1 Abstract

The telecommunications industry has risen to become the modern era’s most crucial necessity. The world is now dependent on these means of global telecommunication, often taken for granted by many. Unfortunately, technicians often must undergo the burdensome task of locating sites of faulty equipment in order to keep our phone system functional. To facilitate the complex tasks of the technicians, we have designed and prototyped an industry grade Android application that simplifies all of their duties into a single compact portable device. The application can replace expensive barcode scanning equipment by utilizing the phone’s built-in-camera. With access to the Ericsson servers, the application is also able to provide information from the company’s location and equipment database within milliseconds, in comparison to the extended time spent searching through the thousands of pages of printed material.

2 Introduction

Telecommunications is one of the largest and fastest growing industries in the world. However, the future and success of the industry is heavily dependent upon its vast infrastructure, thus the reliability of this equipment is essential. As with all types of equipment, malfunctions tend to happen and to prevent long service blackouts, swift repairs must be made in order to retain customers and maintain the company’s brand. As a result, service providers look for efficient and trustworthy companies, such as Ericsson to manage this infrastructure.

Currently, telecommunication companies have both paper and digital records, but without a way to access their electronic databases from mobile devices, technicians are forced to carry cumbersome binders with entries for each destination and unique pieces of equipment or get the information from the online database with a computer with internet access - neither of which is a favorable choice for a busy and mobile technician. The process is highly inefficient and prone to human error. In this digital era, thanks to the omnipresence of mobile devices, it is possible to maximize efficiency, reduce error, secure data and minimize costs with an application for smartphone devices, in particular, for the Android OS. These things will be especially important as Ericsson continues to expand telecommunication networks, specifically mobile networks.

With these goals in mind we built an Android application that caters to the needs of service provider technicians and their clients by providing them access to information about Ericsson’s vast infrastructure around the world with literally the touch of a button. The application allows the user to conveniently and intuitively search, in a multitude of ways, for equipment locations, details about the equipment and even directions on how to find the equipment. In addition, due to the implementation of a barcode scanner through the smartphone’s camera, the
application can potentially replace expensive and highly specialized barcode scanners currently being used by technicians. Finally, keeping corporate security in mind, our application protects the data by accessing secure databases through HTTPS, required certificates, and username and password requirements.

The Android mobile platform is gaining universal prominence as a result of its wide availability and user friendly interface. Because it has desktop caliber features and all of the Java virtual machine compacted into a mobile device, the limits for application concepts are boundless. We have capitalized on the growth of the Android mobile platform to create an application that allows its users to search for equipment locations and details about the equipment on the go, thus facilitating restoration of telecommunications infrastructure. Furthermore, our application would allow telecommunications companies to replace specialized, expensive barcode scanners used to identify equipment with the camera on any Android smartphone.

3 Background

3.1 Development of Mobile Applications

For the past few decades, computers have led the personal aspect of the electronic industry. Their unprecedented computing power revolutionized society with its capabilities, able to manage any task imaginable. With its large monitor and user-friendly operating system, the personal computer revolutionized technology. However, the smartphone has recently emerged to supplant the computer as one’s primary source of computing. Although the small device has its limitations with its battery life, the portability of the device opens up a door of opportunities. By simplifying the user experience while still maintaining the appeal and content output, smartphones have proven to be extremely effective. In addition, it is superior to regular computers with its availability of numerous sensors: accelerometer, gyroscope, GPS, touch screen, camera, etc. Possessing all of these components along with mobile internet access in a compact device, the smartphone is gradually rising to become the one of the main computing devices of the future.

3.2 The Android Operating System

Android is the fastest growing mobile operating system available and currently occupies over 50% of the smartphone market of the world [1]. Unlike Apple’s iOS, which is restricted to devices specifically made by Apple, Android is open source so it is offered by various of manufacturers including HTC, Samsung, and Motorola. There are over a thousand unique devices which almost guarantees that one, if not many, are suitable for a specific use or an individual’s personal preference. Android runs Java, which has its foundation in cross platform compatibility, so code written in Java can run on any operating system.

