

An Android App for the use of Farmers in Agricultural Practices

Rama Kanta Choudhury¹, Shivani Shukla², Geet Sawhney³ and Chandra Kanta Samal⁴

^{1,2,3}M.A.I.T, New Delhi

⁴ANDC, HOD(CS), New Delhi University

E-mail: ¹rkchoudhury1@gmail.com, ²shukla.shivani2009@gmail.com,

³geet_sawhney14@yahoo.co.in, ⁴cksamal@gmail.com

Abstract—Since the use of cell phone is abruptly increased now-a-days we can update information and use it as a resource. It is a challenging task to improvise the usability of APP in the present day scenario. Though different researchers had proposed different approaches for the farmers to provide up to date information here we have try to update their day to day information and make them alert before they go for their work. Here we developed an APP to help the farmers to give latest and up to date information like the current weather forecast, information on best agriculture practices, trending news in the field of agriculture like selling /buying option based on current market demand and it also provides the minimum support price for the current demand. Here we used the Android PHP and MySQL and SQLite.

1. INTRODUCTION

App is a mobile operating system (OS) by Google, based on the Linux kernel which is used for the touch screen mobile devices such as smart phones and tablets. Android's user interface is user friendly based on direct approach to the human body using touch gestures that is usually related to the real and very fast in actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. The OS in Android tablets and smart phones is dominant. Hence Android forms the largest installed base of all operating systems of any kind It is popular with chip based technology companies like FPGA (Field programmable gate array) which is a chip technology that require a ready-made, less cost and flexibility to the user end and customizable operating system for high definition and high intensity devices. Its open source code has made a difference to the end user to make and develop new community-driven projects like online tracking system through cell phone and sending up to date information through cell phone to end user. Android has no centralized update system hence as security is concerned it is not up to the mark. In 2015 survey almost 90% of Android phones in use was known but unpatched security with lack of updates and support. The use of Android has created a smart phone war in field of electronic field.

2. THE USER FRIENDLY INTERFACE

Android's interface is logically based on direct gesture, using touch inputs which is flexible and closely connected to real-world day to day actions, like bidirection movement of the screen, smart activities using finger tip and reverse pinching to manipulate on-screen objects and use of virtual keyboard is more popular. In Emergency we can use Bluetooth or USB and fast extensible devices for further use. The fluid touch interface is more user friendly to the user. The vibration capabilities of the device provide more loneliness feedback to the user and hence makes more comfortable in the silence mode. Internal hardware and highly sophisticated sensors are used by some applications to respond to additional user actions. A screen is made in such a way that we can scroll several pages at a time. Home screen interface is customizable, allowing users to adjust as per their interest of look and feel comfortable with the devices to their tastes. Third-party apps are available on Google to manipulate our own Play and other app stores. We can re-theme the home screen, even mimic the look of other operating systems, such as Windows Phone. By updating Android devices most manufacturers use their own way to satisfy their customers with wireless update techniques so as to compete with their competitors.

3. WHAT WE USED HERE A HIGH A BRIF REVIEW

Applications ("apps"), which help to extend the functionality of devices is use in the form of app and is written using the Android software development kit (SDK). The Java programming language has complete access to the Android APIs. Java which is all in together with C/C++ , together with a choice of non-default runtimes that allow better C++ support. The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Initially, Google supported integrated development environment (IDE) using the Android

Development Tools (ADT) plug-in. In December 2014, Google released Android Studio, based on IntelliJ IDEA, as its primary IDE for Android application development. Other development tools are available, including a native development kit (NDK) for applications or extensions in C or C++, Google App Inventor, a visual environment for novice programmers, and various cross platform mobile web applications frameworks. In January 2014, Google unveiled a framework based on Apache Cordova for porting Chrome HTML 5 web applications to Android, wrapped in a native application shell.

4. MEMORY MANAGEMENT

Android devices are usually battery operated and hence the technology used here is CMOS technology where it manages processes to keep power consumption at a minimum level. It immediately suspends its operation and is available for immediate use which depends on customer satisfaction and can be adjustable and it does not use battery power or CPU resources. Android manages the applications stored in memory automatically when memory is low, the system will begin invisibly and automatically closing inactive processes, starting with those that have been inactive for longest. Life hacker reported in 2011 that third-party task killers were doing more harm than good.

