Android-Based Smart Voting System

Inspiring Excellence

Conducted By:

Tawseef Mahmood	(10301006)
Shamah Mahbub Zoha	(11301007)
Ashis Kumar Das	(11301002)

Supervised By: Dr. Amitabha Chakrabarty

School of Computer Science and Engineering

BRAC University

Submission Date: 17th August 2016

Declaration

This is to certify that this final thesis report is submitted by the authors for the purpose of obtaining the degree of Bachelor of Science in Computer Science, and the degree of Bachelor of Engineering in Computer Science and Engineering. We hereby declare that all the instances of work presented in this thesis are original and inspirations for the work that we have made use of have been duly accredited with proper referencing.

Signature of Supervisor

_____

Dr. Amitabha Chakrabarty

Signature of Authors

_____

Tawseef Mahmood

_____

Shamah Mahbub Zoha

_____

Ashis Kumar Das

Acknowledgements

We would like to express our heartfelt gratitude to our supervisor, Dr. Amitabha Chakrabarty, Assistant Professor of the School of Computer Science and Engineering of BRAC University, for laying the foundation for our thesis concept and providing his valuable insight and guidance at each and every step of the development process.

Additionally, we would like to thank Dr. Md. Khalilur Rahman, Arafa. Mohd. Anis, Hamidur Rahman, Jennifer Sherry Alam, Sohel Islam Nabil and Syed Mahmud Hasan for their work on Electronic Voting Machines, which served as a starting point and major reference source for our thesis. We extend this appreciation towards authors from various other sources who have provided all the relevant information in their work that has made the completion of this thesis possible.

Furthermore, we would like to thank our family and friends for their continued support and encouragement throughout. Without them, this would not have been made possible, as they have helped push us further and further to reach our desired destination.

Last, but not least, we would like to thank our peers who have lent their support, advice and much appreciated words of encouragement. Their valuable input and suggestions with respect to our thesis is thoroughly appreciated.

Abstract

The process of efficient voting is a vital component in the proper functioning of our increasingly-modernized society. The traditional way of conducting a voting process is not only time-consuming, but also prone to security issues and election rigging, as well as being outdated and wasteful. For this purpose, Electronic Voting Machine or EVM was introduced which addressed some of the issues while having limitations of its own, such as its scope of application and its inherent limitation of being a mere electronic device, prone to malfunction and other mechanical issues[1]. Hence, there is a need for a system that would build upon that concept as well as being more compact, elegant and cost-effective, while at the same time capable of catering to a broader range of populace, all at the same time maintaining fairness and impartiality throughout. The Android-Based Smart Voting System that we propose will simplify the voting process by considerably reducing the steps for voting and increasing overall throughput, keeping interaction between voters and the system easy and understandable, as well as eliminating unnecessary and costly hardware that have no overall impact on the final outcome, thus providing for a smarter and more feasible solution.

Index Terms — Fingerprint, Biometric template, Device Driver, Microcontroller, Arduino, Android Smart Phone, Database, Security, Scanner, Thermal Printer.

Table of Contents

CHAPTER 1

Introduction

The process of electing a leader who will be responsible for carrying out the will of the normal populace is one of the most crucial and fundamental elements that comprises a democratic society[2]. Voting is an exercise of power and sovereignty; a powerful tool to show not only approval, but also displeasure if needed and firmly establishes the control of fate in the hands of its people. As societies progress, new complexities arise which make conventional methods of voting more and more difficult. Relying on physical ballots alone is no longer an option[3]. As a result, there has been a gradual shift towards electronic methods in conjunction with traditional, physical methods to preserve the legitimacy and integrity of the voting process, similar to the approach adopted by India[4]. Additionally, there has always remained dispute concerning the voting process and the outcomes derived from it throughout recorded history. There is a distinct lack of trust permeating throughout. Some issues have been addressed over time, but there has never been a completely sound approach that would win back the lost trust among the voters. Therefore, it has become more important than ever to devise a system that would make the process of voting stripped to the bare essentials while still remaining robust and fail safe.

1.1 Motivation

The last major election that was hosted in Bangladesh occurred in 2014, with 47,262,168 votes being cast from a total of 92,007,113 registered voters, an astronomical amount which represented a significant obstacle in ensuring efficiency and complete impartiality, as the population growth continues to increase exponentially[5].

With such an upsurge in population increase, the next elections in the country will become progressively challenging, as newer and newer voters are introduced and keeping track of each and every vote becomes more time consuming and inefficient. As such, credibility and integrity

of the whole political landscape may be affected, as questions regarding the practicality of traditional methods and vote manipulation methods pop up. As a result, the Smart Voting Machine or SVM has been conceived that will take on this enormous challenge and deliver the desired results, while ensuring maximum security and maintaining the upmost credibility.

1.2 Thesis Outline

The outline for our thesis is as follows:

- Chapter 1; the introduction of the thesis, the motivation behind creating this system to address immediate limitations on currently existing systems, as well as the organization of our thesis work.
- Chapter 2; this contains the background study done for the thesis, which includes current election practices that is relevant in our society. The chapter also includes the literature review that sheds light into the purposes of making this system using information from all the current, relevant work as a reference point.
- Chapter 3; presents an overview of the system, a bird's eye view on how the system works. It also includes details on the user types expected to use, maintain or interact with this system in any shape or form, as well as the devices that are needed in order to conduct the process and consequently ensure the successful completion of voting. The final integration of all the different components to produce a single, cohesive voting unit is also highlighted.
- Chapter 4; provides a detailed description of the actual implementation of the system developed, including all the structural and functional details of how voting is achieved, including libraries, device drivers, database and flow of information through the application.
- Chapter 5; asserts the analysis and findings of the experimental results derived from the system. It also provides an insight as to what challenges, obstacles and difficulties were challenged, as well as the technical and ethical considerations that were taken.

- Chapter 6; highlights the system limitations owing to practical circumstances, the work that can be done in the future to address some or all of these limitations, as well as newer features that can be implemented, and the concluding remarks on our final system and how it can revolutionize the way voting is approached in this country or anywhere in the world.

- References are provided at the end of the report for easy access to important terms, and for ease of comprehension on all the relevant information regarding our Smart Voting System.

CHAPTER 2

Background of Study

The introduction of Electronic Voting Machines have brought with them a whole range of solutions but, at the same time, a whole different set of problems that need to be addressed in order for it to be considered a resounding success. The purpose of this background study into what limitations the currently available systems possess is a point of interest for the purpose of our Smart Voting System, as we intend to minimize as much issues and concerns plaguing current systems as plausible, so that the process of voting is as smooth, quick, effective and reliable to conduct as possible.

2.1 Voting System in Bangladesh

The current practices of voting found in Bangladesh consists of multi-faceted and arduous tasks that involves manual collection of information with the help of government officials, who receive information concerning eligible voters and then compile a complete list of potential voters and their individual information. The list is then used by polling agents in the booth when voters come to vote, the information on their National ID is checked against the given information in the list and, if everything matches, they are allowed to proceed with the voting process. More recently, there has been a shift towards smart national ID cards, which are more durable than the traditional paper-based version of the IDs that we use today[6]. The information on the ID card is collected and warehoused in a centralized voter database. This move towards technological dependence is being made to address certain issues that pertain to the current practice, such as the legal loophole of a voter being able to cast their votes on different polling booths, provided they've not cast their vote in any of those booths previously, to name one instance. Further issues include fraudulence and vote tampering that comes along with physical balloting[7]. It is fairly predictable that voting, as we now know it, will continue to change, and lean more and more towards electronic and digital methods. As such, a complete system that addresses the inherent issues of the current implementations need to be addressed for large-scale

deployment in this country without any hindrances that will smoothly handle the ever increasing population count, and as the number of registered voters continue to rise as a result of education being more and more accessible to the majority of people.

2.2 Literature Review

Throughout the course of history, continuous changes have been made in the voting procedures, as progressing knowledge has made room for more sophistication, with which came along a number of accompanying complexities, arising with time. From simplistic, hand-written ballot papers from ancient times to the more recent manual system of having a selection of candidates available on the ballot paper, from where the voters chose their selected candidate using physical markers such as fingerprint stamps, or writing down their choice using pens, voting has evolved a long way, and still continues to do so.

