

Efficiency in Motion: The New Era of E-Tickets

Fan Wu¹, Dwayne Clarke¹, Jian Jiang¹, Adontavius Turner¹, Asif Baba¹ and Sherman Buford²

¹Computer Science Department, ²Police Department
Tuskegee University
Tuskegee, AL 36088

Abstract—The development of mobile applications has played an important role in technology. Due to recent advances in technology, mobile applications are creating more attraction across the world. Mobile application is a very interesting field of research. There are several ongoing research and development in both industries along with in academia. In this paper, we present the design and implementation of a mobile application that creates an electronic ticket or e-ticket application for campus police. The goals for this mobile application are to make the ticket process much faster and easier for campus police by using mobile devices. Furthermore, the results will indicate an increase in performance and productivity for campus police by sending, retrieving data, printing tickets and permits for students to limit the need for paper based ticketing.

Keywords—General; Mobile Application; Mobile Device; E-Ticket

I. INTRODUCTION

Developing mobile applications have been a hot topic in the technology world. As the mobile application market continues to grow, so does the need to support a large number of platforms to ensure applications reach the widest possible audience to provide the maximum return on investment to application owners [1]. In this technological environment, mobile applications are creating more attraction across the world. One example is the Advanced Public Safety (APS).

Today, most patrol officers can issue a moving violation with a standard multi-part ticket form in approximately ten to fifteen minutes with an electronic ticketing resolution. Patrol Officers are able to issue a ticket in two to three minutes [2]. The understanding of concepts for mobile applications has increased considerably in the last few years, thanks to ongoing research and development in technology and the academic world.

Mobile application development has become available to the general public. Ticketing agencies such as Ticketmaster and Tickets.com for entertainment and live events, sport franchises such as Major League Baseball (MLB) in the United States, and the transportation industry (air, rail, and bus) are promoting and developing mobile ticketing applications, trials and services. In addition, there is a range of vendors that work closely with these organizations to make mobile ticketing happen. [3]

Creating a mobile application can be a challenging task and can take a very long time to master the application. The mobile application approach has been the basis for many implemented programs. A simple function such as creating a button from one Java class to another Java class can be

important to successfully achieve its goals. Finding patterns is an important issue and the center focus for mobile applications. Therefore, we can find a reasonable and intelligent solution to mobile applications. The complexity of global electronic ticketing solution is compounded when e-tickets are issued by different legacy based computerized reservation systems and e-tickets are stored in the database [4].

In this paper, we describe our application for designing and implementing an e-ticket based mobile application. The e-ticket mobile application described in this paper is Tigerticketz. It is designed to make the ticket process much faster and easier for campus police. Tigerticketz is a mobile application that creates an electronic report for traffic and parking regulations, and ticketing offenders. Some of the capabilities of the mobile application are to send, retrieve data, and print tickets along with permits to the students. The programming software used for the mobile application was the Android SDK with Eclipse. This software extends the capabilities of Eclipse to quickly develop new Android projects, create an application UI, add packages based on the Android Framework API, debug your applications using the Android SDK tools, and even export signed (or unsigned) .APK files in order to distribute your application [5].

The rest of the paper is organized as follows: Section 2 describes Tigerticketz Design. Section 3 defines System Architecture. Section 4 presents the Tigerticketz Algorithm. Section 5 is the Application Results and Section 6 concludes this paper with suggested recommendations for future research.

II. TIGERTICKETZ DESIGN

Tigerticketz Design is divided into several different layouts that interact with each other to create an application. In order to design and implement this application, some information about each of the layouts is required, which are presented in this section.

A. Main Layout

The main layout is a very important because it is the basic structure for the application. It can be considered essential for the application, where the other layouts are connected. The main layout includes textview for the Welcome text. It is an image view of a campus police logo, and four buttons that connect to four different Java classes, as shown in fig. 1. The Tigerticketz Design can be made for several different designs for a mobile application. However, it is the creator's discretion on the choice of the design.

The design of this layout has the responsibility to maintain the rest of the layouts. The other important thing to consider is

that the other layouts will add structure to the mobile application. Therefore, the creator has to decide how the mobile application's function determines the success of the application.

