AN IDE FOR ANDROID MOBILE PHONES WITH EXTENDED FUNCTIONALITIES USING BEST DEVELOPING METHODOLOGIES

Sakila Banu[1] and Kanakasabapathi Vijayakumar[2]

[1]College of Computer Science and Information Technology, Taif University, Taif, Saudi Arabia
shakeelamjb@gmail.com
[2]Department of Mathematics,Anna University,Chennai,India.
saba.vkumar@gmail.com

ABSTRACT

Google's Android platform is a widely anticipated open source operating system for mobile phones. The mobile phone landscape changed with the introduction of smart phones running Android, a platform marketed by Google. Android phones are the first credible threat to the iPhone market. Google not only target the consumers of iPhone, it also aimed to win the hearts and minds of mobile application developers. As a Result, application developers are developing new software's everyday for Android Smart Phones and are competing with the previous in Market. But so far there is no Specific IDE developed to create mobile application easily by just Drag and Drop method to make even the non-programmers to develop application for the smart phones.

This paper presents an IDE with Extended Functionalities for Developing Mobile Applications for Android Mobile Phones using the Best developing Methodologies. The New IDE comes with the Extended Functionalities like Executing the created Application, Previewing the Application Created, Roll Back and Cancel Functions with the newly added Icons like Execute, Preview, Roll Back and Cancel Respectively. Another important feature of this paper is that the IDE is developed using the Best Developing Methodologies by presenting the possible methods for developing the IDE using JAVA SWING GUI Builder in Android ADT plug-in. The developed IDE is tested using the Android Runtime Emulator in Eclipse Framework.

KEYWORDS

IDE-Integrated Development Environment, GUI-Graphical User Interface, ADT-Android Development Tool.

1. INTRODUCTION

The Mobile Applications are classified into the following types. They are Communications, Games, Multimedia and productivity. Almost all developed applications come into this one category. Mobile devices are gaining acceptance for multimedia applications more as they are attracting the mobile users mostly. Based on the users Area of Interests Developers are providing applications for smart phones.

For example, if the user is more interested in learning Islamic topics, then the developers has to collect the useful pages from various Islamic sources based on the users interests and the pages have to be wrapped in one unit to present it as a Mobile Application for the smart phones. The developer has to write the program every time to present the useful WebPages as a single Mobile Application for the smart phone users.

DOI : 10.5121/ijcnc.2013.5411

Understanding the difficulties in performing the same task by writing program repeatedly, this paper presents an IDE by collecting the common tasks in the format of Drag and Drop Icons to make even non programmers to develop a mobile application easily for smart phones. The IDE has all common functions for wrapping WebPages into one application like selecting the WebPages as a PDF file, Slider View, Icon view with the additional added functionalities like Preview, Roll Back, Cancel and Execute Icons. Thus the useful WebPages can be wrapped into one single application as a multimedia entertainment application for the mobile users within minutes.

The IDE is first aimed to develop Multimedia applications for Android Mobile Phones as Android is an open source software stock for mobile devices and hence developers has the complete rights to create applications for Android smart phones. The IDE is mainly designed for the non-programmers who will develop mobile application without the knowledge of API and programming language. Hence applications created can be easily tested and installed on Android Mobile phones using Android Runtime Emulator that comes with the ADT plug-in in Eclipse Framework.

This paper is structured as follows. The proposed IDE comes with the possible methods of developing methodologies using Swing in Android Mobile phones. Android SDK does not provide a straightforward Swing Vocabulary for building the GUI system and the default GUI system provided by Android SDK is XML, hence creating a GUI using Swing in Android is possible only through two methods. This paper presents the two methods and the IDE is developed using the best method with added functionalities. The created IDE is tested with various applications and the results are shown below. The application thus created with this IDE is tested again with the Android Emulator in Eclipse. A table is provided at last to show the time consuming for different application using this IDE.