As is to be expected, such systems are not without its drawbacks. For one, it is an increasingly laborious and time consuming process, subject to human errors and also a source of perpetual, undesirable monotony when it comes to sorting and keeping count manually, from ballot to ballot. In addition, it suffers from the obvious threat of malicious intentions, as it is very easy to manipulate vote counts by adding or removing ballots, or removing and re-applying marks that can be easily smudged, such as fingerprint stamp pads.

With the advent of technology, Electronic Voting Machines came into the foray of the voting scenario. Usage of these machines eliminated the use of physical means of putting votes, which were also noted to be less prone to manual manipulation (as the marks produced by these machines weren't easily removable), and also facilitated voting efficiency by minimizing time and energy lost undergoing rigorous, physical voting procedures.

However, these machines had their own set of inherited flaws. Firstly, like all machines, they are prone to malfunction, and it is costlier to maintain and make use of these machines on a continuous basis. Additionally, these machines couldn't address the issue of rigging[8], and there have indeed been instances of such an occurrence happening, such as the controversy surrounding the 2004 US National Election[9].

Our proposed Smart Voting System is the next step in the logical progression in the evolution of the voting process. It addresses most of the concerns of both physical and electronic means of voting. Just like its electronic counterparts, the Smart Voting System is mostly reliant on technology and therefore much faster than any conventional, physical form and is also more accurate, effective, efficient and fast. It does not suffer from the disadvantage of higher costs or limitations with respect to hardware, as is seen in some machine-based systems where there is a limit as to how many voters can be registered per machine. It uses minimal hardware, meaning it has much less maintenance issues and costs, and at the same time, it operates on scaled-down, optimized data that does not put a burden on the system. Security regarding the system is also full proof and it is much less prone to vote manipulation when put in conjunction with conventional voting ethics. Overall, it can be fairly assumed that this system is the archetype of what the next step in the voting process is going to be, and will be lauded as a technological and revolutionary landmark that safeguarded the integrity and values of a democratic society in its purest form.

CHAPTER 3

System Architecture

The tasks that the Android-based Smart Voting System shall undertake can be broadly classified into three functions; firstly, taking the fingerprint of the user as input to authenticate users and their type; secondly, the actual voting process after the user is verified, and finally, producing a physical output, a printout of the vote cast by each voter, to serve as evidence of their respective votes, which is to be placed in a physical ballot box to maintain its credibility. These three functions comprise together to form the basis of our system.

The system architecture of the system is as follows:

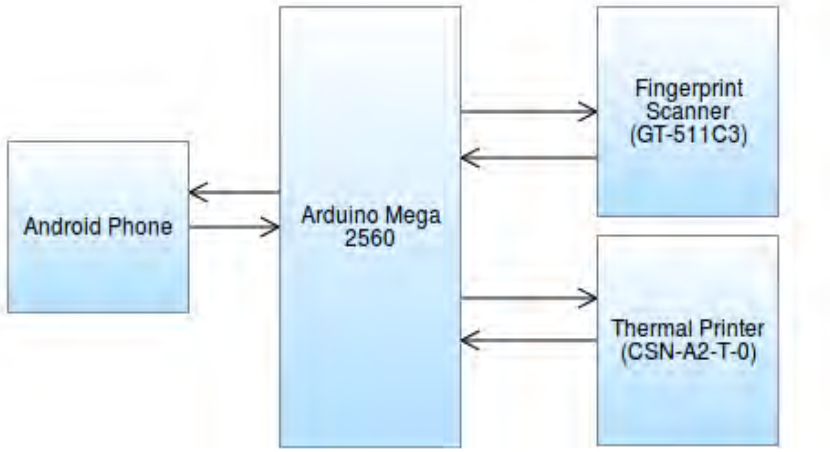

Figure 3.1: System Architecture of the Smart Voting System

3.1 System Overview

As stated earlier, the Smart Voting System divides the total voting process into distinct parts. Initially, the super administrator is in possession of the voter database that has been prepared. He or she loads it onto the internal memory of the Android device after verifying that he or she is the

authorized super administrator. After loading of the database, local administrators have the ability to start the voting process.

Once started, voters undergo a verification process. They place their finger on the designated sensor and the biometric impression is stored in the memory of Arduino temporarily. The biometric fingerprint templates that are stored in the Android device are then passed to the Arduino, which places them in the memory of the sensor. Finally, the live biometric impression and the stored fingerprints are matched to verify the existence of the fingerprint record, before it is cross-referenced against the database to make the final call.

After verification is completed without any problems, the actual voting begins. The voter cast their respective votes by selecting from the available candidates and long press on the name. They are prompted by the system to make sure that the vote they case is legitimate and accurate to their wishes. Once it's confirmed, voters see their ballot information for a brief moment.

The final step occurs after voting has been done. After the ballot information is shown, the printer at the end terminal is issued the print command, which then prints out a physical copy of the ballot. The ballot will contain the name of the candidate who he/she voted for, the party of the candidate and its symbol, along with a timestamp of the occurrence of the vote, as well a randomized system generated number that will act as an additional security measure. After this is done, the voting process is thought to be completed successfully and the next voter can start the process all over again.

Furthermore, the administrators and the super administrator can log in to the system by verifying their fingerprint at any point of time and they are provided access to the administrative panel upon verification of their fingerprint impression. The administrators are provided the option to start or stop the voting process, while also being able to display the current vote count. The super administrator has the extra privilege of adding and removing voter, as well as load new database into the system, the latter of which is to be used in case of an update. The administrator can be a voter as well in special cases, in which he or she will have the opportunity either of the panel, depending on their status - both panels if he or she is yet to cast his or her vote.

3.2 User Groups

The users of this system can be broadly classified into two categories; administrators and general voters. The administrators can be divided into two subclasses as well - general administrators and super administrator.

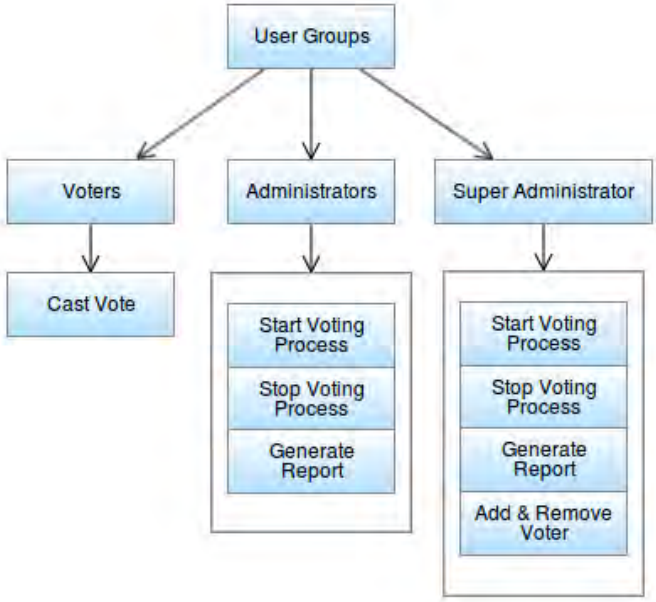

Figure 3.2: System User Groups

The reason why the administrator user group has been divided is to ensure the integrity of the system to the highest level. While the administrators as a whole are capable of making actions that can control the voting process, the options of adding new voters or removing existing voters are bestowed upon the super administrators, who are usually not present during the voting process, in the voting center. The system application takes the fingerprint of the user, verify whether the user is an administrator or a voter by matching the prints against the database entries stored in the Android device, and then open the appropriate panel. If the values of the fingerprints do not match, access is denied to the system. The administrative panel provided to the administrators will have the provision to either access the administrative panel, or when the

voting process is in effect, will display an option to vote as well, if the administrator is assigned to the voting center in context. In the administrative panel, administrators will have features such as activating or deactivating the voting process, and to generate vote count report. The super administrator will have a pair of extra options to his or her disposal, with them being the provision to add new voters or removing existing voters. The process of activating or deactivating the voting process cannot be done by a single administrator. At least three members of the administrative panel, consisting of a group of either general administrators or super administrator, have to authenticate in order to start or stop the voting process. This adds a layer of security and makes it extremely difficult for rigging the voting process.

