

CONTROLLED ACCESS CABINET DOOR LOCK VIA ANDROID PHONE

Diomedes P. Fedelin Jr.

John Neil G. Malabed

Niña Shella A. Manalo

Katrina Marithes P. Roxas

ABSTRACT

This software and hardware development project provides practical application of today's emerging technology particularly the Android Operating System. This study tested the capability of Android Mobile Phone for controlling the locking and unlocking of cabinet door for offices with the use of different software like Eclipse SDK, Arduino 1.0.1., and Microsoft Visual Studio to write the programs of the entire system while Microsoft SQL Server Management Studio was used for managing the database of the users. This research adds a new perspective upon usage of technological advancements to produce dependable applications for convenience and safety.

Keyword: Android Application, Door Lock, Security System, Mobile Phone, Computer

Introduction

The level of capability in contemporary smartphones is remarkable and continues to grow [1]. As the development of mobile communication technology, the mobile device has played a very important role in daily life and it is in great public demand [2], [3], therefore, why not used this as a triggering device to lock and unlock the doors without using door lock keys. This is one way of utilizing the application of mobile technology for replacing an old-age manual security of having padlock and keys, or alternative to biometrics, pin codes, keycards or RFID cards. This is an added feature to the use of mobile phone as the primary key to open not only one cabinet door but as many doors as the administrators allowed the registered user to give access to open the cabinet door. Therefore, the researchers developed a door lock system via Android Phone for the cabinet doors in the offices to secure confidential records and company properties.

Google came up with an innovative operation system termed as ANDROID, which is open system architecture with customizable third party development and debugging environment which helps the user's to manipulate the features [4] and it offers developers free platform to make their own applications [5].

This study aimed to create a prototype cabinet door that is suited for the electronic door lock which is connected to the microcontroller platform and being controlled by the android application.

Android Mobile Platform

Android is a fully integrated mobile “software stack” that consists of an operating system, middleware, user-friendly interface and applications [6] that make life more comfortable and advanced for the users [7]. It gives power to hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast every day another million users powered up their Android devices for the first time and start looking for apps, games, and other digital content [8]. Android SDK is a software development kit (SDK) that enables developers to create applications for the Android platform which includes

sample projects with source code, development tools, an emulator, and required libraries to build Android applications [9].

With Android, users can quickly and seamlessly switch between apps and pick up whatever they were doing. Juggling multiple tasks at once on a mobile device has never been easier [10].

System Design

As shown in Fig. 1, the user shall send request to the Android application by logging in.

