



Design and Development of a User-Friendly Android Reference Finder APP for Browsing e-content

¹N.Padmaja, ²M Bharathi, ³M. G. G. Sumanth Reddy, ⁴Varun Choda Naga Srinivasa

¹Professor, Department of ECE, Mohan Babu University, Erstwhile Sree Vidyanikethan Engineering College, A.P, Tirupati, India

²Asst.Professor, Mohan Babu University, Erstwhile Sree Vidyanikethan Engineering College, A.P, Tirupati, India

³Software Developer, Principal Global services, Bangalore, India

⁴Software Consultant, Daimler Trucks Innovation India, R&D, Bangalore, India

Abstract : This paper presents an Application (APP) designed and programmed using MIT APP Inventor. The prime objective of the app named “E-Content and Reference Finder Application” is to design a complete Android application which might refer the location of references supported subject & topic chosen by the user. The user can find and view videos, references and also the content he wants. We use the MIT App Inventor which incorporates a diversity of custom tools that facilitates us to develop and built mobile applications on the Android platform. The utmost important of these are the android Emulator and MIT Ai2 Companion plug-in. This APP is very useful to learners especially students who wish to learn fast.

IndexTerms - App design; MIT App Inventor; References; Emulator; Companion

I. INTRODUCTION

In the recent era, hand-held devices such as tabs, smart phones and many other user-friendly PDAs have become gradually powerful. Tremendous breakthroughs in the area of power optimization, and miniaturization models with additional features continuously opened the doors to a huge number of commercial applications. Most of the mobiles include variety of added features such as high resolution cameras (Front and back) , processors, using internet access at a very good speeds, etc., The mobiles and ipads have replaced PCs. As mobile devices become more like PCs, they will come to replace objects we tend to carry around such as check books, debit cards, calendars and planners, etc.

The immense use of mobile devices has enabled improving learning processes in an effective and easy way in educational environments also. The “ Android based Reference Finder APP for e-content” is one such application designed for the Google Android Phones which is very useful for fast learners. [1,16] . These applications can further be used for Augmented and Virtual reality applications which that helps the learners more attractive, interesting and easily understandable. Various techniques including signal and image processing, communication networks, data security and antennas , VLSI etc., related content can also be browsed [17, 18,19]

The MIT App Inventor was used for designing and building the functional modules of the mobile app. This app may be made accessible to all the users using the Google Play Store. App Inventor is one of the open-source module and can be easily downloaded and run-on most of the servers. It offers user-friendly code blocks which can be used in developing our own design modules. App Inventor is most widely used by educators and developers in app design. It provides users with two separate editors. The first editor is the Designer that is useful for arranging both on and off-screen components in the design module. The second editor is the Blocks Editor which can be used by the programmers to program the app's behaviour by arranging blocks together.

The paper is organized as follows. Section II presents Introduction to Android. MIT App Inventor info in Section III. Section IV presents the methodology of “E-Content and Reference finder” application. Section V concludes our work.

II. INTRODUCTION TO ANDROID

Android is one of the mobile OS- Operating System based on a improved version of Linux kernel and open source software that is primarily designed for smartphones and tabs which are touchscreen mobile devices. Android^[2] was developed by association of developers “Open Handset Alliance “ and commercially sponsored by Google. It was launched in September 2008. Most versions of the Android are proprietary. The core components from the Android^[3,4] Open Source Project (AOSP) are free and open-source software (FOSS) primarily licensed under the Apache License. When Android is installed on various devices, the ability to adapt to the other FOSS software is restricted by either not providing the corresponding source code or preventing reinstallation. Most of the Android devices liner with additional proprietary software preinstalled, notably like Google Mobile Services (GMS) and core apps [5] such as Google Chrome, associated Google Play Services development platforms.