1.1. RELATED WORK

App Inventor for Android is an application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows anyone familiar with computer programming to create software applications for the Android operating system (OS). It uses a graphical interface, that allows users to drag-and-drop visual objects to create an application that can run on the Android system, which runs on many mobile devices.

Users can build many different types of apps with App Inventor. Often people begin by building games which draw funny pictures on faces. But app building is not limited to simple games. It also build apps that inform and educate and create a quiz app .

2. IDE DEVELOPING METHODOLOGIES

There are two possible methods to develop the IDE using Java Swing in Android. The First method is using an External GUI Builder in Swing and converting the Swing code to XML. The XML code is then installed to the Android Manifest file. The second method is the Direct Method in which the extra step of converting Swing to XML is omitted through Android Network Activity. Androids Network Activity has a way to build UI using Java Swing in Android without using XML.

2.1. Method I Using External GUI Builder

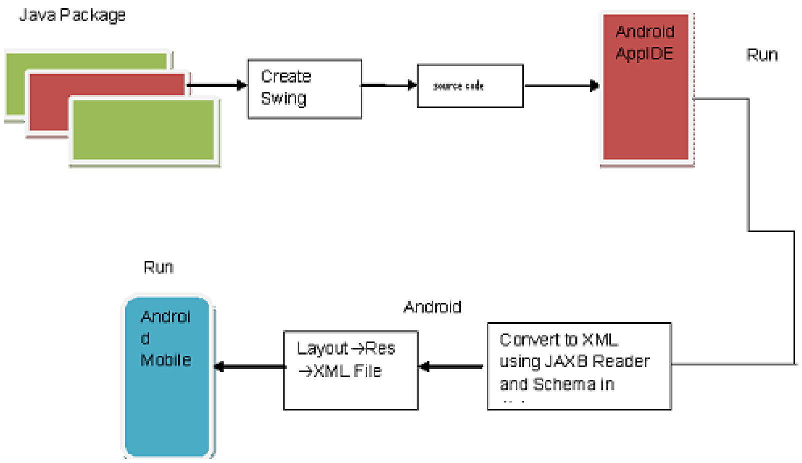
The tools used for this Method are the following:

1. Jigloo GUI Builder
2. JAXB Reader
3. Eclipse
4. Android Development Tools
5. Android Emulator to test the developed Application

The first method uses the External GUI Builder Jigloo. Jigloo is a highly customisable straightforward GUI Builder for Swing and SWT. The main advantage of Jigloo is that it can be integrated with Eclipse. As the IDE is created in Eclipse framework for Android Mobile Phone Jigloo is advisable for external Swing GUI Builder so that both ADT and Jigloo can be integrated into Eclipse framework. The IDE is designed using Swing and the code is converted to XML using the XML parser JAXB Reader.

The XML Code is then placed into the Android Manifest file and the application is created in Android. The created Application results with the .APK Files. The APK files are used to install into the Android Mobile Phone. The application is tested using the Android Emulator.

Method Structure



Steps to create the IDE using External GUI Builder

1. The IDE is developed as a package using JForm Designer as a plugin in eclipse.
   Package Name: org.developerworks.jIDE

2. A Swing Application is created in the Package
   Swing Application Name: NewIDE.java
3. The NewIDE.java contains the main source code required to build the IDE

Sample Source Code for NewIDE.java:

```
NewIDE.java
{
jButton1 = new JButton();
Sliderpanel.add (jButton1);
jButton1.setText ("slide1");
jButton1.setBounds (31, 26, 73, 92);
---------
---------
---------
}
```

4. The Code is copied to the JAXB Reader to convert to the XML as Android Manifest File Accepts only the xml code for GUI designing.
5. The XML code for the design of the IDE is copied and saved in a new file called AppIDE.xml
6. The AppIDE.xml is then copied to the Android Manifest File.
7. The application is tested using the Android Emulator.

Advantages of Method I:

1. Jigloo can be integrated to Eclipse as like Android SDK.
2. Both Swing and SWT applications can be created.
3. Using Drag and Drop method IDE is designed in Jigloo.