The voters in turn will have access to the ballot upon fingerprint submission where they can cast their vote. The only function they can perform on this system is to cast their vote. Other than that, they have no authority to modify or make changes to their information or manipulate their cast votes. This vote is stored electronically in the database and the voter can obtain a physical copy of ballot paper with the help of a thermal printer, which is coupled with the Android device, at the end of the voting process. This physical copy is then dropped in a physical ballot box by the voter, which will add a final level of credibility and legitimacy to our system.

3.3 Hardware Specification

In the hardware layer, careful thought has been put in the planning stage of development in order to ensure minimal hardware to be used in the system, so that hardware redundancies are minimized, as well as reducing the complexity of operating the system, to facilitate usage by normal users, as opposed to highly specialized ones. All the hardware subsystems used in the system communicates through wired serial interface, so that security hacks that can occur with wireless connection is eradicated.

The hardware used for the system is as follows:

1) Fingerprint Scanner

The fingerprint scanner used for the system is TTL-(GT-511C3)[10]. This model has been chosen for reliability and for cost considerations, and it's capability of running verification algorithm within itself, eradicating the need of exhausting fingerprint matching algorithm to be implemented and run on the scarce memory of the Android device. This scanner will be attached to the Arduino board using a JST SH Jumper 4 Wire Assembly[11].

2) Arduino

This system uses the Arduino Mega2560[12], which is a high-end microcontroller unit in comparison to most other similar boards, and has been chosen so that handling large amounts of data is not an issue, as it has a fairly large RAM. It is programmable with the bundle Arduino software, and it contains multiple hardware UART serial interfaces by which maintaining constant communication between the Android phone, the fingerprint sensor and the thermal printer throughout the runtime of the voting process becomes convenient and reliable. In our proposed Smart Voting System, this Arduino Mega2560 microcontroller acts as the base platform for executing custom developed device driver programs in order to control both the fingerprint sensor and the thermal printer throughout the entire voting process. The device driver programs are solely responsible for controlling the hardware devices and executing low-level hardware specific routines. These custom device drivers running on the Arduino Mega2560 microcontroller are designed specifically to suit the needs of our proposed voting system.

3) Android Phone

Any standard smart phone that operates on the Android Operating System version 4.1 or above will suffice. This Android phone should have a minimum of 512 MB of RAM storage space and must offer USB-On-The-Go facility. User database will be stored inside the internal memory of the phone by the super administrator. When a live fingerprint data from the Arduino board is received, the phone will forward each and every biometric fingerprint template stored on its database to the sensor via the device driver running on the Arduino, which will instruct the

fingerprint sensor to run the verification algorithm against the print that has been captured from the authorization process. If the match result is positive, a response is sent to the phone by the device driver notifying that authorization has been successfully completed and voting can ensue. If the result is negative, the response sent to the phone is an error message and the voting process halts indefinitely.

4) Thermal Printer

The thermal printer is tasked with the job of producing the final physical, tangible print that will act as the real evidence of the vote being cast and its consequent credibility. The printer used is a highly specialized, low-cost, small-sized one that will print a single-bit raster image of the symbol of the voting party in question[13] and that supports a standardized print command set by which the printer will be programmed. The printer runs at a power supply of 9 volts and 1.5 ampere, which is provided through an external adapter connected to the Arduino that bridges between the components and the Android device. The printer supports ESC/POS[18] page description language which is used to program the printer for ballot paper generation

CHAPTER 4

System Implementation

Our Smart Voting System comprises of two layers; hardware layer and software layer. Both layers act cohesively and in conjunction to produce a complete system that is much less work intensive and more efficient than other exclusively hardware-based alternatives. The voting system is mainly dependent on the software layer consisting of relevant libraries, namely, the library that is being used in the application inside of the Android device to drive the whole voting process and the hardware library in the form of the Arduino sketch that is needed to communicate with the connected fingerprint scanner to verify and enroll the users, and the thermal printer to print the physical ballot. The database stored inside the Android device also plays a vital part in the proper functioning of the whole process. In the hardware layer, the Arduino is interfaced with the fingerprint sensor, which serves as the starting point, and the input fingerprint data then traverses through the micro controller towards other peripherals, that is, firstly, the Android device for verifying the templates against the database and, once that's done, vote information is finally sent to the printer which is connected as an output terminal with the Arduino that produces the ballot paper.

4.1 System Integration

In the hardware layer, only the most essential components have been chosen in order for us to fulfill the criteria of voting, and all these components used in the system are interconnected by serial (TTL) interface wire connections, so that the possibility of security hacks with wireless connection is eradicated. Moreover, our system utilizes a fingerprint scanner which can convert user's biometric fingerprint impression to proprietary binary template, which cannot be used with any other scanners, allowing us to ensure better level of performance and much more reliability of the system, over other fingerprint verification processes, such as minutiae extraction[14].

However, the limitation of the scanner we used was that it can only store 200 fingerprint templates in its physical memory. This is impractical when it comes to an application which is as

vast as an election process of a country, in which thousands of voters will participate in choosing their representative in the parliament. This issue was addressed through the use of our Android application, where the templates are stored in a database, in the internal memory of the device, and are uploaded into the scanner when required through serial communication, addressing the issue of the limited number of templates that can be stored.

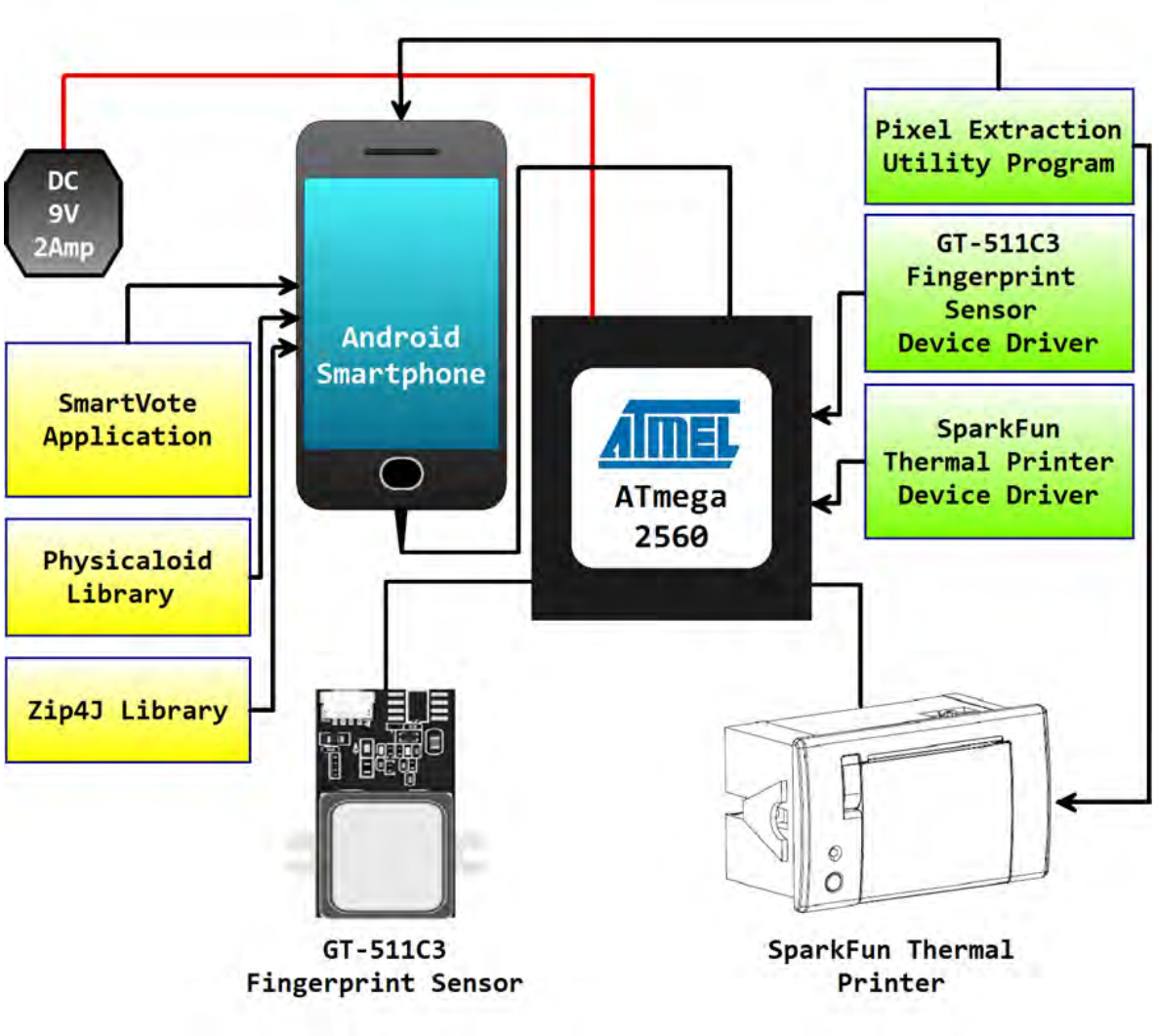

Figure 4.1: Association of Hardware and Software in Smart Voting System

The Android phone, on which the application software is running, is connected to a central Arduino microcontroller unit, which is running the core device driver software in order to allow efficient and convenient device access. The Arduino microcontroller together with device drivers provides a uniform and well-understood programmatic interface to the Android host application.