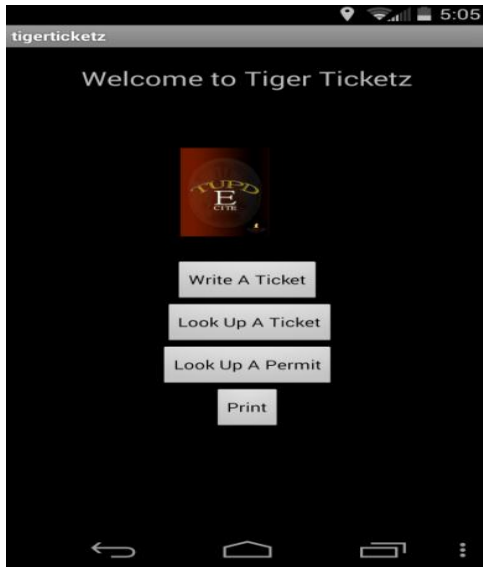


Fig. 1. Interface for Tigerticketz

B. Ticket Layout

The ticket layout is in charge of writing the tickets. In other words, it is the brains or impetus behind the application.

The ticket layout uses WebView that allow the user the ability to view web pages without opening a web browser and searching for the web page. This process makes it faster and easier for the user to access the web page with little changes as possible.

The design of this layout has the responsibility to give digital tickets for parking and permit violations. One important thing to consider is that the application must be able to connect to the Internet as well as Google Spreadsheets and Forms by authorized personnel to access the information.

C. Look up A Ticket

The lookticket layout helps the application by searching for parking and permit tickets. It is the right eye of the application's function of the application. Without this layout, it would be impossible for the application to search or retrieve tickets.

Similar to the ticket layout, lookticket layout also has the ability to view web pages. The design of this layout allows searching for tickets for parking and permit violations. Once the lookticket layout is activated by campus security, the officer can search tickets for past and present violations.

D. Look up A Permit

The permit layout helps the application by searching users that own permits. This is the left eye of the function of the application. It is similar in application to lookticket layout. Without this layout, it would be impossible for the application to search for a permit.

Similar to ticket and lookticket layout, the permit layout has the ability to view web pages. The design of this layout is in control of the look up permits to see if a user has a valid permit. Once the permit layout is activated by campus security, it can retrieve permits from the previous and current users.

E. Print

The print layout is in force behind the application. The design of this layout focuses on printout tickets for permit and ticket violations. Print layout uses ticket layout to print tickets. However, this can only be accomplished by connecting to the internet.

Similar to the rest of the layouts, the print layout uses a WebView that has the ability to view web pages. The WebView will send the user to a website that illustrates a step-by-step procedure of how to print from your tablet or smartphone device, as shown in fig. 2.

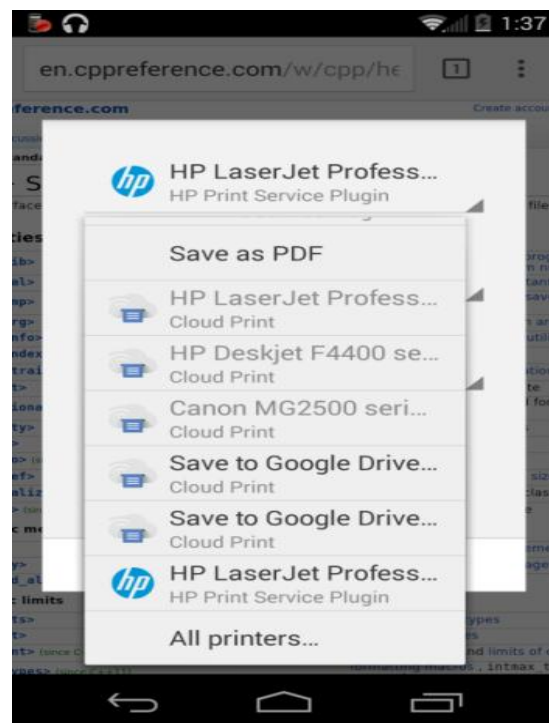


Fig. 2. Layout for Print.xml

III. SYSTEM ARCHITECTURE

A. Android Eclipse

It is an open source IDE (Integrated Development Environment) for Java projects and other programs. Theoretically, the open source is where the application software is crafted, and being supported through various stages of its lifecycle. Google officially supports it, and has created the Android Development Tools' plugin for Eclipse and integrated its AVD Manager virtual device management into the tool as well. [6] Moreover, a user can quickly build an Android base program, and test it through emulators that are the virtual devices that show users how the android's application runs on the android devices.