5. HARDWARE

ARM (ARMv7 and ARMv8-A architectures) is the hardware required for Android, with x86. Later version of Android also officially supports MIPS architectures. In addition to the 32-bit variants, Android 5.0 "Lollipop" supports 64-bit variants of all platforms. Since 2012, Android devices including phones and tablets with Intel processors began to appear. While acquiring support for 64-bit platforms, Android was first implemented on 64-bit x86 and then on ARM64. For devices running Android 5.1, minimum amount of RAM required range from 512 MB(normal-density screens) to about 1.8 GB(high-density screens). Android 4.4 requires at least 512MB of RAM, 32-bit ARMv7, MIPS or x86 architecture processor (latter two through unofficial ports), together with an OpenGL ES 2.0 compatible graphics processing unit (GPU). Android supports OpenGL ES 1.1, 2.0, 3.0 and 3.1. GPU hardware is required to run such applications which may explicitly require a certain version of the OpenGL ES.

6. USE OF ANDROID EMULATOR

A virtual mobile device known as emulator that runs on our computer is included in the Android SDK. Without using a physical device, one can develop and test Android applications with the emulator. The command line features that can be used with the Android Emulator are described in this page. For information about using the Android Emulator UI, see Using the Android Emulator.

Starting and Stopping the Emulator

During the development and testing of our application, we install and run our application in the Android Emulator. The emulator is used as a temporary application for a command line within our Android Studio development environment. In both cases we used the AVD configuration to load and startup options. We can use a single emulator or we can start multiple emulator instances and run our application in more than one emulated device. We can use the emulator's built-in commands to simulate GSM phone calling or SMS between emulator instances, and we can set up network accordingly which allows emulators to send data to one another. For instance information we use the Telephony Emulation, SMS Emulation, and Emulator Networking. Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human and machine readable. It is defined by the W3C's XML 1.0 Specification and by several other related specifications, all of which are free open standards. Parsing XML refers to going through XML document to access data or to modify data in one or other way.

XML Parser provides way to access or modify data present in an XML document. Java provides multiple options to parse XML document. Here we use various types of parsers which are commonly used to parse XML documents.

7. CONCLUSION

With the increasing population and demand for food, it has become increasingly important to introduce new and efficient methods for farming. The farmers provide the food for existence of human population yet, their own existence is now endangered. Thousands of farmers commit suicide every year and thousands migrate to urban areas in search of work. The population of farmers has been reducing every day, the consequences of which are already visible to us. The prices of food products are soaring to new heights and there is severe shortage of food. Also reducing farming ground and conversion of fertile land to urban development is also a big contributing factor. Therefore, we have tried to make our contribution towards a big sector which is completely ignored and lack required knowledge and technology to strive towards a better future. We have successfully completed our project and our mobile application is up and running to help our country's farmers for a better tomorrow.