The fingerprint scanner, as well as the thermal printer, is connected to the Arduino board and the board is responsible for driving the hardware components properly.

When a voter comes in to cast his or her vote, he or she has to enter their fingerprint for biometric verification. The fingerprint device driver instructs the fingerprint sensor to scan for the fingerprint image. As the fingerprint sensor gets all the required instructions from the fingerprint device driver, it starts scanning for the fingerprint, converts it to a binary biometric template and notifies the respective device driver. The fingerprint device driver then asks for the scanned biometric template, receives the template data from the sensor and then stores it to a known memory location of Arduino. The Android phone then transfers the existing templates that are stored in its database to the Arduino microcontroller, and the fingerprint driver receives those templates and set them in the fingerprint scanner, where the verification process will be performed. The templates that are transferred are then matched against the recorded template to verify the user.

Furthermore, when a voter is added to the database, with the guidance from the device driver the fingerprint scanner starts to enroll the voter's biometric information by taking the fingerprint impression and then convert it to a binary biometric template. This template is then transferred to the Android device through Arduino microcontroller and is stored in the database. All the templates that are thus generated are all unique for each voter, and the size of each template file is 498 bytes, which ensures that storage inside of the internal memory of the smart phone is not an issue. It should be noted that each and every attempt to access the devices made by the Android voting application is carried out by the specific device drivers programmed on the Arduino microcontroller. It is neither practical nor possible for the Android application to access the devices directly.

The system also includes a thermal printer, which is very compact in size and utilizes TTL serial protocol and exposes ESC/POS[18] command set, making it easier for us to interface it with our existing system. This printer is used to obtain a physical ballot paper after the user has casted his or her vote, such to ensure the credibility of our system, with a physical evidence. Our system prints the single-bit raster bitmap image of the entity symbol at the end of the voting process, along with a timestamp and a randomized number, and this physical copy will be placed

in a physical ballot box, which can be cross referenced with the data stored in the Android device, at the end of the voting process.

4.2 Device Drivers and Device Operations

Our proposed Smart Voting System is built on top of two essential hardware devices; one is a biometric fingerprint scanner and the other is a thermal printer. For the system to perform correctly and efficiently, the system must control and handle the devices in the proper way. Both of the devices we use in our proposed system communicate through serial interface and TTL protocol, and they expose very low-level programming interface to the host applications that uses the devices. It will be very convenient if a high-level task oriented programming interface could be offered to the Android voting application, by which interaction between the devices and Android voting application will become much more reliable and controllable. To make this bridge between the Android and the hardware devices, we have developed custom device driver software, which will handle the devices effectively and efficiently, and at the same time offer a high-level device API to the Android.

The system requires separate device drivers for each device in order to effectively operate two separate devices from the voting application. Therefore, we have developed device drivers for both the GT-511C3 fingerprint scanner and the SparkFun CSN-A2-T-0 thermal printer. Both of these device drivers are written in C and designed to run on an Arduino Mega2560 microcontroller. Data communication between the Android voting application and the device drivers is accomplished by utilizing a simple message-passing paradigm between hardware subsystems, which is easily understandable, maintainable and extensible.

Each message is realized to be a single byte value, which is unique among other message and defines a specific operation or command to perform. Each message is well-understood between the Android voting application and the device drivers. There are several well-defined messages on our voting system; each of them is used to control either the fingerprint sensor or the thermal printer in a task-oriented way.

As an example, to scan for the biometric fingerprint impression of a voter, the Android voting application sends a particular message to the Arduino microcontroller through serial connection. On the Arduino microcontroller, device drivers are running which are programmed to look for incoming messages over and over again. As the Android application sends a message to the microcontroller, the fingerprint device driver receives this message and selects a particular device-dependent procedure which is mapped from the received message. The device driver of the fingerprint sensor (GT-511C3) then starts the procedure; in this case a hardware-specific '**devGT_ScanLiveTemplate ()'** procedure implemented on the device driver software, starts interacting with the fingerprint sensor device back and forth, and at the end obtains data from the fingerprint sensor and sends the result back to the Android voting application. Every hardware and device-dependent task is handled by programming and utilizing separate device drivers, therefore the Arduino microcontroller and the associated device drivers for the fingerprint sensor and the thermal printer plays a very critical role in our voting system.

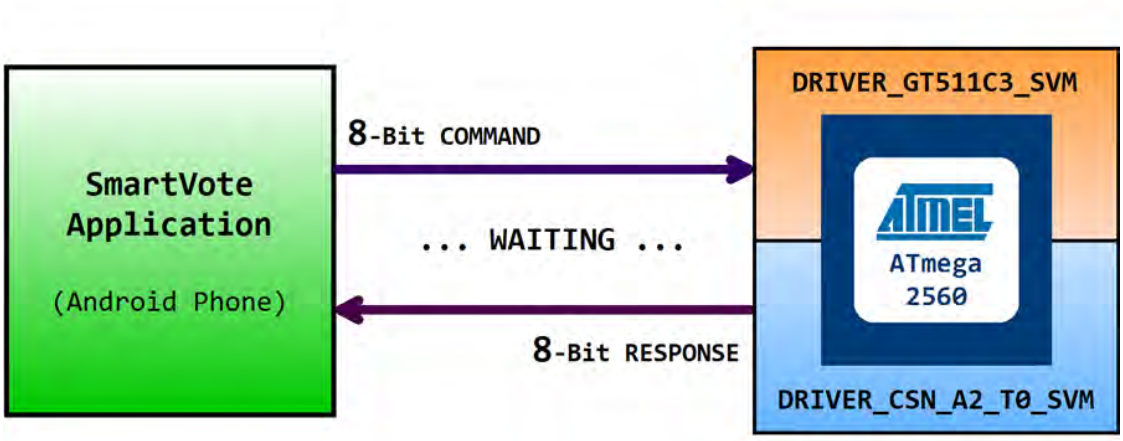

Figure 4.2: Message-passing paradigm between subsystems for executing device-dependent operations

Another example that exploits an important role played by our device drivers in the Smart Voting System is the generation of the physical ballot paper at the end of vote casting. To print a ballot paper by the thermal printer, it is necessary to firstly communicate with the printer in the way it expects. Our custom printer driver manages each and every detail of handling the printer device by exposing appropriate ESC/POS[18] language to the printer, which is the native

language to modern thermal printers and as well as printable application data. Details like what will be the size of the printer font, how the text will be printed, alignment of text and images regarded to the supplied paper; numbers of line feeds to be applied etc and much more are managed directly by our custom printer driver, running on the Arduino microcontroller. The Android voting application will only need to send the text and image data to the printer driver, and the driver will handle all the details of generating the complete ballot paper as well as other printer-related operations.

Using these device drivers to control and interact with the hardware leads our proposed voting system closer to become portable and hardware-independent. When porting our Android voting application to a new operating system or to a new platform, it will be sufficient to modify or rewrite the voting application only for the new environment. Device drivers run on microcontroller module which is separate from the application environment and does not need to be changed by any means. Similarly, if the current fingerprint sensor or the thermal printer needs to be replaced by a different hardware module, only the respective device driver should be created with the high-level driver interface left unchanged, and our voting system will be able to work with the new hardware systems as it works with the current hardware set.

4.3 Libraries

The Smart Voting System makes use of a number of software tools and libraries currently available, which are then used together to form the basis of the software layer that has been described in the system, including the application that runs the whole process.

