

AAMple-G Reader – PDF reading application

Aashish Shetye, Amaan Siddiqui, Manish Yadav, Gaurav Wankhede

Student, Dept. of Information Technology, Atharva College of Engineering

Abstract - In recent years, with the rapid development in technology and the popularity of mobile intelligent terminals (such as smart phones, tablets, etc.), mobile learning begins to rise and attracts widespread attention. Mobile learning is a new form of learning using wireless mobile communication network technology and wireless mobile communication devices (such as mobile phones, PDA, Pocket PC, etc.) to obtain education information, education resources and education services. The characteristics of mobile learning determine that it has unique advantages compared with traditional education, including the following characteristics: (1) Readers can read anytime and anywhere; (2) Readers can make full use of trivial time to learn; (3) It can meet readers personalized learning needs. Mobile phone is the most important carrier of mobile reading, currently the functions of reading apps in the platform is gradually becoming perfect, but they still have some shortcomings such as unable to display meaning of a word within the app itself, not tracking users reading progress and text-to-speech for given content. To cater these problems, we have set out to develop a reading app. It has inbuilt feature to display the meaning of a word and also provides various user-friendly functionalities which improve users' experience. The app is developed with appropriate system configuration and Android Studio framework.

Key Words: mobile intelligent terminals, shortcomings....

1. INTRODUCTION

Today's mobile phone users are often extensively dependent upon applications for accessing email, navigation, chatting, reading e Books etc.

All these purposes are fulfilled only when there is a framework, which provides a complete mobile platform to achieve these tasks. Android is one of the mobile application based platforms for providing a wide range of applications that are reliable.

An E-Book Reader is a software platform for reading books which can be installed in computer laptops and also on mobile devices. This software allows users to read books without internet, change view and navigate through the books easily.

This PDF Reader is the ultimate app for viewing and managing all your PDF documents on your Android device. Whether you're a student, a professional, or just an avid reader, our app offers a user-friendly interface that makes reading and editing PDFs a breeze.

With AAMple-G PDF Reader, you can quickly access all your PDF files from your device's internal storage. Our app supports a wide range of PDF features, including zooming, scrolling, and bookmarking. You can also use our app to fill out forms, sign documents, and even annotate PDFs with highlights, comments, and drawings.

Our app is designed with convenience in mind. With just a few taps, you can easily search for specific words or phrases in your PDF files. You can also customize the app's settings to suit your preferences, such as changing the font size, brightness, and color scheme.

1.1 Problem statement

The PDF Reader android application has been designed to provide a comprehensive solution for viewing, editing, and managing PDF files on Android devices. However, to meet the needs of users, the app needs to overcome several advanced challenges.

One of the significant advanced challenge is providing users with advanced PDF editing features, such as form filling, text editing, and document merging. The app can be enhanced by integrating advanced editing tools and collaborating with other software providers to provide a seamless user experience.

By addressing these advanced challenges, the PDF Reader app can provide a more advanced and comprehensive solution for managing PDF documents, enabling users to efficiently manage, edit, and secure their PDF files on their Android devices

Depending on the device, user must be able to read the e-book in low light, change its fonts, find word meanings and allow annotations.

An e-book must be able to use Text-to-speech software to read the text aloud for visually impaired, partially sighted, or just for convenience.

2. Proposed solution

To address the challenges faced by the PDF Reader android application, we propose the following solutions:

Compatibility and Performance: The app can improve its compatibility and performance by implementing advanced optimization techniques such as preloading, caching, and background loading. Additionally, the app can utilize modern technologies such as GPU acceleration and multi-threading to provide a faster and more responsive user experience.

Advanced Editing Features: The app can enhance its editing capabilities by integrating with third-party software providers or developing its advanced editing tools. This will enable users to edit PDF files, fill out forms, and merge documents more efficiently, providing a more comprehensive solution for managing PDF files.

Text to speech: The text to speech function in an Android app synthesizes speech from text for immediate playback or to create a sound file. It is a powerful tool that can enhance the user experience by converting written text into spoken words. This feature can benefit individuals with visual impairments, learning disabilities, or those who prefer to listen rather than read.

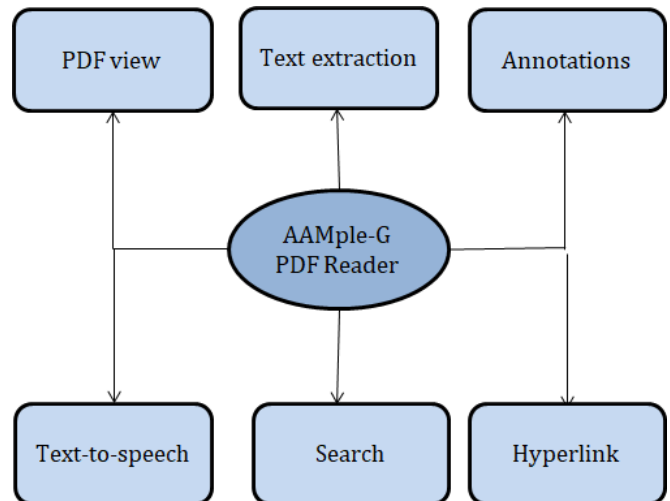
Hyperlink: Hyperlinks in an Android app are an essential feature that allows users to navigate between different screens or web pages within the app. Hyperlinks are clickable links that, when selected, can take the user to a new location within the app or to a website outside of the app. By implementing hyperlinks within an app, developers can make the app more user-friendly and engaging, improving its overall quality and usability.