Disadvantages of Method I:

1. The Extra Round Step is not easier using JAXB XML parser.
2. JAXB parser doesn’t provide exact Xml conversion.
3. IDE sometimes results in error
4. The converted xml file is again placed to the Android Manifest file.
5. It increases time due to the extra Round XML Convertion.

2.2. Method II Using Android Network Activity:

The tools used for this Method are the following:

1. Eclipse
2. Android Development Tool
3. Android Emulator to test the developed Application

This is the Direct Method Where there is no need to create swing application for IDE in external software. The first method is used because android does not have swing application to develop GUI applications and only through XML GUI application can be created. Hence, the first method uses an external swing tool and converts the swing code to XML.

But, there is another way to create GUI in Android SDK tools using Swing Code but the main entrance should be the Android Activity Process. This main Activity class is called the Network Activity. It has the methods to create the UI screen that allows swing java code.

```
public class NetworkActivity extends Activity {
public EditText  Slide1;
public EditText  Slide2;
public void onCreate(Bundle icicle) {
  super.onCreate(icicle);

  // create the panel to enclose everything
  View mainPanel = _createInputForm();

  // show the panel on the screen
  setContentView(mainPanel);
}
...

}
```

The createInputForm in the Activity gives entrance to the Java swing code as given below.

Sample code:

```
private ViewGroup _createInputForm() {
  LinearLayout panel = new LinearLayout(this);
  panel.setLayoutParams(new LayoutParams(LayoutParams.WRAP_CONTENT,
LayoutParams.WRAP_CONTENT));
  panel.setOrientation(LinearLayout.VERTICAL);

  // Userid : label and text field
     TextView lblUserid = new TextView(this);
  lblUserid.setText("Userid");
```

Hence, the swing application is directly created in Android using Eclipse Framework and the IDE is tested for creating Mobile applications. The created application is then installed to the Android Emulator to test the application.

Advantages of Method I:

1. No need of External Swing GUI Builder
2. Time Consuming
3. Error is Reduced
4. No need of Convertion of External File
5. Provides Good Result as it is designed in the Android Framework.

Disadvantages of Method I:

1. Space is limited due to application installed in Mobile Device.

3. PROPOSED IDE USING METHOD II

The IDE is divided into the following. They are Widgets, Main View and Slider View. The Main View is the Heart of this IDE where the Dragged Icons and Slides are placed to this Main View. The slider view is shown in the Right Side of the IDE. The slides will be distributed in the Slider View once the file has been chosen from the File option. The Slides are dragged to the Main View by selecting the Slides from the Slider View and dragging it to the Main View.

The widget is shown in the left side of the IDE. The widget has the icons arranged vertically. The icons are dragged to the main view and are placed in the selected slides.

Once the icons are dropped into the slide, the input box is shown to enter the slide number based on the icon selected. Finally, the slides are consolidated and converted to .apk files to install into the android mobile phones.

The IDE is divided into three views.

I. **Icon View**
II. **Main View**
III. **Slider View**

Icon View:

The Icons like Previous, Next, Home and Scroll Bar are placed in the Icon View. These icons are programmed with Drag and Drop functionalities. The Icon is dragged to the main view to place on the selected slide. Once the icon is dropped on the slide in the Main View, input box will be shown to enter the Next or Previous Slide Number based on the Icon selected.