Unfortunately, the Android Operating System lacks support for serial communication and the Android phones those are available in the market does not have a hardware serial interface. Developers of the Android Operating System and their related organizations have never felt the necessity of including a native serial interface as well as native serial communication API built on the Android systems, partially because it is a mobile phone and unlike full-featured computer systems, mobile phones are designed for different purposes and are targeted to be used on different tasks. Moreover, serial communication is a very old hardware communication protocol,

and today's mobile phones have built-in functionality for modern communication protocols like USB and wireless Bluetooth.

The Smart Vote Android application that we have developed borrows heavily from a previously existing software library known as Physicaloid Library that provides serial communication facilities between an Android device and any other device which is capable of serial communication[15]. The Physicaloid Library comes with native support for Android Operating System, which makes the integration easier. In our case, the Android device is serially connected to the Arduino microcontroller, which is also connected serially with the thermal printer, and so the customized Physicaloid Library helps us to establish and maintain the serial connection between all these devices.

In addition, we wanted to keep the voter database inside of a zip file that will be password protected, so that all the sensitive information is kept out of reach of the wrong hands. In order to handle this password-protected voter data zip file, we use a library called Zip4J, as a result of which we can load the database onto the phone without any security worries[16].

Furthermore, we made use of Apache Commons IO file utility, with which we've accomplished the easy management of files throughout the application.

4.4 Database

The database intended for this system has been developed using SQLite (version 3). It also has native support for Android Operating System, which means that it is convenient to bridge the application and the database. Careful thought has been put in its development, so that it uses as less space as possible. This is a critical requirement as the Android device has a relatively limited capacity for processing and storing data.

candidate_name	entity_name	entity_symbol	vote_count

Figure 4.3: Database Schema for the Entity Table

The database mainly consists of three main parts. The entity, which essentially contains the candidate information, the name of the party he or she represents, the symbol of that party, and so on. For this purpose, attributes named candidate_name, entity_name, entity_symbol are created respectively that represents these values in the entity table. The vote_count attribute keeps count of the number of votes cast in favor of the candidate in question. In addition, there is also the user table that stores information about individual voters. Relevant information includes the voter's name, address, date of birth, his/her parent's name, etc. Finally, there are boolean attributes that categorize users into certain groups. Three of these, namely, isVoter, isAdmin, and isSAdmin, determine the classification of the user from their value combination. For example, if a user is a voter, the boolean attributes assigns a value of 1 to his isVoter attribute, and vice versa.

An exceptional case involves the user being and administrator as well as a voter, and the value assigned to the respective fields are 1 and 1 respectively. There is a fourth Boolean variable named as hasVoted, which keeps count of whether a user has already voted or not. If so, the value assigned to it is 1, and if no previous vote history exists, the value assigned to it remains 0.

4.5 Image Pre-processing and Printing Candidate Symbols

Most of the thermal printers, including the one we use in our proposed system, adhere strictly to the ESC/POS[18] printing specification, and therefore not capable of printing bitmap formatted images without some level of image pre-processing applied on the image. ESC/POS[18] is a standardized Page Description Language developed at Epson Corporation, and it is targeted to bring every thermal printer under a standard set of command set, in order to simplify the task of programming thermal printers.

ESC/POS specifies a single-bit raster bitmap format, where the image to be printed will only contain each pixel values as a single bit. This specification is neither compatible with the popular Microsoft Bitmap (BMP)[19] format nor with any other well-known image formats, as those image formats store images with lots of header data[19], pads scan lines with extra bits[19], or sometimes store the whole pixel array upside down.

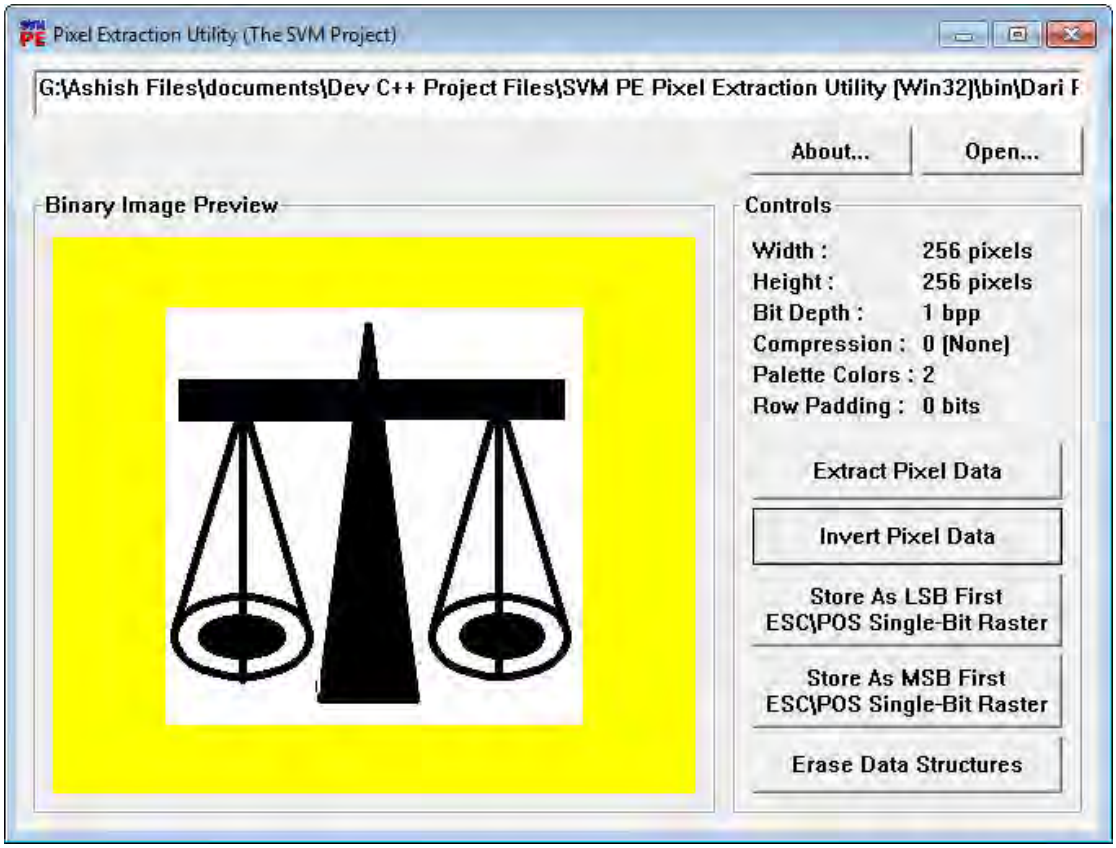

Figure 4.4: Pixel Extraction Utility program converting a candidate symbol to an ESC/POS printer compatible Single-Bit Raster image

To address this barrier and to make our system capable of printing bitmap images, we have developed an image conversion utility program, by which a Monochrome bitmap formatted image can be transformed into an ESC/POS[18] specified single-bit raster image. This image conversion program reads a user specified monochrome bitmap image, remove extra information

associated with the image, for example headers and extra padded bits, then extracts the actual pixel array from the image and optionally invert the pixel array if the image is stored upside down. The converted single-bit raster image is then exported to the database of the Android voting application as a candidate image, and this exported image is printed on the ballot paper by simply sending it directly to our custom printer driver software.

This pixel extractor program is developed as part of our proposed system, and designed to be a utility program, separated from the Android voting application in order to simplify the task of printing candidate images on the physical ballot paper. With this pixel extractor program, once an image is converted to ESC/POS compatible raster and exported to the Android voting application database, that copy of the image can be used over and over again by the Android application for printing. There is absolutely no necessity of running any image pre-processing algorithm every time a candidate image needs to be printed, as the candidate image does not change and remains static throughout the whole voting process. This results in an increase in system performance, as the overhead of pre-processing candidate images is removed from the Android voting application.