The Android offers users a plugin, which is called ADT (Android Development Tools). It is a very powerful plugin tool that integrates the environment for android development. It is an extension of Eclipse to quickly set up new android projects, create applications, and export signed and unsigned packaging. Before installing the ADT plugin, Eclipse must be installed in order to add the ADT. Some of the requirements for Eclipse to work are downloading Eclipse Indigo or greater, Eclipse JDT plugin, and JDK 6.

B. Programming Languages

Android Eclipse is written on Java code, and it is the main language for the Android, Android Eclipse and the use of other programming languages, including C, C++, Fortran, JavaScript, Perl, PHP, and Ruby. Development settings include Eclipse JDT, Eclipse CDT, and Eclipse PDT.

Java is a programming language and computing platform first released by Sun Microsystems in 1995. Several applications and websites will not work unless you have Java installed, and more applications are created every day. Java is fast, secure, and reliable from laptops to data centers, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere! [7]

The JRE (Java Runtime Environment) contains JVM (Java Virtual Machine) which is a Java stand and a support for Java programming. The JRE also allows the users to write Java code inside the web browser. The Java plugin software is not just a standalone program. It cannot be installed in isolation.

C. Google Drive

Google Drive is a way to transform your Google Docs account into an online and offline storage system. Google Drive is basically Google Docs for the Web. It can share direct information over the web by sharing documents, folders, and files. A user is able to retrieve the Google Docs portion and have the convenience of a virtual folder on the computer, which can be simply dragged to drop files to sync between laptops, tablets, and mobile phones. [8]

Some of the features can create documents, upload documents, image, video, application and convert Microsoft office documents to Google Drive. In our mobile application, we created a Google Spreadsheet and Form for TU Vehicle Registration and the T.U.P.D. Citation Entry Form.

D. Google Spreadsheet and Google Form

Google Spreadsheets is a web-based application that allows users to create, update and modify spreadsheets and share the data online in real time. The Ajax-based program is compatible with Microsoft Excel and CSV (Comma-Separated Values) files. Spreadsheets can also be saved as HTML [9].

Google Forms is a web-based application that allows users to create, update and modify forms and share the data live online. Moreover, Google Form can create online polls and surveys.

E. Mobile Device

Mobile device is any handheld computer device. It is designed to be portable, be able to fit in the hand or hands of a user. Some mobile devices are more powerful than other

devices. These mobile devices include smartphones, tablets, and watches. Mobile devices allow the user to do several things similar to desktop or laptop.

For this android application, it uses several mobile devices that were tested to make sure that the android application works properly. These devices include Samsung Galaxy Tab 2, Samsung Galaxy Tab 4, and Motorola Moto X smartphone.

1) Samsung Galaxy Tab 2

The Samsung Galaxy Tab 2 7.0 4G LTE (Version) is a 7-inch touchscreen which weighs about 12 ounces running on Android 4.0 or Android Ice Cream Sandwich operating system with a 1.2 GHz dual core processor. Ice Cream Sandwich operating system includes an Input Framework, TextureView, Switch Widget, and GridLayout. In addition, it accesses the internet and stream media over a wireless network which are needed to run the Tigerticketz app.

2) Samsung Galaxy Tab 4

The Samsung Galaxy Tab 4 is an 8-inch touchscreen which weighs about 11 ounces running on Android 4.4, KitKat, and offers 16 GB memory on the device that includes a microSD card that can go up to 64 GB memory which makes it easy to store more information. Kitkat includes updating your target API Level, Wireless and Connectivity, User Input, and Printing Framework that are very important to the success of the Tigerticketz app.

3) Motorola Moto X Smartphone

The Motorola Moto X Smartphone is a 4.7-inch touchscreen that weighs about 5 ounces running on Android 4.2.2 Jelly Bean operating system, and has the power of a 1.7 GHz dual core processor. Jelly Bean includes Lockscreen Widgets, Multiple Users, Nested Fragments, and Secondary Displays.

IV. TIGER TICKET ALGORITHM

Tigerticketz Algorithm is divided into multiple classes that interact with one other in order to create a functioning application. To implement this application, some information is needed from each class that is presented in this section.

A. Tiger Ticketz Activity

The Tiger Ticketz Activity is a class that connects to all of the other classes in the application. This is the most important class in the Tiger Ticketz application, because without this class it is physically impossible to connect to other classes in this application.

The Tiger Ticketz Activity includes an icon that represents a push of a button widget. A button can be pressed or clicked on, when the user touches the application button to perform an action or actions dependent on the event. In this application, buttons are used to connect to other classes in the application by using onClick and Intent.