The source code is used to develop variants of Android^[6] on a variety of other electronics, such as game consoles, portable media players, digital cameras, tablets, PCs, each with a specific user interface. Software packages [7] on the Android, which use the APK format, are generally distributed through registered application stores like Google Play Store, Samsung Galaxy Store, Huawei App Gallery, Amazon Appstore (including for Windows 11), Cafe Bazaar, etc.

III. MIT APP INVENTOR

MIT App Inventor [8,9] is an in-built, visual programming situation that allows everyone to design and build various fully functional apps for smartphones and tablets [10,11]. An introductory session of less than 30 minutes can help the MIT App Inventor to develop a simple first app. The block based tools enable the creation of complex design apps within very less time than routine and traditional programming environments. App Inventor enables the user to develop applications for Android phones using a web browser, and a connected phone or emulator. The App Inventor also enables to store your work in the servers and thus helps you keep track of your current projects. The MIT App Inventor user interface includes two main editors: the design editor and the blocks editor [12]

The Blocks Editor lets you program the app's behaviour by arranging blocks together and the Designer lets you create the app's interface.

Parts present in Designer are as follows:

Palette: To find your components and drag them to the viewer to add them to your app.

Designer Button: To Click from any tab to go to the Designer tab.

Viewer: To Drag components from the Palette to the Viewer to see what your app will look like.

Properties: To Select a Component in the Componentlist to change its properties here.

Parts present in Block Editor are shown below.

Built-In Drawers: To Find Blocks for general behaviors you may want to add to your app and drag them to the Block Viewer.

Viewer: To Drag blocks from the drawers to the blocks viewer to build relationships and behavior.

Component-Specific Drawers: To Find Blocks for behaviors and drag them to block viewer.