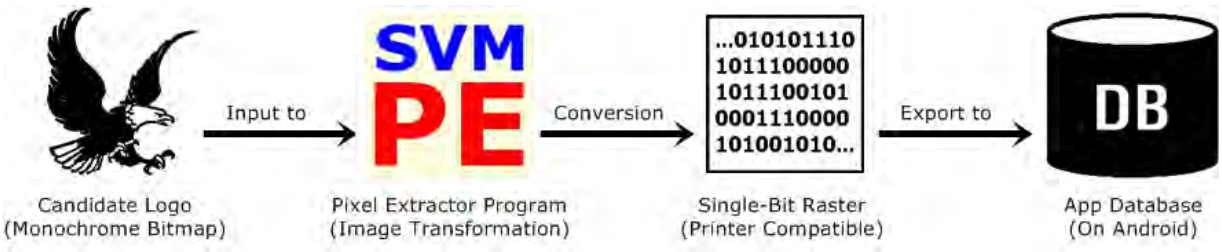

Figure 4.5: Process of creating printer compatible images from Monochrome Bitmaps

Moreover, utilizing this pixel extraction utility program to create printable versions of candidate symbols relieves our Android voting application from becoming hardware-specific. In the future, if some other kind of printer is being used with our voting system and those printers expect images with a different format, the replacement of old printable image files with the new

one is only required. It is not necessary to modify the voting application to work with new printer, and it helps our voting system to be hardware-independent.

4.6 System Flow

The information flow of the system is determined and regulated by the customized voting application that has been developed to be deployed in the Android smart phone being used in the system. The basic rundown of how the system works and the information flow on the software level determined by the application is as described below.

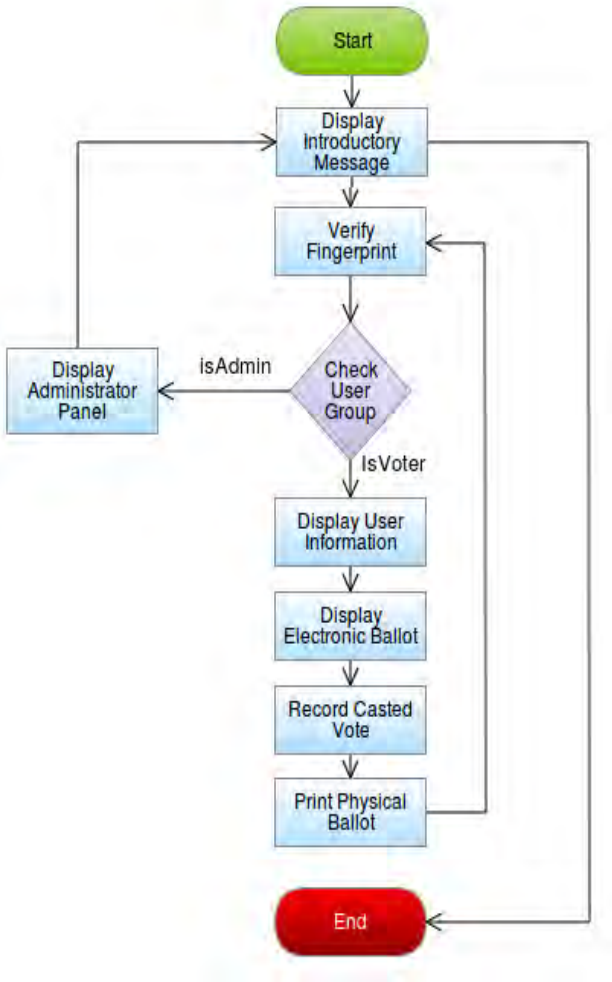

Figure 4.6: Flowchart of System Application

Initially, when the application is run on the Android device, the user - in this case, the Super Administrator - is greeted with a Splash screen, behind which the connection with Arduino is established and checking is done regarding whether the voter database exists in the device or not. If a connection with Arduino exists, the user is prompted for permission on the first run. This is done to ensure that the bridge between the Android device and the Arduino is a success.

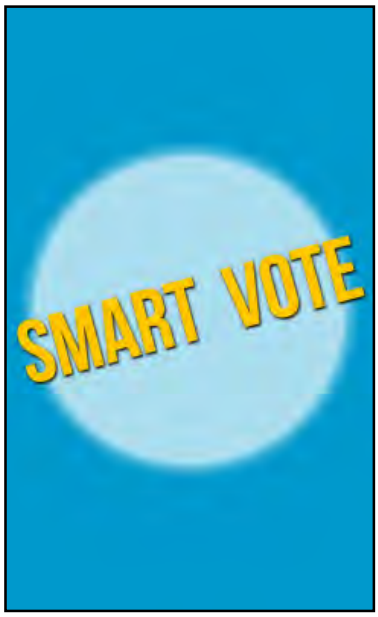

Figure 4.7: Splash Screen of the Application

If the files are not found (usually in the first run and if the database is tampered with), then the user is redirected to the load database screen, where the user is required to provide the path of a password-protected zip file, along with a password. Once the Super Administrator presses the "Load Database" button, he has to verify his fingerprint through the scanner and proceed to the next step.

Figure 4.8: Load Database Screen

If the super administrator is verified, he is directed to the administrator panel, where he may choose whether to start or stop the voting process, or add or remove voters from the database, and generate reports based on the poll.

However, if the files are already available in the application's data directory, then the user is greeted with the welcome screen, where they are given a brief instruction about the application.

Figure 4.9: Introduction Screen

Once the user presses the "Next" button, they are redirected to a screen where the instruction for placing fingerprint on the scanner is illustrated, as well as provided in text. The fingerprint scanner lights up to scan user's fingerprint on this step.

Figure 4.10: Fingerprint Verification Screen

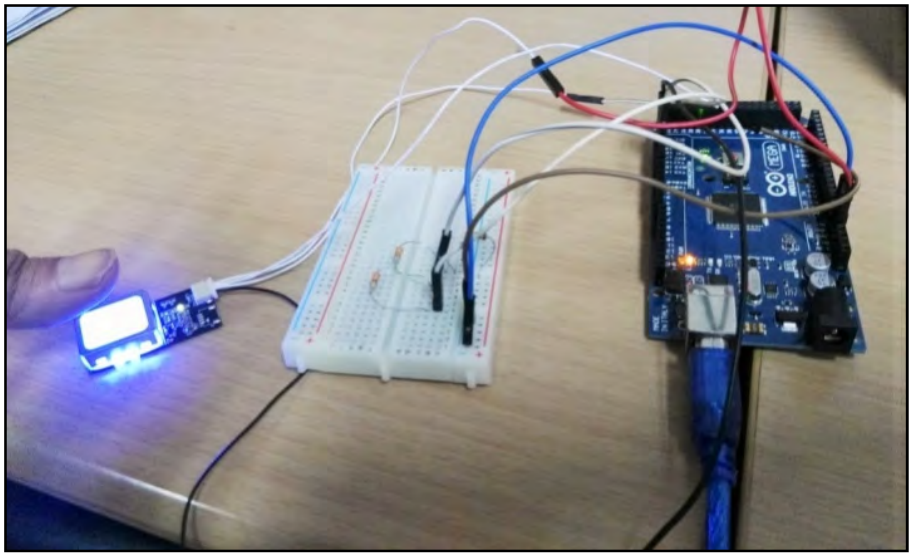

Figure 4.11: Live Fingerprint Scanning Process

If the user's fingerprint matches one of the entries in the database, then the user is shown a confirmation window, with his or her information. This screen lasts only for ten seconds, before the user is redirected to the eBallot screen.

Figure 4.12: User Details Screen

The voter then finds the eBallot screen, where they can vote for their desired candidate. The list provides the voter with candidate name and entity symbol (not displayed in the current revision) and they may cast their vote by long pressing the item on the list. A confirmation dialog is displayed to validate the user's selection.

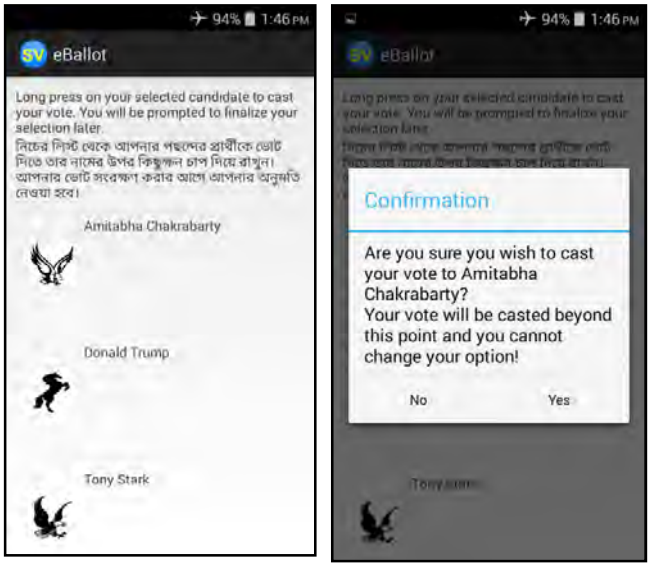

Figure 4.13a and 4.13b: The eBallot Screen, Along with the Confirmation Dialog

Upon casting vote, the user will be displayed a success screen, where the printer will print the image of the entity, along with a timestamp and a system-generated, unique random number.