OnClick is a method that is called when a view or event has been clicked. Intent is an abstract that sketch an operation that is going to be performed. For this application, the startActivity is used to launch an Activity from one class to another class. The Algorithm 1 labeled, "Tiger Ticketz", displays the code of the Tiger Ticketz Activity class presented in fig. 3.

```
        @Override
public void onCreate(Bundle savedInstanceState) {
    final Context context = this;
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    final Button
interb=(Button)findViewById(R.id.button1);
    final Button
ticketb=(Button)findViewById(R.id.button2);
    final Button
permitb=(Button)findViewById(R.id.button3);
    final Button
printb=(Button)findViewById(R.id.button4);
    //Ticket Button listener
interb.setOnClickListener(new OnClickListener()
    {
        public void onClick(View v)
        {
            //Start Ticket Activity
            Intent intent = new Intent(context,
            TicketActivity.class);
            startActivity(intent);
        }
    });
    //Ticket Search Button listener
ticketb.setOnClickListener(new OnClickListener()
    {
        public void onClick(View v)
        {
            //Start LookTicket Activity
            Intent intent = new Intent(context,
            LookTicketActivity.class);
            startActivity(intent);
        }
    });
    //PermitSearch Button listener
permitb.setOnClickListener(new OnClickListener()
    {
        public void onClick(View v)
        {
            //Start PermitSearch Activity
            Intent intent = new Intent(context,
            PermitSearchActivity.class);
            startActivity(intent);
        }
    });
    //Bluetooth Print Button listener
printb.setOnClickListener(new OnClickListener()
    {
        public void onClick(View v)
        {
            //Start LookTicket Activity
            Intent intent = new Intent(context,
            BluetoothPrint.class);
            startActivity(intent);
        }
    });
}
```

Fig. 3. Algorithm 1. Tiger Ticketz Activity Code

The Intent runs a facility for execution delay runtime between the different types of code for different types of application at once. The most important use is to launch an activity or activities, where it can pass data that can perform an action where Tiger Ticketz Activity's structure is based upon.

B. Ticket Activity

The Ticket Activity is a class that connects the classes with layouts. This class has the responsibility of digital tickets for parking and permit violation.

The Tiger Activity includes WebView, which we mentioned earlier, gives the user the ability to view web pages without opening a web browser and searching for the web page. The WebSettings' objective maintains the usage to control the basic settings for the WebView. The WebView runs no browser widgets, with the goal to display web content.

In addition, if the user needs an actual full web browser, then, the creator can invoke the browser application with URL intent. In this application, we used the URL intent by startActivity from the class to Google Spreadsheet, as shown in fig. 4.

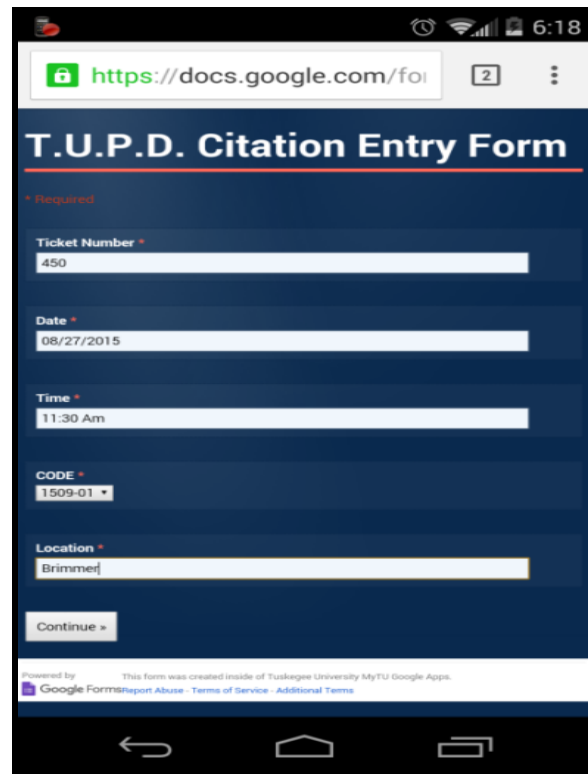


Fig. 4. Start activity from the class to Google Spreadsheet

C. Look Ticket Activity

The Look Ticket Activity is a class that connects classes with layouts. This class assists with searching for parking and permit tickets. The Look Ticket Activity includes Web View and setJavaScriptEnabled that enables or activates JavaScript. Another goal of WebView is to allow the user to display some web content without interacting with the web page outside of reading the content.