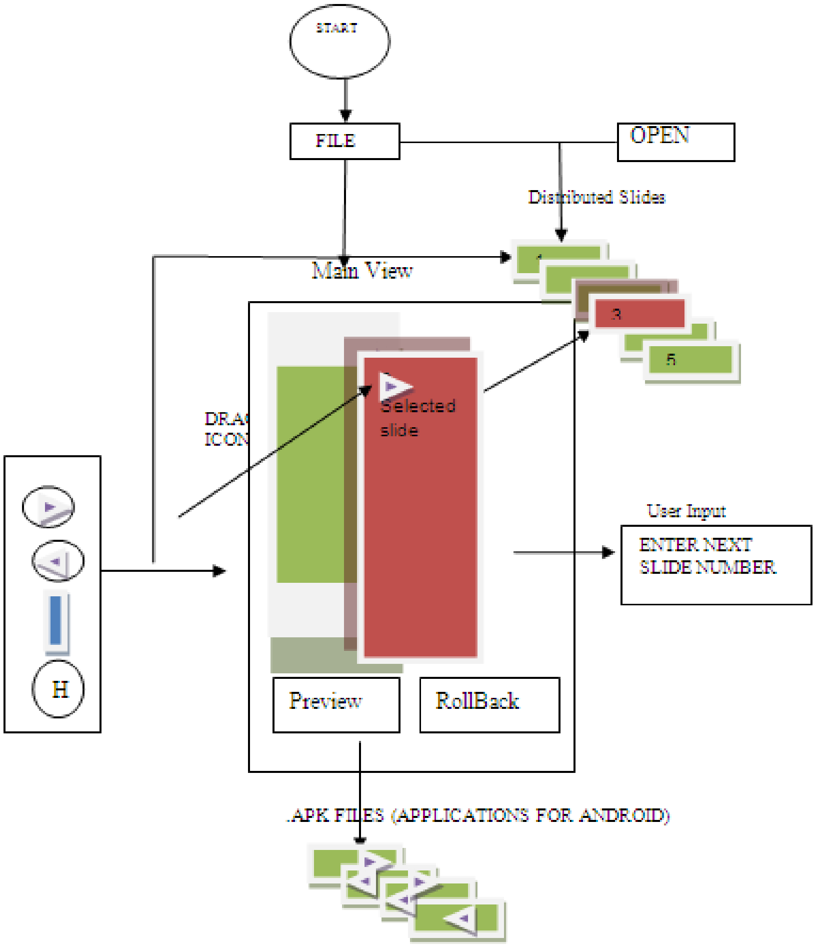
Main View:

The Main View is the heart of the IDE where the slides and Icons are Dragged and Dropped to this Main View. The slides are dragged to the Main View to place the icon on the Slide. The slides with the icons are then consolidated by the Slide Number which the user enters while placing the icons on the slide. The consolidated slides are then saved with the .apk file type and are send to the Android Emulator to test the developed application and installed to the Android Device.

Slider View:

The Slides are placed vertically one by one in the Slider View. Once the file is selected from the File Menu, the entire slides from the selected file are distributed to the Slider View. The slides are arranged vertically with the slide number.

Figure 1:
Architecture of the Proposed IDE:



3.1 Tools used to Develop IDE:

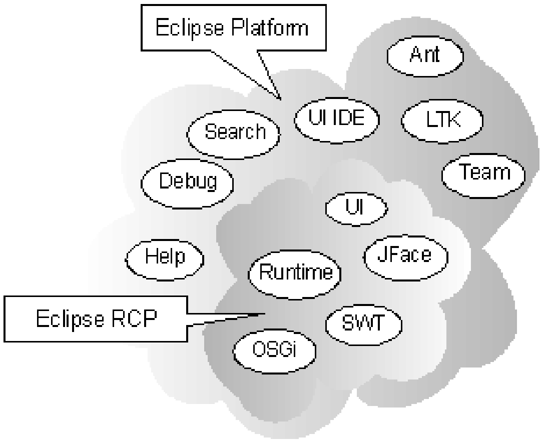
The tools used for this IDE are the following;

I. Eclipse
II. Android Development Tools.
III. Android Emulator to test the developed Application.

I.Eclipse:

Nowadays Eclipse IDE is used by all software developers to develop applications. Eclipse is easy to use and any number of plug-ins can be added to the Eclipse IDE. Moreover, eclipse is an open source IDE and includes a number of unique features such as code refactoring, automatic code updates. In its entirety, the Eclipse Platform contains the functionality required to build an IDE. However, the Eclipse Platform is itself a composition of components; by using a subset of these components, it is possible to build arbitrary applications. The Eclipse Rich Client Platform (RCP) is one such subset of components. Figure 2 shows a representation of some of the components in the Eclipse Platform and highlights the subset that makes up the RCP (in reality there are a great many more components).