Figure 4.14: Success Screen

The application will automatically move to the introduction screen following the success message, to allow the next voter to begin their voting process.

Furthermore, the super administrator is able to enroll new users in the database. In such cases, the voter is asked to complete a form, with the help of the administrator.

Figure 4.15: New Voter Screen

CHAPTER 5

Experimental Results

5.1 Analysis and Results

Our primary vision while developing the system was to ensure security first, followed by user friendliness, and it can be said that we have managed to work it out to perfection, making our system as secured as possible, while not making the trade off that would see our system to be rendered difficult for the users to use. We ensured the highest level of security by choosing a fingerprint sensor which is capable of matching binary templates with live fingerprint. This ensures that the fingerprint data that are stored in the Android device for future verification cannot be tampered by any means. Furthermore, the process of identification is done in the scanner itself, which allows the system to be more secured and transparent, since the algorithm used for matching is provided by the vendor and is unique.

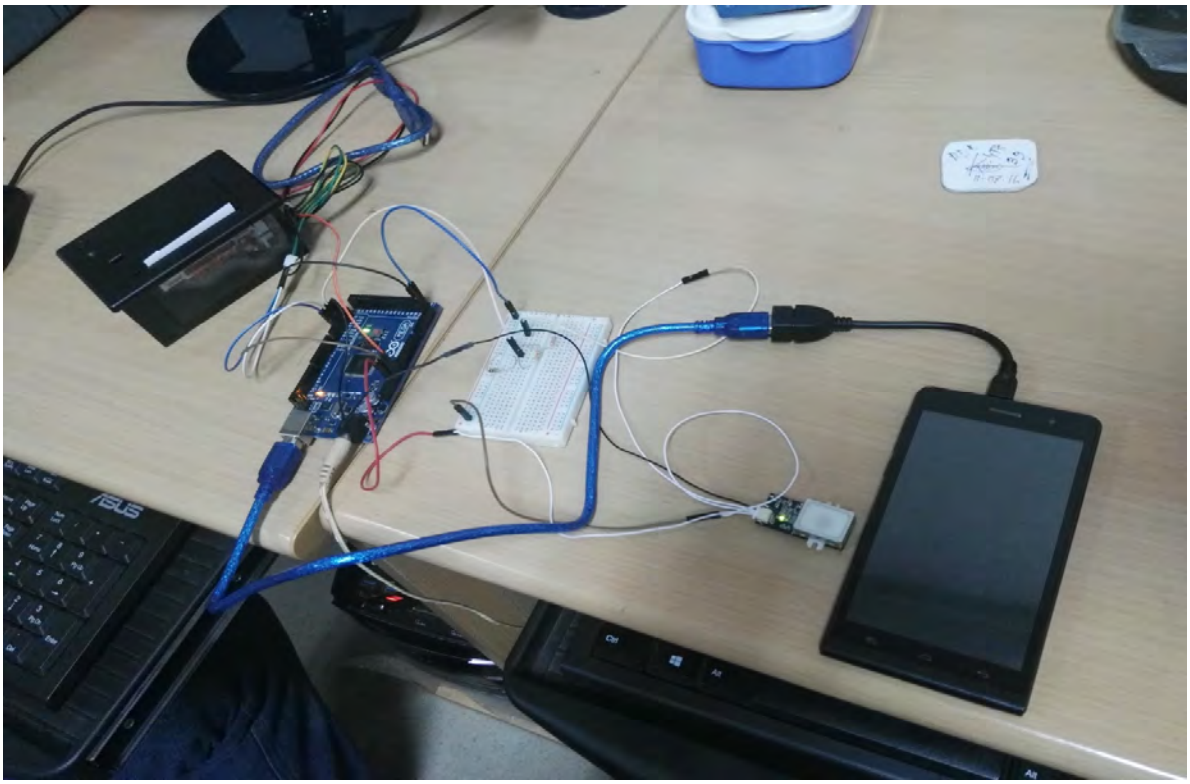

Figure 5.1: The Complete Smart Voting System

One of our primary goals while making the system secured was to ensure that the fingerprint verification do not result identification that are wrong, that is, not detecting a different user for the user in context. It can be said after our development that we have managed to ensure that our system is full proof on that front and does not detect fingerprint improperly, ensuring absolutely zero percent False Acceptance Rate (FAR) and 15-20 percent False Recognition Rate (FRR)[16]. However, our system may fail to identify fingerprints altogether, due to various issues, such as a dirty fingerprint impression or scanner surface, problem with the library, etc. However, we have found that the probability of such an event taking place is very low, almost zero.

Our system was found to be competent of distinguishing between various user groups, such as the voters, the administrators and the super administrator. We have tested our application several times to check if the identification process works up to the mark and we have found that our system works exceedingly well to verify the identity of the user, failing altogether if the system reaches a false rejection scenario.

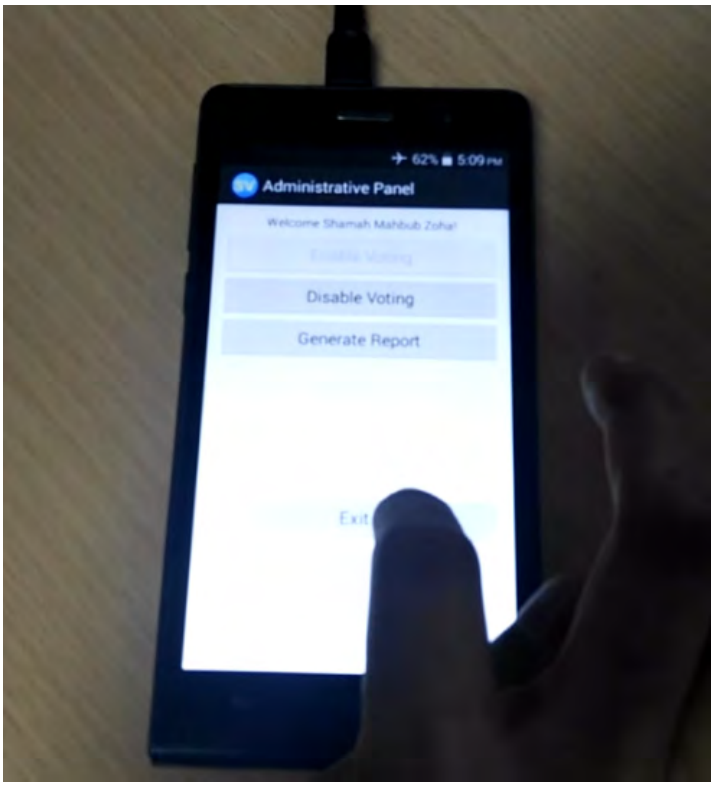

Figure 5.2: The Administrative Panel (Voting Enabled)