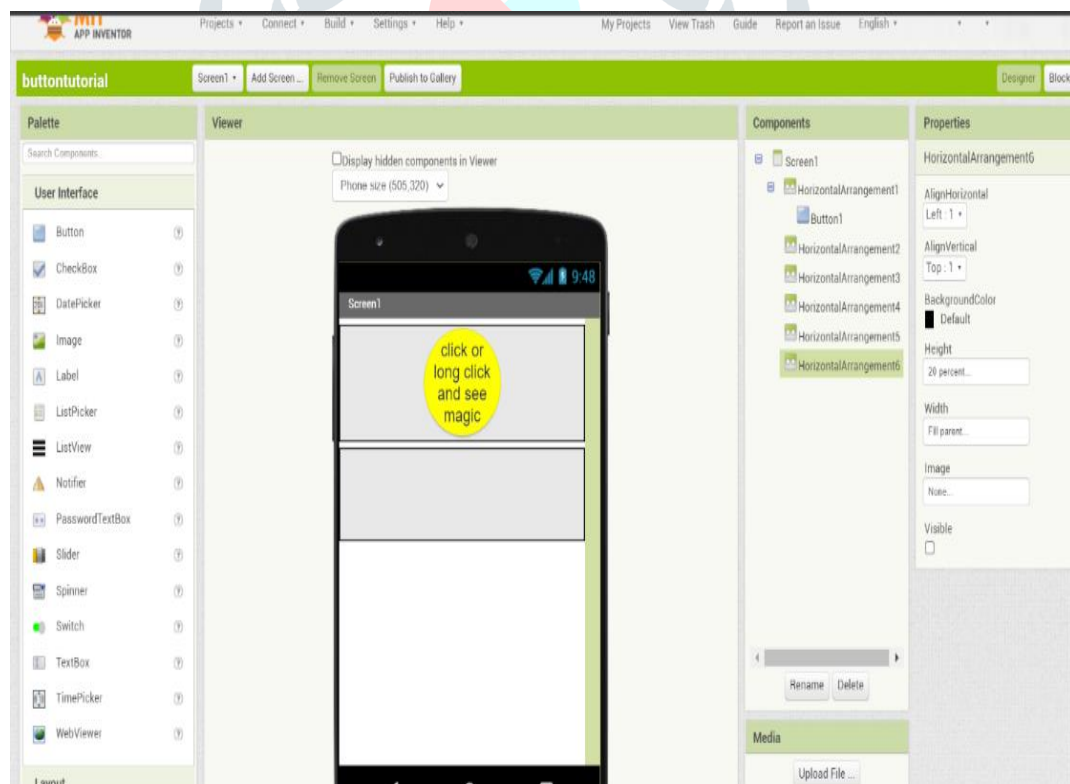


Figure 1: Design Editor in MIT App Inventor

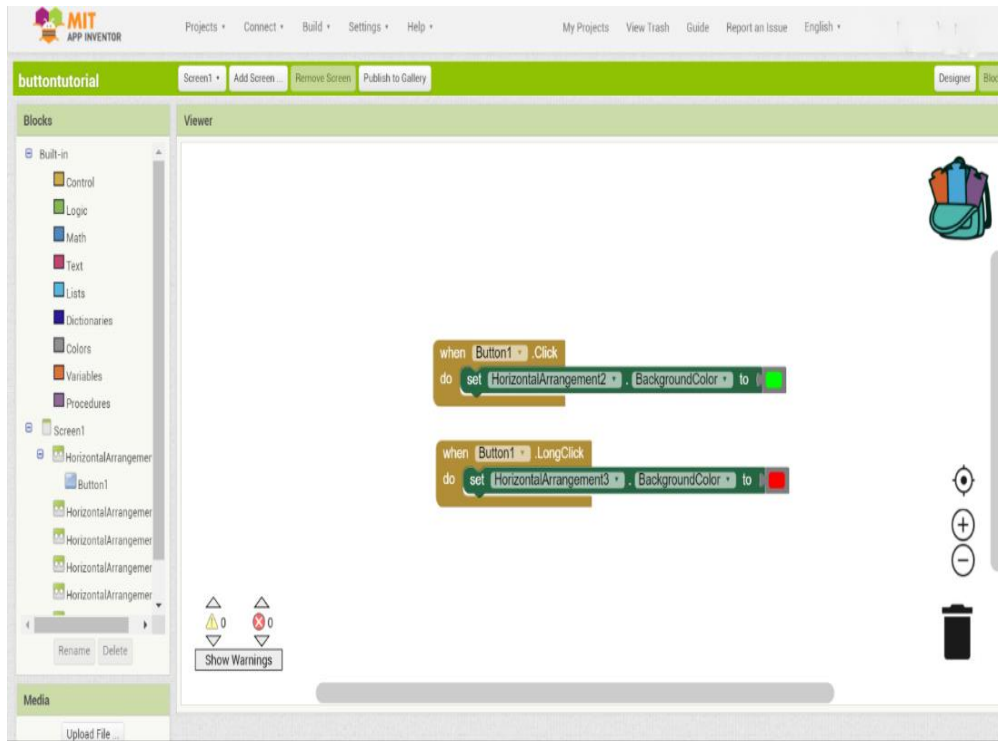


Figure 2: Block Editor in MIT App Inventor

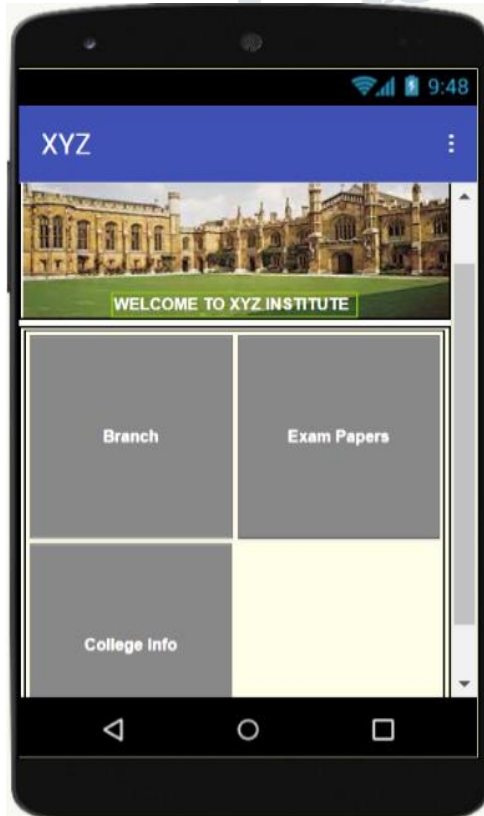


Figure 3. Home Screen



Figure 4. The Branch Screen

IV. METHODOLOGY: DESIGN CONCEPTS OF THE E-CONTENT AND REFERENCE FINDER

The main purpose of “E-Content and Reference Finder Application” is to refer the location of references based on subject & topic chosen by the user which helps in saving a lot of time and work [13]. The user can not only find video references but also, he/she can find the Content that required. This app is developed in Massachusetts Institute of Technology App Inventor Integrated Development Environment (IDE). Fig. 3 shows the home screen in the app where the user has to select which he has to choose [14, 15]. After selecting the branch, the app redirect to Fig. 4. Here we have to select the branch that contain the required topic to user. When the branch is selected, it will show the page where we can “Choose Subject”. The app will show the list of subjects related to the branch selected. Fig. 5 shows the list of subjects and once the subject is selected, further topics have to be selected. Fig. 6. Shows the screen that has the options to select the type of reference the user wants like a Video reference or Content reference. Once the type of reference selected, the app shows the list of topics related to the subject selected. Fig. 6 shows the list of topics the user can select. Once it is selected the user can see the details of the topic he required. Figs. 7-10 represent the background block codes in the application developed