Figure:2



II. Android Development Tools:

Android Development Tools (ADT) is a plug-in for the Eclipse IDE that is designed to give you a powerful, integrated environment to build Android applications. ADT extends the capabilities of Eclipse to let quickly set up new Android projects, create an application UI, add components based on the Android Framework API, and debugs the applications using the Android SDK tools, and even export signed (or unsigned) .apk files in order to distribute the application to the smart phones.

Developing in Eclipse with ADT is highly recommended and is the fastest way to get started. With the guided project setup, as well as tools integration, custom XML editors, and debug output pane, ADT gives an incredible boost in developing Android applications.

Developing applications for Android devices is facilitated by a group of tools that are provided with the SDK. Using the ADT (Android Development Tool) Eclipse plug-in these tools can be accessed efficiently.

III. Android Emulator:

Android is a software stack for mobile devices that includes an operating system, middleware and key Applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

Android emulator is a device that appears like Android phones, where the application developed can be tested right there without the need of external android device.

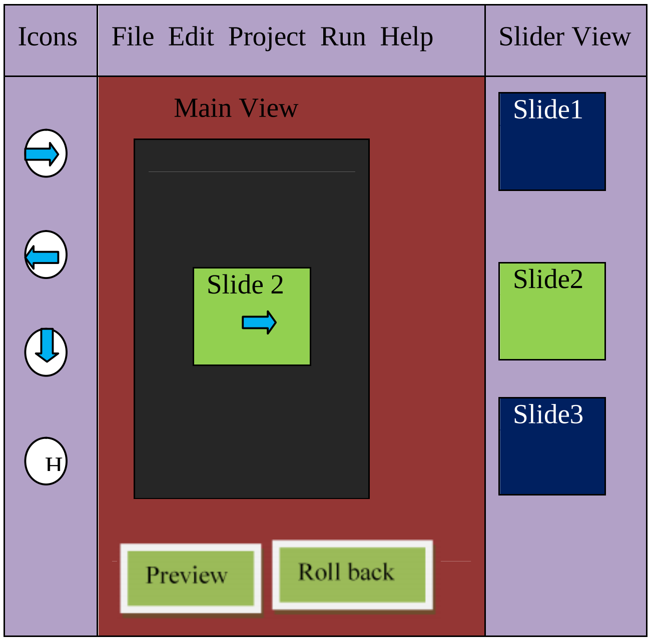
Android Application Framework:

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities. This same mechanism allows components to be replaced by the user.

3.2 Image of IDE:

Figure 3:



The left side panel shows the Icon View, the Middle panel shows the Main View and the Right Panel shows the Slider View. The Previous, Next, Scroll view and the Home Icons are arranged vertically in the Icon View. The Main View is the place where the selected slide and the Icons reside. The Slider View has the Slides that are arranged one by one vertically.

Extended Functionalities:

1. Preview
2. RollBack
3. Execute
4. Cancel

Preview:

The Preview Option is used to view the application created by the IDE before installing to the Android Emulator. This is useful to check the errors and the slides are correctly integrated. If the slides are not correctly integrated according to the users expected view then it is corrected using the RollBack Option.

RollBack:

The RollBack is an operation which returns the integrated slide to its previous state. This option is useful to rollback a particular slide in order to make changes. The RollBack and the Preview option are useful to check whether the application integrated is in the order of the users Expected Way.

Execute:

The Execute option is used to execute the integrated file before installing to the Android emulator. This option is found in the File Menu.

Cancel:

The Cancel option is used to cancel the whole operation and to return back to the File Menu. This Option is found in the File Menu.