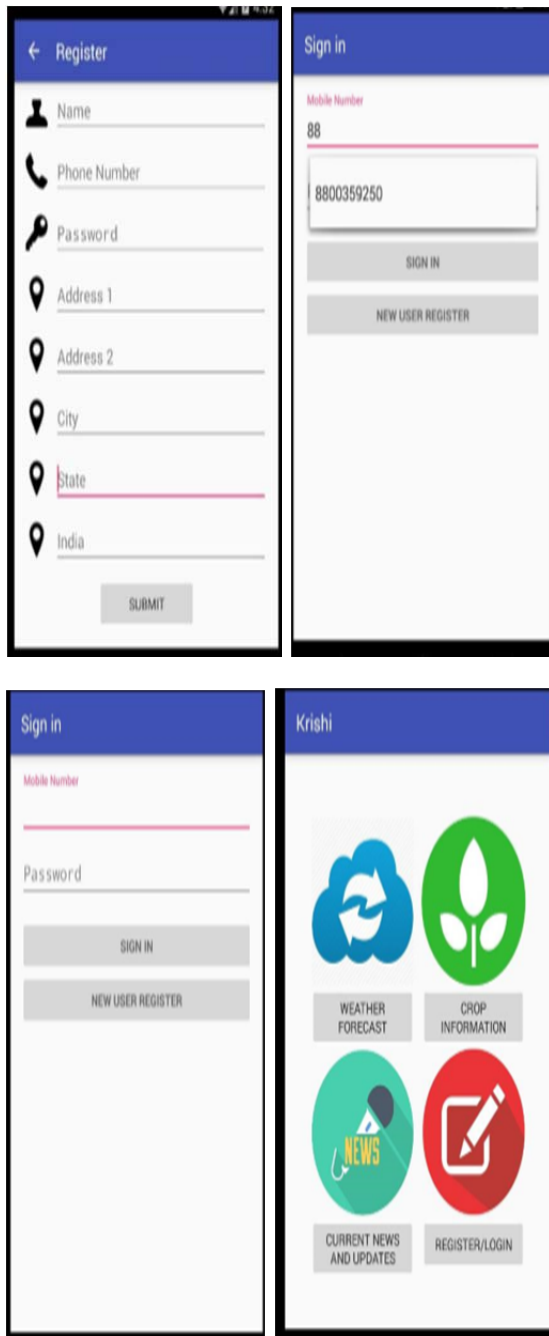


Fig. 1(a) User interface (b) Sign in activity (c) Sign In (d) Register profile

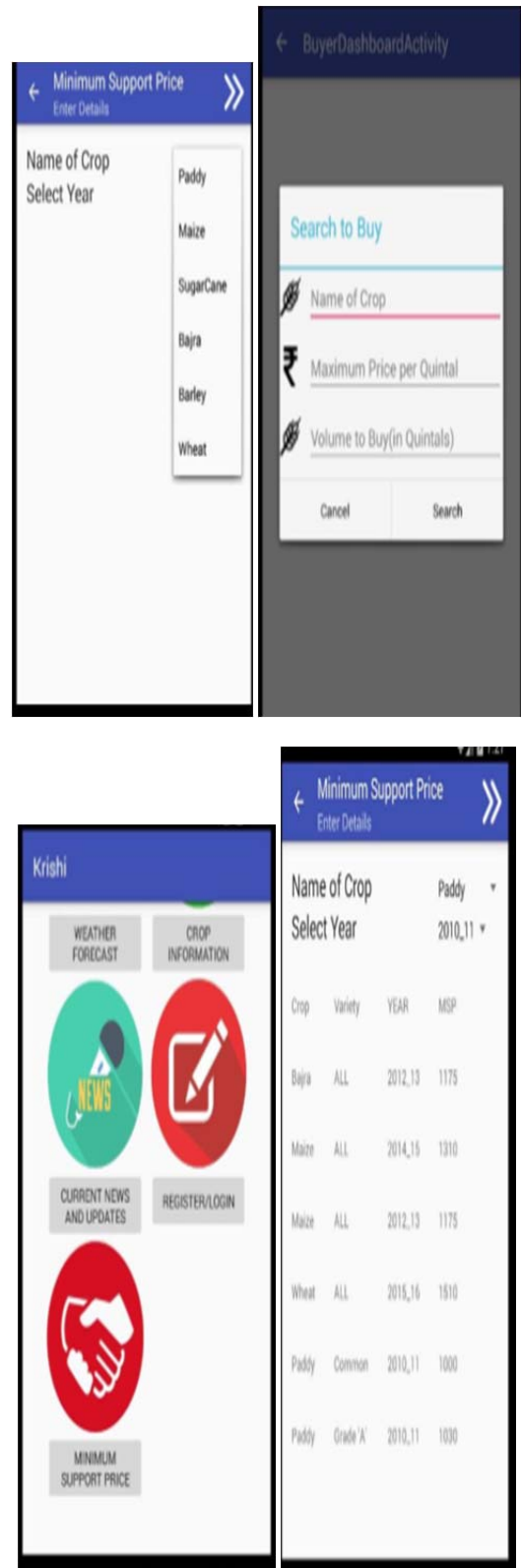


Fig. 2 (a) Buyer Dashboard Activity (b) (c) Minimum Support Price1 (d) User Interface2