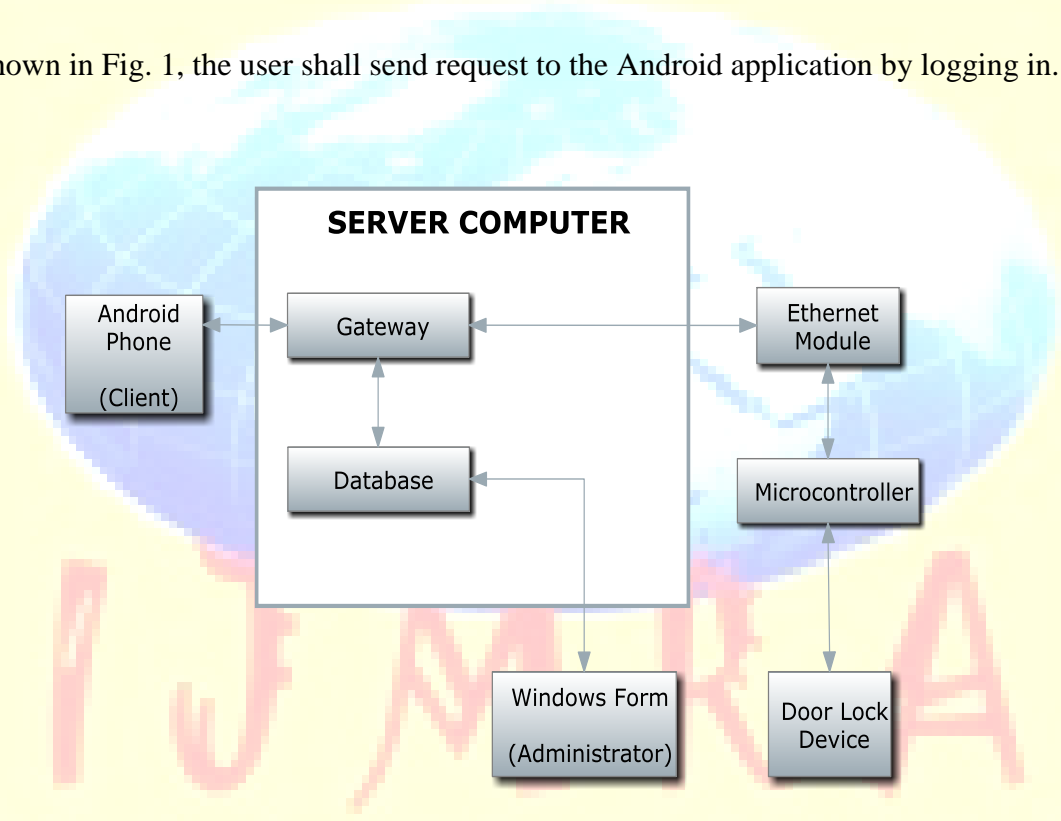


Fig. 1. System Design Block Diagram

The data from the user which are the username and password will pass through a server for the Android phone. This server acts as the gateway to the database. The server will check the details given by the user. The server will send data to the Android phone to verify if the “log in” details from the user are accurate. If the “log in” details are correct, the server will send the list of the appropriate doors assigned to a specific authorized person from the database. Otherwise, the

Android application will inform the user that his “log in” details are wrong. After the success of logging in and the list are already in the Android application, the user can now select the door that shall be opened. The request will pass through the gateway again. The gateway will send the request to the Ethernet to Serial Shield. Since this shield has the ability to talk to other devices, it can send the data to the microcontroller platform. This platform is connected to the Ethernet to Serial module through serial connection. This gives signal to the microcontroller. The microcontroller will send signal to the door lock system to unlock. The door lock has timer, it has a specific time when it will be locked again. During the testing stage, the designers set the timer to one (1) minute.

Software and Hardware Development

The proponents decided to produce three doors in a single line for the prototype to better demonstrate the functions of the system. The electronic lock was then connected to the relay driver wherein it gives the signal to trigger the solenoid to lock or unlock. Since the electronic lock is normally closed, the solenoid will remain locked until there is an electric signal sent from the relay to the solenoid. The relay can hold a maximum of two solenoids so the proponents used two relays. The relays were connected in parallel for the power adapter. Also connected in the relay was the microcontroller unit which is the gizDuino. The gizDuino holds the program for the locking and unlocking of the solenoid. All of these components are connected to a 12 volt power supply.