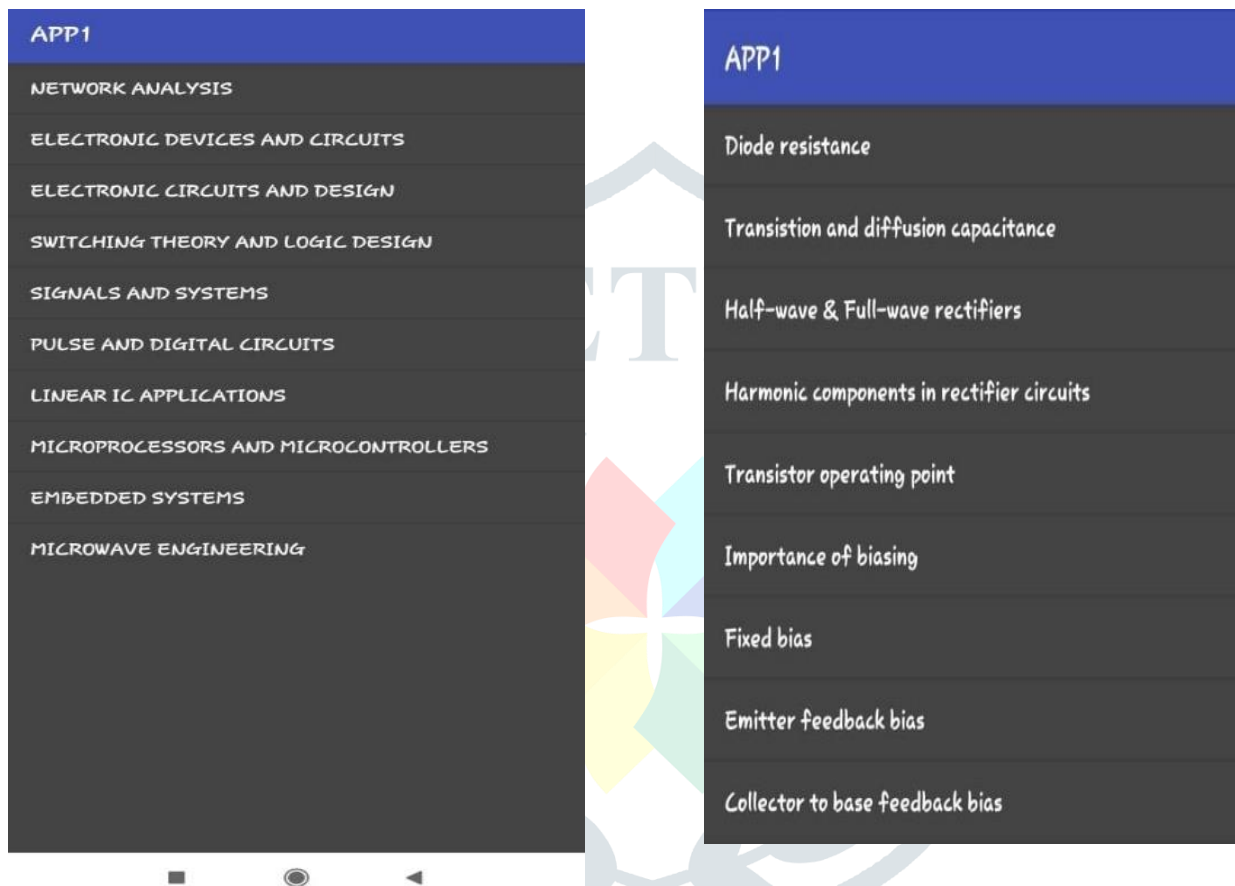


Figure 5. List of Subjects t and Topics to select



Figure 6. Type of Reference to Select

```

when Button1.Click
do open another screen screenName "Subject"

when Button2.Click
do open another screen screenName "Exam_Papers"

when Button3.Click
do open another screen screenName "About"

initialize global subject to make a list
NETWORK ANALYSIS
ELECTRONIC DEVICES AND CIRCUITS
ELECTRONIC CIRCUITS AND DESIGN
SWITCHING THEORY AND LOGIC DESIGN
SIGNALS AND SYSTEMS
PULSE AND DIGITAL CIRCUITS
LINEAR IC APPLICATIONS
MICROPROCESSORS AND MICROCONTROLLERS
EMBEDDED SYSTEMS
MICROWAVE ENGINEERING

when ListPicker1.BeforePicking
do set ListPicker1.Elements to get global subject

when ListPicker1.AfterPicking
do if ListPicker1.Selection = NETWORK ANALYSIS
then set HorizontalArrangement3.Visible to true
set HorizontalArrangement4.Visible to false
set HorizontalArrangement5.Visible to false
set HorizontalArrangement6.Visible to false
set HorizontalArrangement7.Visible to false
set HorizontalArrangement8.Visible to false
set HorizontalArrangement9.Visible to false
set HorizontalArrangement10.Visible to false
set HorizontalArrangement11.Visible to false
set HorizontalArrangement12.Visible to false
else if ListPicker1.Selection = ELECTRONIC DEVICES AND CIRCUITS

```

Figure 7. Background code for home screen and to select Subjects

```

initialize global ds to make a list
Linked List
Stacks & Queues
Trees & Heaps
Multway Trees & Graphs
Sorting & Hashing

when ListPicker8.BeforePicking
do set ListPicker8.Elements to get global ds

when ListPicker8.AfterPicking
do if ListPicker8.Selection = Linked List
then set ActivityStarter1.DataUri to https://youtu.be/R9PTBwOzceo
set ActivityStarter1.Action to android.intent.action.VIEW
call ActivityStarter1.StartActivity
else if ListPicker8.Selection = Stacks & Queues
then set ActivityStarter1.DataUri to https://youtu.be/j7q_LfV0hQ
set ActivityStarter1.Action to android.intent.action.VIEW
call ActivityStarter1.StartActivity
else if ListPicker8.Selection = Trees & Heaps
then set ActivityStarter1.DataUri to https://youtu.be/YAdLFsTG70w
set ActivityStarter1.Action to android.intent.action.VIEW
call ActivityStarter1.StartActivity

when Button1.Click
do open another screen screenName "ECE"

when Button2.Click
do open another screen screenName "EEE"

when Button3.Click
do open another screen screenName "EIE"

when Button4.Click
do open another screen screenName "CSE"

```

Figure 8. Background code to select required Topic and for selecting required Branch

V CONCLUSION

This paper presented a mobile app named E-Content and Reference Finder, which was designed and coded using MIT App Inventor. The purpose of the app is to provide the video and content references which saves a lot of time to the user and facilitates easy and interactive learning.