On modern Android devices, the raw power of hardware components is nearing the netbooks and cheap laptops of a few years ago, including dual and quad core processors, one or two gigabytes of RAM and eight megapixel cameras. This impressive hardware makes Android devices up to the task for a limitless number of tasks [2].

3.3 Terminology

XML: (Extensible Markup Language) a language that defines a set of rules for documents in a format that is both human-readable and machine-readable. In this application, XML files are used to send
requests from the mobile devices to the database servers and receive the results.

Web services: the standard way to communicate requests to the server and receive responses back from its databases.

HTTPS: (Hypertext Transfer Protocol Secure) a protocol for secure communication over a computer network, which in this scenario is used to maintain a private, secure connection to the database servers in a corporate environment.

Common Language Database System:

CLEI Code: (Common Language Equipment Identifier Code) 10 character alphanumeric codes that have a 1:1 relationship to a product identifier (part number). These codes are used to identify equipment.

ECI Code: 6 character numeric codes used on equipment barcodes that correspond to a specific revision of a piece of equipment. Like CLEI codes these identify equipment, but are of a shorter length.

CLLI code: (Common Language Location Identifier Code): a code used to specify a telecom location and to specify the functionality in that location.

3.4 Prior Work

The Telcordia subsidiary of Ericsson is responsible for maintaining the data for telecom providers around the world. Ericsson possesses information on the locations of all main distribution frames and specific plug-in equipment in each location, the company distributes this vital information to service providers. Errors or issues with the equipment inevitably occur and with Ericsson’s source of invaluable information, technicians can immediately respond to issues and avoid possible blackout of local phone systems. However, this vast database is only accessible through the online web services or large uncomfortable binders. These limited means of access have been a clear impediment for technicians. The technicians lack any convenient method of immediately accessing the information on duty and must revert to the original two methods. With an Android application, the technician can access the entire database from any location with internet connectivity. With a simple user interface, the data can be accessed with ease rather than searching through thousands of pages of CLEI and CLLI codes or locating a computer to access a complex online service.

4 Software Creation

4.1 Overview

Although Ericsson owns the Common Language database system in which all their
location and equipment information is stored, it is not accessible on mobile devices and limited to computers. When redesigning the interface and placing all of the necessary features into a mobile device, much deliberation and extensive testing were required to maximize the simplicity and data output. Our application provides a simplistic and user-friendly method of accessing the online database swiftly and devoid of the extraneous steps needed to access the desired information. Putting much emphasis on Common Language’s CLEI and CLLI codes, we have implemented a graphical user interface, similar to the online version, to search for location information and a barcode scanner to quickly access vital equipment information with the touch of a button. This retrieval of information was made possible through XML parsing. By parsing XML files returned from the server, our Android application is able to communicate with the Ericsson database server by sending and retrieving specific commands. Within the content of these XML files contain details about the location or equipment asked about by the user. The application is then able to parse and separate the information intended for the user and the information intended for use in the database backends. The application is able to directly contact the server to provide a swift and accurate response through either Wifi or a cellular network, thereby eliminating tedious and inefficient occasions in the field involving searching through paper records and helping technicians find their destinations. Removing slowdowns caused by these lookups will help reduce telecommunications downtime in the event of malfunctioning deployed hardware.

4.2 First Mock Ups

Upon hearing the initial problem, we decided to break the application into three main parts, scanning equipment, equipment lookup, and location lookup, which could be selected through tabs. Placeholder menu buttons for access to a settings screen and an about screen were also included. The user interface was designed in Android layout files from the beginning, which allowed them to be used immediately after transitioning from the mock up phase to actual development.