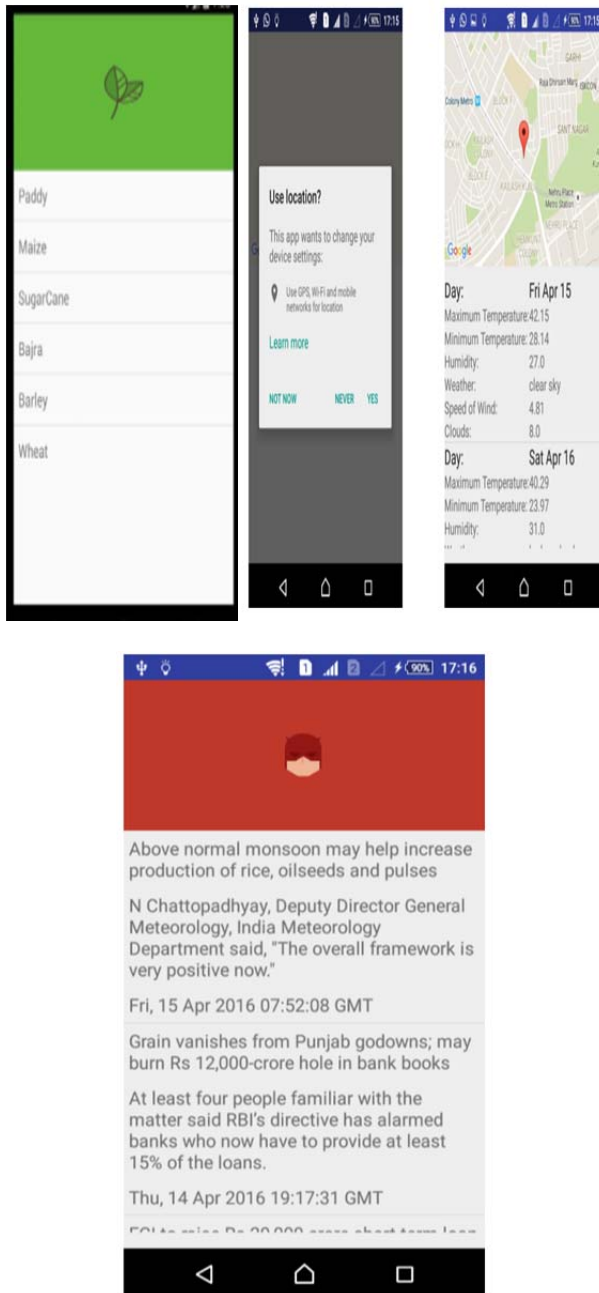


Fig. 3 (a) Minimum Support Price (b) Location (c) Weather forecast

REFERENCES

- [1] Davis, F.D. (1989). Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology. *MIS Quarterly*, 319-340.
- [2] Dutoit, T. *An Introduction to Text-to-Speech Synthesis*, Springer 2001.
- [3] Google Android TTS API <http://developer.android.com/reference/android/speech/tts/TextToSpeech.html>.
- [4] Hart, S.G. & Staveland, L.E. (1988). Development of a multi-dimensional workload rating scale: Results of empirical and theoretical research. In P.A. Hancock & N. Meshkati (Hrsg.), *Human mental workload*, 139-183. Amsterdam (NL): Elsevier.
- [5] Java Community Process: Java Speech API JSR 113 <http://jcp.org/en/jsr/detail?id=113>.
- [6] JSGF Java Speech Grammar Format <http://java.sun.com/products/java-media/speech/forDevelopers/JSGF>.
- [7] Lewis, C. H. (1982). Using the "Thinking Aloud" Method In Cognitive Interface Design. Technical Report IBM RC-9265 [8] SVOX TTS <http://www.svox.com/>.
- [9] Wickens, C.D. (2002). Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3(2), 159-177.