D. Permit Search Activity

The Permit Search Activity is a class similar to the rest connecting classes with layouts. This is a class that can retrieve permits for users with valid permits. The Permit Search Activity includes WebView and loadUrl that loads the given URL that the creator places in the class.

E. Print Tutorial Activity

The Print Tutorial Activity is the last class of this application. This is a class that shows the user how to setup the application on their mobile device to printout content. The Permit Search Activity includes WebView that will link to a site that shows the user a built-in printing function that can connect to a printer that allows the user to print from Google Drive by using the Google Chrome web browser.

V. EXPERIMENTAL RESULTS

We used an android mobile device to test our algorithm, and to make sure that the application works with all of the layouts and classes. We used three android mobile devices, which are Samsung Galaxy Tab 2, Samsung Galaxy Tab 4, and Motorola Moto X Smartphone to test all the functions of the Tigerticketz's application. This experiment was created to test our application, and to see if it is an usable and reliable tool for campus police. Many trials and tests were completed to figure out the best practices for writing tickets, retrieving tickets, searching permits, and printing tickets. This application is best suited to accomplish these tasks on newer android mobile devices where print functions can be easy used rather than creating code from scratch. Currently, the application can only work if an authorized user is connected to the internet to retrieve Google Spreadsheets and Forms to access the information. Moreover, users need to activate the Google Chrome web browser in order to print a ticket. The good news is that the code can be easily adjustable when a user is trying to make the layouts or even classes for Android Application. The experimental results reveal that our mobile application can be used by campus police to write tickets, retrieve tickets, search for a permit, and print tickets, as illustrated in fig. 5.

Fig. 5. Result of Print Function for Citation

VI. CONCLUSION AND FUTURE WORK

In this paper, we designed, created, and programmed a mobile application using a wide variety of classes and layouts to write tickets, retrieve tickets, search for permits, and print tickets. Initially, it was a bit of a struggle, but the application was successful. The application should not have any problems accomplishing these tasks with different types of android mobile devices with internet access and a printer. However, it may present future challenges in a real world scenario, since not all locations on campus have an internet connection or Wi-Fi. This suggests that users are not able to access the application. Moreover, the print function may not work on all android mobile devices, and the user will need to be near a printer with a Wi-Fi connect in order to print tickets in the application. Further research and application testing with Tigerticketz will involve improving the design by making the application more user-friendly, using a wireless portable printer, using Bluetooth instead of based printers, creating or changing the database from Google Drive, and implementing the application to give users access to the application with or without the internet.

VII. ACKNOWLEDGMENT

This work has been supported in part by US.NSF grant # DUE-1241670 and US Department of Homeland Security Scientific Leadership Award grant # 2012-ST-062-000055.

REFERENCES

- [1] Oliveiro-Priestnall, Darren. "mobileapps_whitepaper." www.redskiessoftware.com.2011. http://www.redskiessoftware.com/content/mobileapps_whitepaper.pdf.
- [2] Safety, AdvancedPublic. <http://www.aps.us>.2008. http://www.aps.us/landingpage/whitepaper/white_paper_download.shtml.
- [3] Wilcox, Howard. "MobileTicketingWhitepaper." <http://www.betterticketing.com>.2010. <http://www.betterticketing.com/articles/MobileTicketingWhitepaper.pdf>.
- [4] Sawant, LakshmiNarasimhan&Nitin. "tt-aa-wp-1." <http://www.hexaware.com>. 2006. <http://hexaware.com/casestudies/tt-aa-wp-1.pdf>.
- [5] Android.ADTPluginReleaseNotes.2015. <http://developer.android.com/tools/sdk/eclipse-adt.html>.
- [6] Electronicsweekly. Whatis...AndroidEclipse?2012. <http://www.electronicsweekly.com/blogs/eyes-on-android/whatis/android-eclipse-2012-01/>.
- [7] Oracle. What is Java technology and why do I need it? 2015. https://java.com/en/download/faq/whatis_java.xml.
- [8] Karch, Marziah. WhatIsGoogleDrive?2015. <http://google.about.com/od/d/a/What-Is-Google-Drive.htm>.
- [9] TechTarget. Google Spread sheetsde finition.2006. <http://searchsoa.techtarget.com/definition/Google-Spreadsheets>.