4.3 Preliminary Functionality

The first code written was integration with the Zxing Barcode Scanner library, which can read both one and two dimensional barcodes [5]. Next came behind the scenes utilities, such as a parser which could translate received XML messages from the server into usable, readable data which would be outputted. Because network communication was not working at the time, pre-fetched XML files were used to test the parser and how it outputted data in the user interface. On a standard XML response using the Galaxy Nexus parsing performance was more than satisfactory, with the average time needed to parse being less than ten milliseconds. The XML parser takes the data it receives and extracts all the information with specific tags (for example, latitude, or product id). The parser then creates two descriptions from the data, a short one which is used in displaying results in lists, and a long one which features all the details. In the user interface, clicking on the short details in the list will display the long list for more information.

4.4 Implementing Networking Services

The application uses encrypted HTTPS connections and username and password when connecting to the database servers. Because networking requests are of unknown lengths, all networking logic had
to be put into a separate thread – or else the user interface would experience sudden slowdowns and halts. Also, various timeouts had to be put in place to stop trying to contact the server if the connection is dropped or if the server takes too long to respond. If the connection could not be made, or if the server appears to have dropped connection, an appropriate error message is given to the user.

Communication to the server is done by sending and receiving XML messages. When creating XML requests, it is common to look at the server’s WSDL which defines what commands are valid. However, our application creates the XML request by plugging its user’s data into a premade XML file. Because the server’s WSDL is not constructed every time by the program during this phase, request creations only takes a millisecond or two to accomplish.

### 4.5 Equipment Lookup

Equipment lookup appeared easier than location lookup because it only required the input of one number without any pre-parsing, so it was connected to the networking backend first. A user can either enter a CLEI code or scan a barcode, but both methods make the same networking backend request, with the sole difference that the generated XML request tells the server it has a CLEI or ECI code respectively. Once the networking thread reports that it is finished, the received XML response is given to the application’s parser. Upon the parser’s completion, the user interface is updated with the short and long descriptions.

### 4.6 Location Lookup

Locations are searched for with latitude and longitude, but because this tends to be a cumbersome method of identifying a location, it was realized early on that a way to search by addresses would be necessary. There is a method to request the server to
search based on address, but the specific formatting required by those requests would be too complex to simply display on the screen. Therefore, the application uses the Android geocoder service to translate addresses into latitude and longitude. This allows for more natural and flexible address input. Furthermore, Google’s map services are updated much closer to real time than the locations are. This means that in the event of a street name change, it is much more likely that translating the new street name through Google and searching through latitude and longitude has a higher chance of success than searching the database server, which might not recognize the new street name. Searching by latitude and longitude was left in the application to cover any event in which it would be more convenient to search by coordinates than a street address.

4.7 About Page
As part of our effort to maximize the user experience an about page was added in the case the user faces any issues with the application. The page was split into two sections “About” and “Contact Us”. In the about section we allow the user to directly access Ericsson’s legal information along with its hours from the application, using a Linkify Intent to make links clickable.

Similarly in the Contact Us section we also use Linkify Intents to allow the user to directly contact Ericsson’s Customer Service via phone or email from the application itself.

4.8 Feature Finalization
The last features to be implemented included a settings screen to change usernames, passwords, and server URLs. These settings are hooked directly into the XML creator and etwork backend. On the first launch of the application, the user is prompted to enter his or her credentials.

Moreover, during location lookup the user can tap a button to instantly get GPS navigation to one of the results. This is done using a Google Maps Intent, present in every Android phone, thus avoiding legal
licensing issues and minimizing future costs.

Also, a layout, which leverages on the extra screen real-estate, specifically for tablets was added. The tabbed interface with the search inputs and list of results remains on the left side of the screen, but instead of showing a dialog box for more details on a result, the details are shown in a pane on the right half of the tablet.

![Image of tablet interface]

4.9 User Interface Polishing

Near the end of development, the user interface received a major redesign. The three tabs (Equipment CLEI Search, Location Search, and Barcode Scanning) were consolidated into two tabs, Equipment and Location in an attempt to make the application flow more obvious. In the location tab, searching by address or latitude and longitude was separated by a drop down menu. Tapping the drop down menu allows the user to change between searching by address or coordinates and changes the text input boxes accordingly. In the equipment tab, buttons to search by the entered text or barcode are separated.

