在阅读页面，从右边往左边滑动，为翻到下一页，根据滑动的启动位置不同，翻页的样式也不同，这里演示的是仿真翻页样式，从右下角开始往左滑动。如图 8 所示。

目录功能：会显示阅读者当前正在阅读的书籍的全部章节名称，包括章、回、卷、节。在显示出来的章节名称中，阅读者选择想阅读的章节，并点击它，阅读器就会把文本内容翻至对应的章节开始的地方。如图 9 所示。

图 8 翻页功能

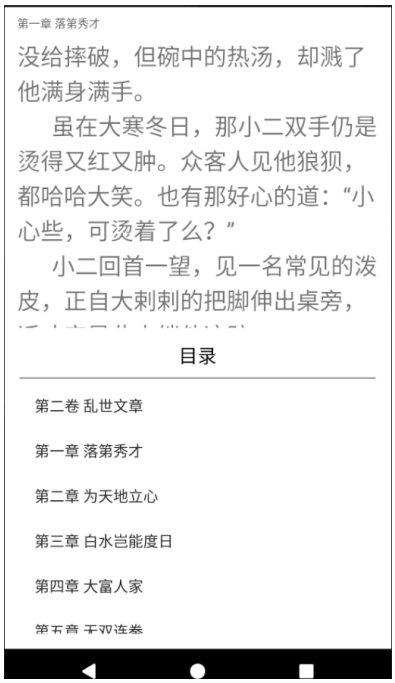

图 9 目录功能

设置功能：包含四个小功能，从上到下分别是亮度调节功能、阅读主题功能、字体大小功能以及翻页模式功能。其中亮度调节功能：可以选择手动调节以及根据光线自动调节两种模式，如果选择了根据光线自动调节亮度模式，阅读器就会自动判断外界光线的变化，自动地调整屏幕亮度，而手动调节亮度模式就是用户可以根据自身需求通过亮度调节滚动条自由地选择屏幕指定亮度。阅读主题选择功能：共有四种阅读主题供阅读者选择，分别是复古主题、常规主题、护眼主题和夜间主题。阅读者点击对应地阅读主题，主屏幕的内容背景色和字体色都会变换到对应的颜色，给阅读者提供更舒适的阅读体验。字体大小调节功能：阅读者可以根据自身的阅读需求以及对应的调节滚动条自由地调整阅读字体的大小。翻页模式调节功能：共有三种翻页模式供阅读者选择，分别是仿真翻页、覆盖翻页和无效果翻页。仿真翻页就是模仿实体书翻页的样子来进行翻页，覆盖翻页则是

模仿滑走一整页的样子来进行翻页，而无效果翻页模式就是直接改变阅读页，阅读者不会看到任何变化就完成了翻页。

书签功能：书签功能有四种操作，分别是添加书签、显示书签、打开书签以及删除书签。单击工具条弹窗中的书签按钮就是把当前阅读的位置记录并添加为一个书签。长按书签按钮就会弹出新的窗口用于显示添加过的书签。在显示书签中，选择需要打开的书签，并点击它，阅读器就会把文本内容翻至书签对应的地方。在显示书签弹窗的右下角有一个删除书签的按钮，点击该按钮就会删除添加的全部书签。

图 10 设置功能

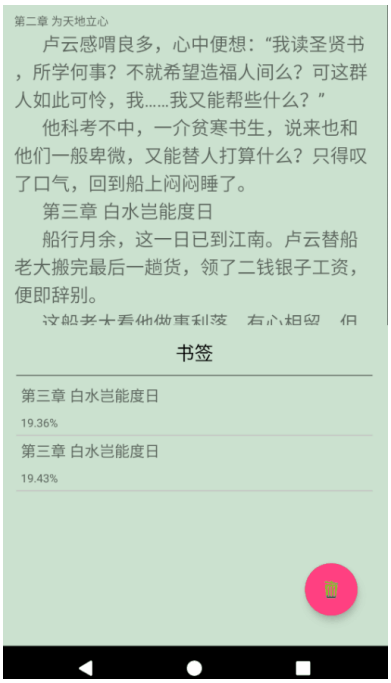

图 11 书签功能

5.2 运行评价

该系统实现了作为一个电子书阅读器该有的基本功能：基本电子书浏览功能、翻页功能、目录和书签功能、文字大小调节功能、阅读模式主题功能。该系统具有良好的操作界面，操作简单方便，系统运行稳定具有可实用性，符合生活实际。

6.总结与展望

本设计以 java 为开发语言，很好的解决了实际开发中遇到的技术问题。本设计的界面特点：界面友好、易于操作维护，能很好的完成阅读的全过程，使阅读者一进入软件就可以独立的进行自己需要的操作，一目了然。

经过测试，本系统设计的所有模块均可正常运行，并且达到了预期的效果。与此同时本设计还存在着很多的不足，如没能实现网络下载资源的功能。

后期希望为该电子书阅读器增添以上没有实现，但又是应该具有的功能。

参考文献

[1] 王向辉,张国印,沈洁.Android 应用程序开发.清华出版社. 2010

[2] 韩超,梁泉. Android 系统原理及开发要点详解.电子工业出版社. 2010

[3] 韩超，梁泉.Android 系统原理及开发要点详解[M].北京:电子工业出版社，2009

[4] 高焕堂,李立文. Android 应用软件架构设计[M].广悦出版社, 2009

[5] Android 开发中使用 SQLite 数据库 [EB/OL]

[6] 孙晓宇.Android 手机界面管理系统的设计与实现.北京邮电大学.2009

[7] 苏青山.移动时代 UI 设计中用户体验浅谈[J].华章,2011,(11): 3-5

[8] 倪天龙,张贤高,王培.数据库 SQLite 在嵌入式系统中的应用[J].单片机与嵌入式系统应用,2005,(10): 35-37

[9] 公磊,周聪.基于 Android 的移动终端应用程序开发与研究[J].计算机与现代化，2008(8):85~89

[10] Android 应用界面设计[R]. http://www.ifanr.com/65085

附录

（1） 个人负责内容情况描述

实现基本的电子书浏览功能：一个电子书最基本的功能就是阅读功能，在手机设备主界面上显示出书的文本内容。其文本内容应按照正常纸质书的文本格式，有良好的分段，清晰的章节、回、卷等名称。

一个电子书还应该具有顶部显示当前阅读章节、卷、回等名称，以提示阅读者当前的阅读章节。底部显示当前阅读量占整本书的量的比率，让阅读者可以更好地分配阅读的速度以及知道当前的阅读进度。还应具有显示当前系统时间的功能，以随时提示阅读者当前时间，让阅读者可以更好的分配自己的时间。

实现书架功能：模拟一个书架，就像书店里的书架一样，把书一本一本地排列在书架上，只显示每本书的封面，让阅读者通过点击对应的书的封面来开始阅读指定的书籍。

表 1 个人大作业情况表

个人代码量（行）	总代码量（行）	团队人数（人）	人均代码量（行）	团队贡献度（%）	团队协作度（%）	作业达成度（%）
557	2188	5	437	25.4%	9.1%	185.7%

（2）团队规范

① 驼峰式命名（大小写交替）

②所有文件编码格式为 UTF-8。

③ 一行代码不超过 100 字符。超过 100 字符，请换行或者提取变量。

④ Java 文件中不允许有多余的 import，不允许 importjava.io. *。

（3）心得体会

这次大作业让我知道了自己还有很多不足，很多地方都没有学透，在知识运用方面有待加强。希望可以通过更多的练习来锻炼自己的实际操作能力。