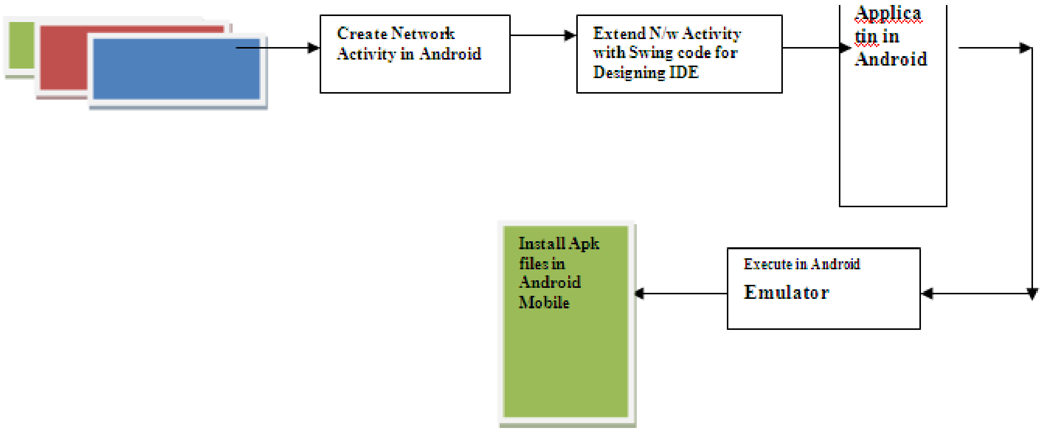
3.3 Program Structure:

The IDE is developed using Jigloo GUI Builder. Jigloo is an Eclipse Plugin. Jigloo can be installed using the eclipse powerful automatic updates.

The Structure of the Program is shown from the following figure

Figure 4:

Java Package


Create Network Activity in Android
Extend N/w Activity with Swing code for Designing IDE
Applica tin in Android
Execute in Android Emulator
Install Apk files in Android Mobile

Steps Followed in Creating the IDE:

- The IDE is developed as a package using Android Network Activity(Main Activity)in eclipse.
    - Package Name: org.developerworks.jiglooIDE

- A Swing code is created in the Package through Androids Network Activity.
    - Swing Application Name:NewIDE.java

The NewIDE.java contains the main source code required to build the IDE

Sample code
Using the Main Activity in Android

```
public class NetworkActivity extends Activity {
public EditText  Slide1;
public EditText  slide2;
public void onCreate(Bundle icicle) {
  super.onCreate(icicle);

  // create the panel to enclose everything
  View mainPanel = _createInputForm();

  // show the panel on the screen
  setContentView(mainPanel);
}
...

}
```

The createInputForm in the Activity gives entrance to the Java swing code as given below.

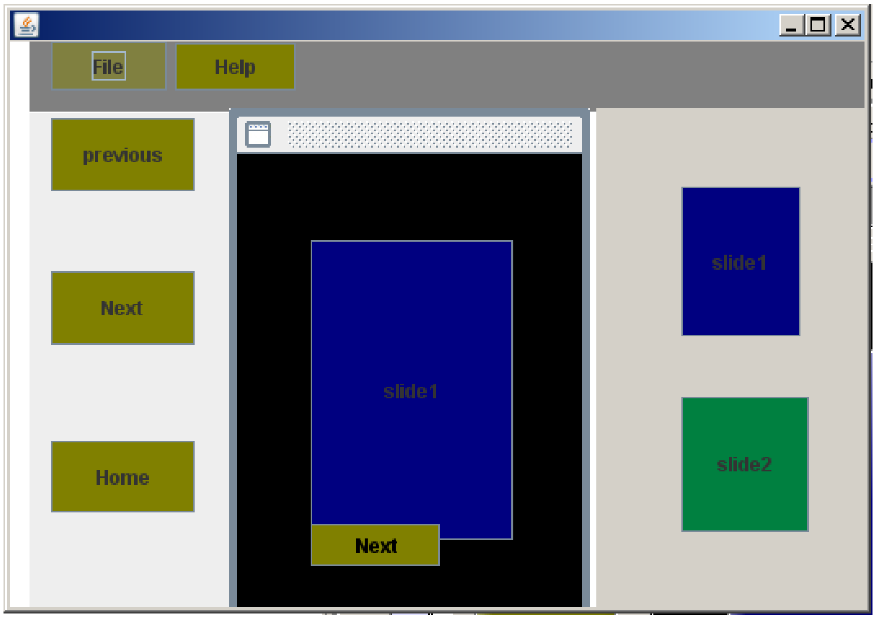
Sample code:

```
private ViewGroup _createInputForm() {
  LinearLayout panel = new LinearLayout(this);
  panel.setLayoutParams(new LayoutParams(LayoutParams.WRAP_CONTENT,
LayoutParams.WRAP_CONTENT));
  panel.setOrientation(LinearLayout.VERTICAL);

  // Userid : label and text field
     TextView lblUserid = new TextView(this);
  lblUserid.setText("Userid");
```