Error checking on user input was added. For example, CLEI codes are always ten characters long. If the app’s user tries to search with an eight character CLEI code, a dialog saying “CLEI codes have 10 characters, the one entered has 8” will appear. This small means of alerting the user has dramatically helped new users get past the learning curve of a new app.

Finally, small animations were included to add another degree of usability to the app. For example, performing a search slides the old results out of view to the right while the new results enter from the left, and choosing search by coordinates will slide away the address input boxes. These small hints in the interface help make the experience more intuitive and visually attractive.

5 Results and Discussion

5.1 Testing

The application was given to various mentors and other students at Governor’s School as well as Ericsson employees. The feedback we received included feature suggestions such as the ability to get directions and user interface changes, such as the consolidation of the original three tabs into two and an About screen. By watching test subjects use the app, we realized that errors needed to be handled more clearly and thoroughly.
5.2 Results

After thoroughly testing and making the suggested changes, a second round of field testing was carried out with other program mentors and Ericsson employees in which a considerable amount of intuitive knowledge about the workings of the app was shown, suggesting that one of our primary goals for an intuitive interface was attained. Furthermore, during the testing period the speed at which the application both sent and retrieved the data was also noted. Average times of about 2 seconds for equipment lookups and anywhere between two to four seconds for location queries were calculated; all of which also suggest that our other goal for efficiency was also accomplished. Finally with positive reviews from Ericsson employees a final release was pushed out that accomplished all of our initial goals. Striving to demonstrate the potential of this application, a tablet version was also released that met our initial goals.

5.3 Discussion

Every feature originally planned for the app and more has been fully functional. The original goal for this application was to be able to search for equipment simply with a CLEI code, but the current version has the added functionality to search for location, both by coordinates and street addresses as well as the ability to scan barcodes [5]. The location search by both address and coordinates were implemented primarily to make it easy to search based on the information given, and making it convenient for technicians to get directions to their destination. Upon presentation, Ericsson was impressed by our application and has decided to use it as a prototype for a future mobile application to be deployed to its client’s technicians.

A future area of research can involve having Ericsson possibly move their web services from XML, widely known to be outdated and slow, to a more modern technology such as JSON, which produces both smaller files, shortening the time to download responses, and simpler syntax, allowing faster data parsing. [4]

6 Acknowledgements

We would like to thank Darold Hemphill and Cynthia Gipson Lee of Ericsson for giving us the opportunity to gain real world experience in programming and telecommunications. We also would like to thank Christine Hung of Ozmosis Learning & Consulting for being our project advisor and aiding us through the entire development process. We also would like to thank Amanda Rumsey as our project mentor and overseeing all of our progress. We would also like to thank Ericsson and the Common Language Group including Chris Bain, Rucksapol Jiwungkul, Alicia De Sordi, and Christine Fabrycki. We also would like to thank Adrien Perkins and Stoyan Lazarov as our residential teacher’s assistants. We also would like to thank the Governor’s School Board of Overseers and the sponsors of the 2011 School of Engineering and Technology: Rutgers University, the State of New Jersey, Morgan Stanley, Lockheed Martin, South Jersey Industria, Inc., and PSE&G for their support of the Governor’s School Program. Finally we would like to thank Dean Ilene Rosen, Ed. D. – Director, Jean Patrick Antoine - Assistant Director, and the Governor’s School Board of Overseers for the entire Governor’s School of Engineering and Technology program.
7 References

   http://www.comscore.com/Press_Events/Press_Releases/2012/7/comScore_Reports_May_2012_U.S._Mobile_Subscriber_Market_Share
   http://www.openhandsetalliance.com/and

[4] JSON
   http://www.json.org/
[5] Zxing Barcode Scanner
   http://code.google.com/p/zxing/