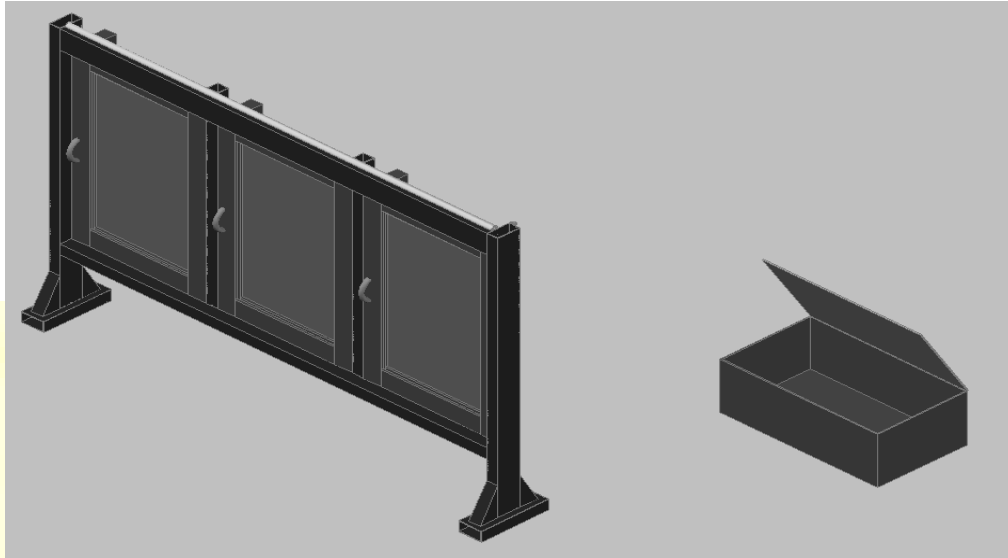


Fig. 2. Northwest Isometric View of the Cabinet Door.

For the system to function properly, the Database of the system, Gateway Console application and Door Lock System Administrator application can be installed in a Windows 7 operating system. The Administrator application is viewed in a windows form which is used to view the history and create accounts. Android v.2.2 is the minimum requirement for the smartphone since the system will not use a lot of graphics. Another software requirement was programmed in Arduino. The code generated is loaded into the microcontroller.

One of the minimum software requirements is the Windows 7 OS. It is included in the requirements because some application doesn't run in the earlier version of operating systems. Eclipse SDK is also used because the Android application is being generated in this program. Eclipse SDK platform makes it easier to build great looking rich client applications. It is utilized because of the facilities that make it easy to design and customize an application user interface (UI). The application also adds greater flexibility to UI layout, making it easier to make a user interface looks different from an Integrated Drive Electronics (IDE) [11]. Arduino 1.0.1 was used to generate the program code for the microcontroller while the Microsoft Visual Studio was used to generate the program for the Administrator and the gateway part. The proponents decided to use MS Visual Studio since it is a high-level language and it is object-oriented. Another software

requirement is the Microsoft SQL Server Management Studio which is a software product whose primary function is to store and retrieve data as computer or those running on another computer across a network [12].

The program for the gizDuino microcontroller was written using the Arduino 1.0.1. The program loaded to the gizDuino microcontroller through a USB cable. The code will be the controller of the pins of the MCU. Pins 2 to 15 are configured to act as input for the solenoid. The programs for the Gateway and the Administrator were developed in Microsoft Visual Studio 2010. Basically, the main function of the Gateway is to give connection. The Gateway is coded in a console application since it will only show if the connection between the other system components is active. The Gateway is being coded to check the details given by the user. The Gateway checks if the credentials given by the user is already in the database. After the request has been made, the gateway should be able to send the request to the Ethernet to Serial module through serial communication. The Gateway includes protocols that are used to communicate with the Android application. The Gateway is also coded in such a way that it listens to port 2000. Port 2000 is used because the higher the port number, the higher the possibility that it is not used. The Administrator part was developed to access the information of users and the door history. The system's database is being developed using MS SQL Server Management Studio 2008. The database depends on the requirements of the system. The design of the tables is reliant on the components of the system. Since the system has basically users, doors, and history of transaction as its components, the database has a minimum of three tables. The tables have its corresponding queries and relationships in order to give the user the correct information. And the last program is the Android application.

The Android application was programmed using the Java language. The only way to connect the Android to the Gateway is thru Hypertext Transfer Protocol (HTTP), the proponents integrated it into the program, since HTTP is in the Network layer of the OSI model.

Results and Discussion

The researcher designed the following flowcharts based on the step – by – step program instructions and the processes involved on how the system worked and achieved the objectives of the study.