Result: Design of NewIDE.java using Method II

Figure:5

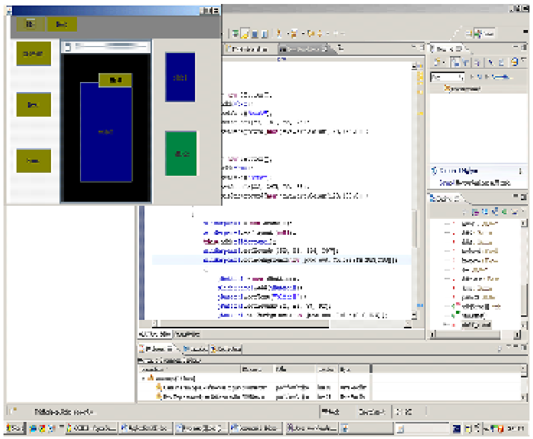


Eclipse With ADT Plugin(Android) to test the Application

- ADT Plug in is used with Eclipse to test the Application.
- A New Android Test Project is created. The created file in android activity AppIDE.xml is imported to the Android ->res->Layout Folder
- Applications are created using this ADT plug-in in Eclipse using the new Developed IDE AppIDE.xml file.
- Once the Application is created with ADT plug-in it will create the .APK files automatically.
- The Application .APK files are installed to the Android Mobile Phones.Thus Website Application for Android Mobile Devices is developed in few minutes and the Experimental Results are shown below.

Image Of the Created IDE in Jigloo

Figure 6:



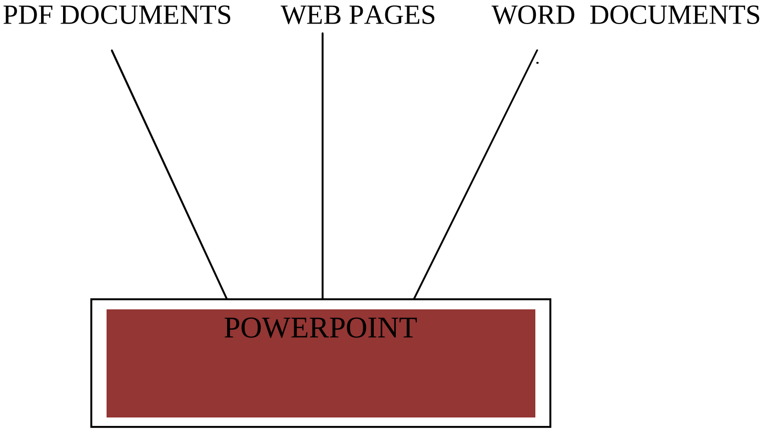
Experimental Setup and Results:

The IDE is first experimented for Islamic Sources. Various Islamic Sites are collected for Dhua, Historic Places and Prophets Sayings etc. Selected Pages from various sites are collected for different types of application like audio files, PowerPoint and PDF. Audio files are tested by collecting Dhuas from various sources and developed into mobile application as .apk files using the IDE .Applications to install into the Android Phones are experimented with the Created IDE and the Results are Shown Below.