User Interface: The app can improve its user interface by incorporating user feedback and conducting user experience testing. By simplifying the app's design and making it more intuitive, users will find it easier to navigate and use, improving their overall experience.

By implementing these proposed solutions, the PDF Reader android application can provide users with a more efficient, comprehensive, and secure solution for managing PDF documents on their Android devices, improving the app's functionality and user experience.

3. IMPLEMENTATION

The project is built in android studio using java as primary language.



3.1 Annotations

To implement annotations in a PDF reader in Android Studio, you can follow these steps:

Use a PDF library: There are several open-source PDF libraries available for Android, such as PDFBox, iText, and MuPDF. Choose a library that suits your needs and import it into your Android Studio project.

Create a PDF viewer: You can create a PDF viewer using a WebView, or you can use a library that provides a PDF viewer component. You can add the necessary UI elements, such as navigation controls and zoom controls, to the viewer.

Add annotation functionality: To add annotation functionality to the PDF viewer, you can use the annotation features provided by the PDF library. You can add the ability to highlight text, underline text, add notes, and draw shapes.

Store annotations: You need to store the annotations made by the user so that they can be retrieved later. You can use a local database, such as SQLite, to store the annotations. You can create a table to store the PDF file name, page number, annotation type, coordinates, and annotation text.

Retrieve annotations: When the user opens the PDF file again, you can retrieve the stored annotations from the database and display them on the PDF viewer. You can use the PDF library's annotation features to draw the annotations on the PDF pages.

Edit and delete annotations: You can allow the user to edit and delete annotations by selecting them and providing an editing interface. You can update the database with the changes made by the user.

By default each annotation type has an associated tool to allow you to create those annotations. When creating a custom annotation you'll need to create an associated annotation tool to define how the user can create that annotation.

Annotation tools handle interactions by receiving touch and gesture events. When touch & gesture are made, PDFViewCtrl sends corresponding event to ToolManager. ToolManager passes event to current tool which perform specified functions.

Table 1: Annotation Tools

Annot	Tool
Highlight	TextHighlightCreate
Line	RulerCreate
Underline	TextUnderlineCreate
widget	Signature

3.2 Text-To-Speech Feature

The Text-to-Speech function is a technology that allows a device to read out written text in a synthesized voice. It involves the user inputting text into a text field, selecting a language, and the text being analyzed by the TTS engine to identify the correct pronunciation, pitch, and emphasis of the words and phrases. To implement text to speech functionality in a PDF reader in Android Studio, you can follow these steps:

Use a PDF library: There are several open-source PDF libraries available for Android, such as PDFBox, iText, and MuPDF. Choose a library that suits your needs and import it into your Android Studio project.

Create a PDF viewer: You can create a PDF viewer using a WebView, or you can use a library that provides a PDF viewer component. You can add the necessary UI elements, such as navigation controls and zoom controls, to the viewer.

Extract text from PDF: To enable text to speech functionality, you need to extract the text from the PDF file. You can use the PDF library to extract the text from the PDF file and store it in a string.

Implement text to speech: You can use the Android TextToSpeech API to implement text to speech functionality. You can create a TextToSpeech object and set its language to the user's preferred language. You can then pass the

extracted text to the TextToSpeech object to convert it to speech.

Add text to speech controls: You can add UI elements to control the text to speech functionality, such as play, pause, stop, and speed controls. You can also add a progress bar to show the current position of the text being read.

Highlight the text being read: To improve the user experience, you can highlight the text being read by the text to speech engine. You can use the PDF library to identify the location of the text being read and highlight it on the PDF viewer.

3.3 Hyperlink

Hyperlink function allows users to navigate through digital documents by clicking on links that take them to other locations within the same document or to different documents or websites. To implement hyperlink functionality in a PDF reader in Android Studio, you can follow these steps:

Use a PDF library: There are several open-source PDF libraries available for Android, such as PDFBox, iText, and MuPDF. Choose a library that suits your needs and import it into your Android Studio project.

Create a PDF viewer: You can create a PDF viewer using a WebView, or you can use a library that provides a PDF viewer component. You can add the necessary UI elements, such as navigation controls and zoom controls, to the viewer.

Identify hyperlinks in PDF: You need to identify the hyperlinks in the PDF file. You can use the PDF library to extract the annotations from the PDF file and identify the hyperlinks.

Handle external hyperlinks: Some hyperlinks may lead to external websites. You can handle external hyperlinks by launching the default web browser of the user's device and opening the external link in it.

Add UI controls: You can add UI controls to enable the user to navigate through the hyperlinks in the PDF file. You can add a back button to go back to the previous hyperlink and a forward button to go forward to the next hyperlink.

This project includes adding links, images and audio file into the PDF.