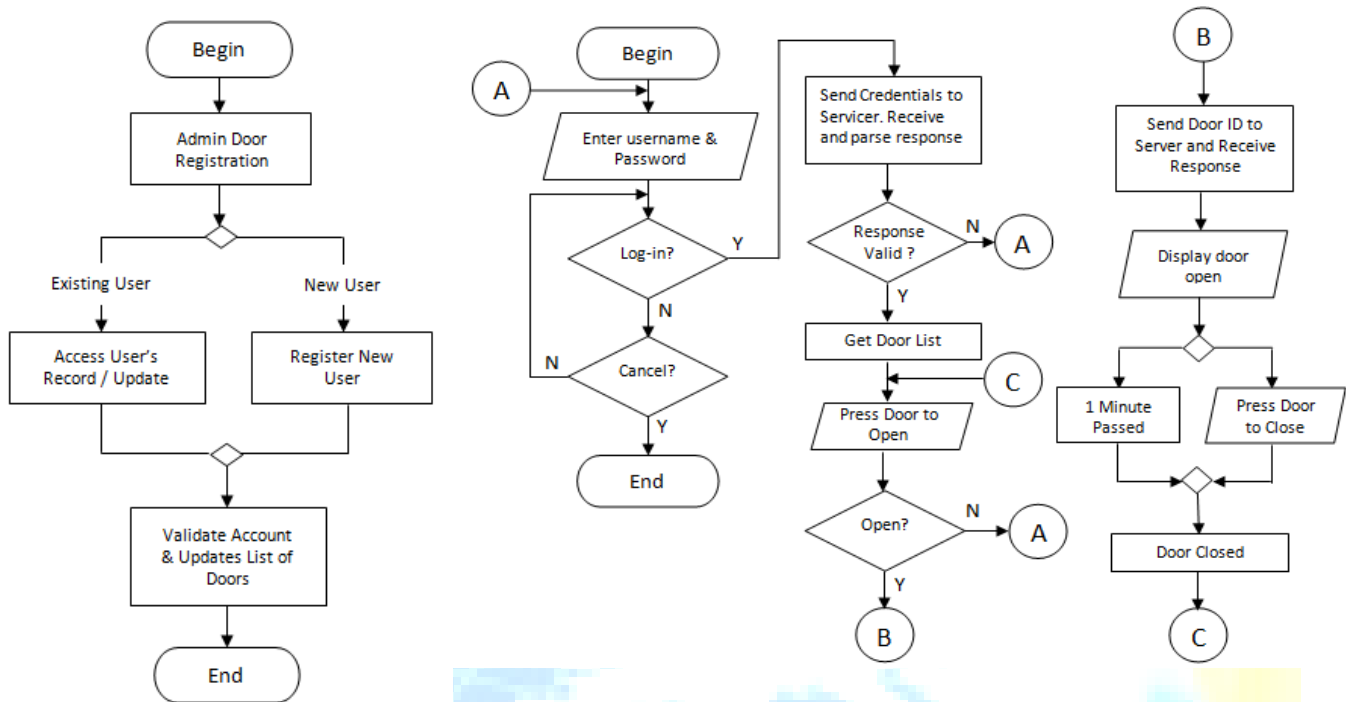


Fig. 3. Door User Registration Flowchart Fig. 4. Validation of Door Opening Request Flowchart

Fig. 3 demonstrates how the administrator registers a door to a user. The administrator has to identify first if the user already exists or a new user. If the user already exists, the admin has the capability to update the registration of the user and it has to be validated. If the user doesn't exist, the administrator has to register him first before validation. The registration needs valid employee number, username and password. Upon validating, the administrator can now update the list of doors registered to a certain user.

Fig. 4 shows how the request of the user will pass through the Android application. The user has to encode the username and password and click the action button ("Log in" or "Cancel" Button). If the action is 'cancel', then the program terminates itself, but if not, the request will be sent to the server. The server will receive and parse the request. If the request of the user is invalid, the page will return to the home screen; otherwise, only the list of the doors which the user is

authorized to enter will appear on the screen. The user will select the desired door. If the action is not to open the door, it will return to the home screen of the application. But, if the action to be done is opening the door, the door ID will be sent to the server. The server will receive response again, until the “Successful” message will appear.