Use Case 1: Development using PowerPoint

The applications that use similar file types are shown below.

FIGURE 8:



The above figure shows three different kinds of applications that fall into the same category. For Example, the PDF, the web pages and the word documents are treated here as a slides.
These applications will disperse the contents as the slides and the slides are consolidated using this IDE. The slides are then converted to .APK files to install into the smart phones.

The Android OS will accept only .APK files to install into the smart phones. For this the file type is converted to .APK extensions. This is done by the Android plug-in that is plugged into the eclipse IDE. The xml source code is then placed into the Android Plug-in into the Eclipse IDE to obtain the Target file Extension .APK.

Some Applications are tested with the developed IDE using both Method I and Method II and the time required to create the application using this IDE is described below in the Table.

TABLEI

Test cases:

Using Method I

Applications Tested				
Serial No	Type Of Applications	Type of Files	No of Pages	Time Taken
1	Dhua	Audio Files	10	5minutes
2	Historic Places	PowerPoint	10	4 minutes
3	Quran	PDF	30	10 minutes
4	Prophets Sayings	Audio Files	15	6minutes
5	Prophets Life	PowerPoint slide	20	6minutees

Time Taken for both the Method to Design the IDE

Serial No	Method Number	Time Taken
1	Method I	15Minutes
2	Method II	8Minutes

4. Conclusion and Future work:

This IDE is a useful tool for Developing Mobile applications easily and the main advantage of this IDE is even a non-programmer who is programming for first time can use this IDE to develop applications for mobile devices. The IDE is extended with the Useful Option for the user like Preview, RollBack, Execute and Cancel. Hence the New IDE ensures for accuracy for the application created before installing to the android Mobile Phone.

Future Work:

The IDE currently has limited icons that are necessary to develop application, and in future more icons will be added to develop advance applications for android smart phones. Currently the IDE is designed for Android Smart Phones and in future it will be extended to develop applications for all types of smart phones.

ACKNOWLEDGEMENTS

I would like to thank Everyone who supported for this Paper. Especially I would like to thank my Guide Dr.Nisar Hundewale who helped for this project and insisted to extend the Project by giving Ideas.

REFERENCES

[1] Understanding Android Security by Enck, W.; Ongtang, M.; McDaniel, P.; Pennsylvania State Univ., University Park, PA

[2] Android: Changing the Mobile Landscap by Margaret Butler from http://developerlife.com/tutorials/?p=289

[3] Android – How to build a service-enabled Android app – Part 1/3 UI Posted June 4th, 2008 by Nazmul

[4] Android-An Open Handset Alliance Project. http://code.google.com/android/.

[5] Bloom S.Book, M.Gruhn, V.Hrushchak, R.Kohler, A.(2008). Write Once Run Anywhere. A survey of Mobile Runtime Environments. Proceedings of the 3rd International Conference On Grid and Pervasive Computing(GPC2008):132-137

[6] Holzer, A.Ondrus,J.(2009). Trends in Mobile Application Development. Proceedings of the 2nd International Conference Mobile Wireless Middleware, Operating Systems and Applications(Mobile ware 2009):55-64

[7] http://www.vogella.com/articles/AndroidDragAndDrop/article.html

[8] http://javapapers.com/android/android-drag-and-drop/

[9] http://developer.android.com/guide/topics/ui/drag-drop.html

[10] http://en.wikipedia.org/wiki/App_Inventor_for_Android

[11] JForm Designer from http://www.formdev.com/jformdesigner/

[12] http://beta.appinventor.mit.edu/about/moreinfo/

[13] Jigloo GUI Builder ,http://www.ibm.com/developerworks/opensource/tutorials/

Corresponding Author

Sakila Banu Ali