REFERENCES

- [1]. Developers A. What is Android[J]. 2011.
- [2]. Zhaojian M. Android-based Mobile Intelligent Application Development - The Development and Implementation of the Game Lianliankan[D]. Beijing University of Posts and Telecommunications, 2010.
- [3]. Is Android Really Open Source? And Does It Even Matter". *Make Use Of*. March 28, 2016.
- [4]. "Android - Google Mobile Services" *Android*. Retrieved October 21, 2018. *While the Android Open-Source Project (AOSP) provides common, device-level functionalities such as email and calling, GMS is not part of AOSP. GMS is only available through a license with Google [...] We ask GMS partners to pass a simple compatibility test and adhere to our compatibility requirements for their Android devices.*
- [5]. ZHANG S C. Development and Research of Application Based on Google Android [J][J]. *Computer Knowledge and Technology*, 2009, 28.
- [6]. <https://docs.microsoft.com/en-us/xamarin/android/app-fundamentals/android-api-levels>.
- [7]. http://www.tutorialspoint.com/android/android_architecture.html
- [8]. MIT APP INVENTOR <http://appinventor.mit.edu>
- [9]. S. C. Pokress, J.J.D. Veiga, "MIT App Inventor: Enabling Personal Mobile Computing," arXiv preprint arXiv:1310.2830, 2013.
- [10]. H. Zhu, H. Xiong, G. Y. Hui, E. Chen, "Mobile app recommendations with security and privacy awareness," 951-960, 2014.
- [11]. Dekhane and M. Tsoi, "Designing a Mobile Application for Conceptual Understanding", *International Journal of Mobile and Blended Learning*, vol. 4, no. 3, pp. 34-52, 2012.
- [12]. Wolber, David. (2011). App Inventor and Real-World Motivation.
- [13]. J. Zydney and Z. Warner, "Mobile apps for science learning: Review of research", *Computers & Education*, vol. 94, pp. 1-17, 2016.
- [14]. Padmaja, N., Kumar, B. S., Prasad, P. B., Irfan, D., & Srinivasa, V. C. N. DESIGN AND DEVELOPMENT OF AUGMENTED REALITY BASED LABORATORIES FOR SMART LEARNING.
- [15]. Prasad, P. Bhanu, et al. "Industry 4.0: Augmented and virtual reality in education." *Innovating with Augmented Reality*. Auerbach Publications, 2021. 29-52.
- [16]. Nimmagadda, Padmaja, Kondru Ayyappa Swamy, Samuda Prathima, Sushma Chintha, and Zachariah Callottu Alex. "Short-Term Uncleaned Signal to Noise Threshold Ratio Based End-to-End Time Domain Speech Enhancement in Digital Hearing Aids." *Indonesian Journal of Electrical Engineering and Computer Science* 27, no. 1 (July 1, 2022): 131
- [17]. C. Sushma and N. Padmaja, "Design of Hearing Aid using Variable Bandwidth Digital FIR Filterbank," 2022 International Conference on Breakthrough in Heuristics And Reciprocation of Advanced Technologies (BHARAT), Visakhapatnam, India, 2022, pp. 77-82, doi: 10.1109/BHARAT53139.2022.00026.
- [18]. Morasa, Balaji, and Padmaja Nimmagadda. "Low Power Residue Number System Using Lookup Table Decomposition and Finite State Machine Based Post Computation." *Indonesian Journal of Electrical Engineering and Computer Science* 6, no. 1 (2022): 127-134.