The system has also the capability to change the password of the Administrator. Each door has also a unique name, its own IP Address and Port Number.

Username	Door Name	Date Time
dlsadmin	Room 309 opening	2/10/2013 11:07:13 AM
dlsadmin	Room 309 opening	2/10/2013 11:07:28 AM
dlsadmin	Room 308 Opening	2/10/2013 11:07:48 AM
dlsadmin	Room 308 Opening	2/10/2013 11:08:03 AM
rommelpureza	Room 308 Opening	2/11/2013 11:32:26 AM
rommelpureza	Room 308 Opening	2/11/2013 11:32:34 AM
rommelpureza	Room 308 Opening	2/11/2013 11:32:49 AM
rommelpureza	Room 308 Opening	2/11/2013 11:33:54 AM
rommelpureza	Room 308 Opening	2/11/2013 11:34:09 AM
dlsadmin	Room 308 Opening	2/11/2013 11:38:15 AM
dlsadmin	Room 308 Opening	2/11/2013 11:38:31 AM
rommelpureza	Room 308 Opening	2/11/2013 11:39:42 AM
rommelpureza	Room 308 Opening	2/11/2013 11:39:51 AM
rommelpureza	Room 308 Opening	2/11/2013 11:39:57 AM
dlsadmin	Room 309 opening	2/11/2013 3:37:29 PM
dlsadmin	Room 309 opening	2/11/2013 3:37:44 PM
dlsadmin	Room 308 Opening	2/11/2013 3:38:24 PM
dlsadmin	Room 308 Opening	2/11/2013 3:38:30 PM

Fig. 5. Database Transaction History

Fig. 5 illustrates the database transaction history which shows the name of the door with its corresponding employee’s username who opens the door which also includes the time and the date when it was accessed.

Fig. 6 shows the process of sending credentials to server and how the server will always have to listen to the device connected to it to receive request.

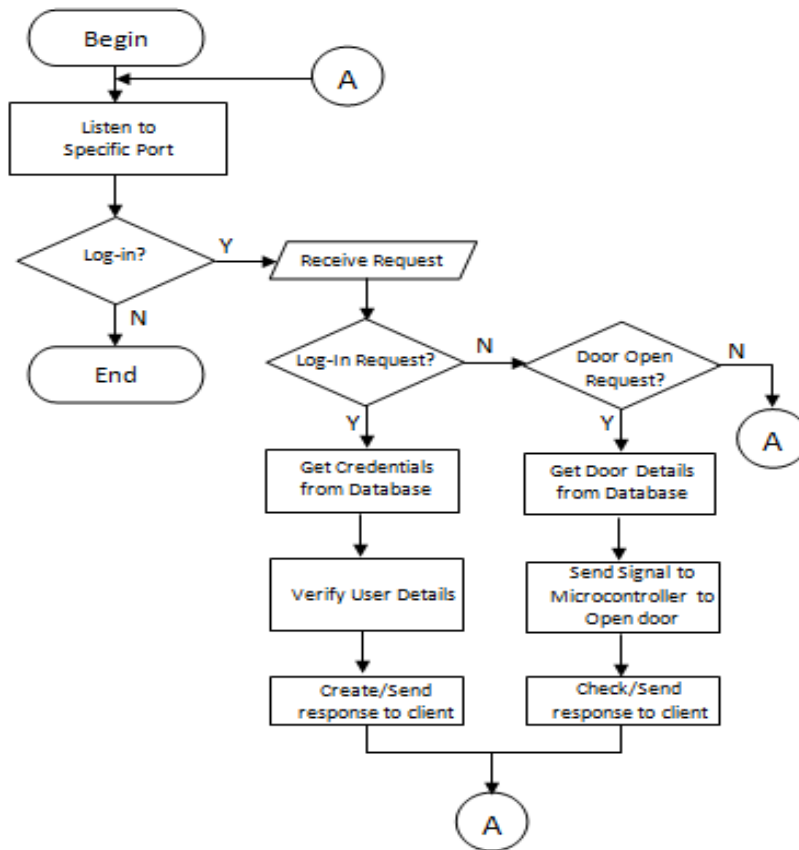


Fig. 6. Flowchart of Sending Credentials to Server

If the request is to “Log in”, then the server will get the account details of the user from the server and verify it. It will create/send response to the client. After that, it will have to listen again. But if the request is not to “Log in” but to open the door, the server will get the door details from the database. It will send the signal to the Ethernet to Serial Module to the microcontroller. After checking and sending response, it goes back to listening after. If the system found out the response is valid the list of doors will be displayed and the user can be allowed to access the designated door.

After series of testing and evaluating the functionality and accuracy of the prototype design, the researchers sometimes encountered in the delayed of updating the image of locked and unlocked door in the mobile phone but the opening and locking of the door are working properly without experiencing delay.

Conclusions and Recommendations

The growing mobile technology gives way to different types of applications. The researchers used this as an advantage to associate it with convenience, portability and accessibility of Android phones where the proponents used as the major platform for the system.

This study provides extra security to doors and cabinet doors alike. Because of the microcontroller included in the system, all the signals will be processed by the program embedded in it. The microcontroller can then now give the command to the relay and into the electronic lock to open or lock it. The relay will just be the signal generator for the electronic lock.

For further improvements of designing a Controlled Access Cabinet Door Lock via Android Phone, proponents of the study recommended to explore the mobile applications security to ensure the safety of the system. This security is very important taking in account the high number of mobile devices and users [13]. When it comes to the design of the prototype, the connection between all electric components should be properly connected to avoid short circuit; place the solenoid on the door with the proper orientation of the plunger; the door should have a Hydraulic Door Closer instead of using rubber for the door to close properly; and the network connection should be configured properly to avoid miscommunication of the devices.

References

- [1] D. Sora (2010). Monitoring and Controlling Automation Systems using Smartphones/PDA. *Journal of Defense Resources Management*, 1 (1), p. 117
- [2] X. Yu and Q. Wen., (2011). Trusted Mobile Client for Document Security in Mobile Office Automation, *International Journal of Information Technology and Computer Science*, 3(1), p. 54-62.
- [3] R. Senjaya, A. Y. Pratama and D. Agushinta R., (2013) Mobile Application for Diabetes Risk Diagnosis and Information Based on Android, *World of Computer Science and Information Technology Journal*, 3 (2), pp 44-49.
- [4] A. Ramalingam, P. Dorairaj and S. Ramamoorthy. (2012). Personal Safety Triggering System on Android Mobile Platform. *International Journal of Network Security & Its Applications*. 4(4), p. 179.
- [5] H. J. Yoon (April 2012). A Study on the Performance of Android Platform. *International Journal on Computer Science and Engineering*, 4 (4), p. 532
- [6] O. Malik, Google Launches Mobile Phone Platform, Android: What it Means, What Experts Think, Nov. 5, 2007, available: <http://gigaom.com/2007/11/05/google-launches-mobile-phone-platform-android/>
- [7] J. B. Chintalapati and S. Rao. (September 2012), Remote computer access through Android mobiles, *International Journal of Computer Science Issues*, 9(5) p.p. 363-364
- [8] Android, the world's most popular mobile platform, available: <http://developer.android.com/about/index.html>, date retrieved: April 4, 2013.
- [9] Android SDK, available: http://www.webopedia.com/TERM/A/Android_SDK.html, date retrieved: April 4, 2013.
- [10] Android – Multitasking, available: <http://www.android.com/about/>, data retrieved: April 4, 2013
- [11] Eclipse SDK 4.2, Available: <http://www.eclipse.org/eclipse4/>
- [12] What is SQL Server?, available: <http://www.easeus.com/resource/sql-server.htm>
- [13] P. Pocatilu. (2011). Android Applications Security, *Informatica Economica*, 15(3), p. 163