Figure 5.3: Screen Showing Voter Information

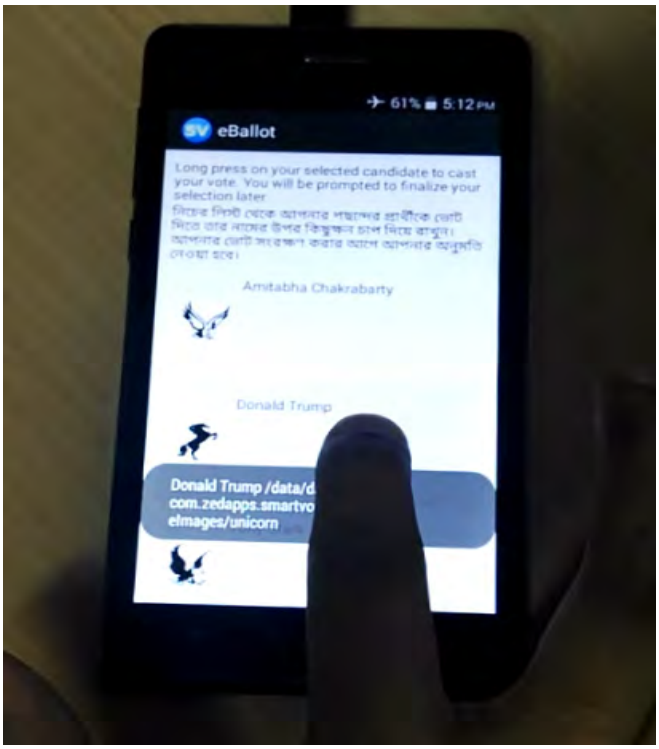

Figure 5.4: eBallot Screen

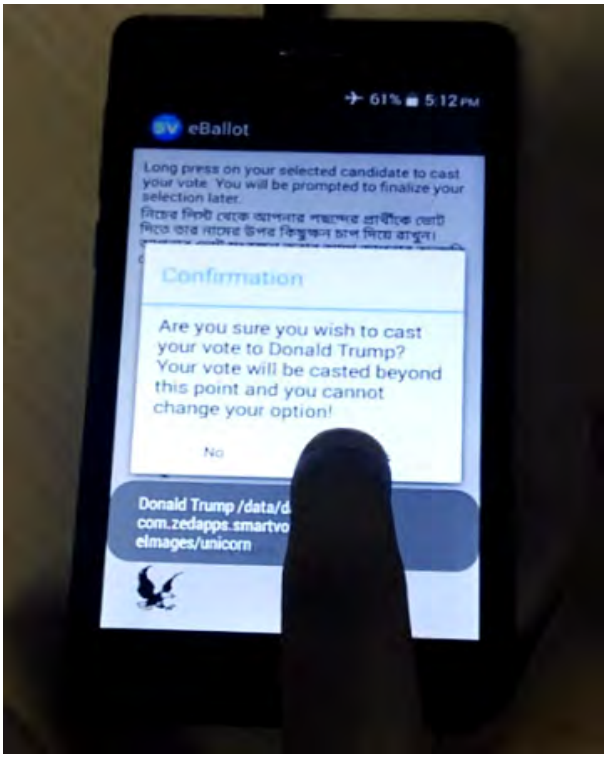

Figure 5.5: User Prompted To Verify Vote

The thermal printer was observed to work according to our need, printing timestamp and entity image to perfection when required, along with the randomized number, which we have added for security. However, it should be noted that this randomized number is only unique in a single voting center, since our application cannot synchronize with other devices to ensure a perfect unique value. This, however, could not be overcome, since we have decided to disable any kind of wireless communication for the device, to keep our security level to maximum.

Figure 5.6: The Physical Ballot

5.2 Discussion

Throughout the development of our thesis, we've encountered various issues that needed to be addressed. The degree of difficulty presented depended mainly on the materials that we worked with, both hardware and software related.

Initially, our plan was to take the fingerprints directly from the Android device. However, that proved to be an impossible task, as the fingerprint API for Android was proprietary and therefore could not be accessed. At the same time, there was the problem of low-end or older Android devices that did not support fingerprint API of the later Android versions. To maneuver around that situation, we intended to customize the Operating System so that it would support the features we wanted to include in the system. However, that proved too complicated and so that concept had to be abandoned in favor of a more feasible alternative. Therefore, we included a

fingerprint scanner and an Arduino board, which helped us obtain the fingerprints for verification purposes of the voting process we were aiming for.

Moreover, we had to program the GT-511C3 fingerprint sensor and SparkFun CSN-A2-T-0 thermal printer devices in order to use them form the Android voting application. Existing code base by the open-source community seemed to be inappropriate and in many cases, incomplete and totally unacceptable for our system. As a result, we had to develop our own device driver software for the GT-511C3 fingerprint sensor and the SparkFun Thermal printer. We had learned how to program those devices from the User Manuals provided by the device manufacturer, example codes and the open-source community.

Also, the Physicaloid Library is substantially limited. It had not undergone updates in a long time, and contains numerous bugs and glitches that hamper functionality and dependability as a whole. During the development of the system, these existing bugs proved to be a detrimental factor and, therefore, we had to work our way to figure out bypassing these bugs that would help us in achieving our goals. To work around the issue, we had to use a dated version of the library to ensure that the level of error from this application is minimal, as the library was found to provide insufficient support with the latest Android Operating Systems. After thorough debugging, most of the issues were dealt with, while more work needs to be done to remove others, in order to add robustness to it and more support for functionalities in order to fully optimize the efficiency and reliability of the system.

CHAPTER 6

Conclusion

As we have seen, our Smart Voting System offers a dynamic and robust solution to the current voting scenario that is in accordance to modern practices and technological advancement. Having said that, there is scope to further develop the system into an even stronger unit, while also making it more accessible and easing large-scale use. Some of the issues and works that can be done of the currently proposed system, and concluding remarks are presented below.

6.1 Limitations

The Smart Voting System builds upon the concept of electronic voting that was achieved with the introduction of EVMs, and broadens its scope of practical application by addressing some critical issues that remained in their hardware-based predecessors, essentially meeting many more of the previously unfulfilled modern voting criteria. However, there are obvious limitations regarding the current system that has been developed to fully actualize the vision and purpose for its existence.

Firstly, the Physicaloid library that has been used is provided by a third party source. It had not undergone updates in a long time and thus does not support several of the modern tweaks that have been added to the latest revisions of the Android Operating System. As a result, it contains numerous bugs and glitches that cause issues related to unreliability, as occasional crashes make the user experience less desirable. After thoroughly debugging problematic areas, most of these

issues were successfully resolved. However, some of these issues are still prevalent as they are highly persistent, and lie deep within the architecture of the library. These issues could not be resolved and their presence may hamper the user experience on occasions. More work needs to be done in order to provide users with a less problematic experience by removing all the remaining bugs and glitches that plagues the usability of the system.

In addition, some of the features of this system perform slowly, as the response time of those affected features is high, and as such is not ideal for practical purposes. The fingerprint verification process takes a considerable length of time since the fingerprint binary templates have to be sent to the fingerprint scanner for verification and goes through the communication bridge of the Arduino, which responds slowly in transferring the data. Similarly, the same can be said about the printing process of the physical ballot. The size of the image file containing the symbol of the entity represented by the candidate, though not causing memory constraints for the Android device, proves to be an issue as when transmission of the image data to the printer over Arduino is markedly slow. Thus, it takes quite a bit of time for the physical ballot to finish printing, which is not ideal for real life use. We plan to overcome this challenge in our future versions.

Furthermore, this system has not been tested on a large-scale scenario. Therefore, it is not easy to cannot forecast how the system to going to react to events that would include more resources, such as a large database, responding to large turnout of voters, etc. It is likely that it may take a lot of time processing out voters within a manageable time and so may be lacking on a practical level. Other factors that may come into play are providing 9V supply to the system in venues where such sockets are not available.

Finally, some of the components that have been used for the thesis are highly specialized and customized, such as the fingerprint scanner used, as well as the printer, which may not always be available on the market. This may act as a detriment for large-scale implementation of the system as its use is directly linked to the availability of these devices.

6.2 Future Works

The system that has been developed in its current version can still be improved further, and there is scope for more features to be added which can ensure that the Smart Voting System is the only feasible way of conducting the voting process. For starters, we intend to make this system compatible for devices that do not run in Android operating system. To facilitate and encourage continued use of this system across various platforms, we intend to make it available for devices running on Windows, as well as Apple devices running on iOS and also on the Firefox OS. This will make the implementation less device-specific and also lower costs in the long run, as more and more generic devices, such as commonly used household printers and other devices, are used for this purpose.

6.3 Conclusion

This paper proposes a smart voting system that is reliable, cost-effective, secured and efficient enough to be used in various practical scenarios where an impartial voting system is required. This system also offers an ideal solution to those voting scenarios where the number of voters is very high, and where the large-scale voter data renders manually inputting the information next to impossible.

With addition to this, our proposed system also offers a biometric voter verification system, by which security and reliability is guaranteed to an acceptable level. In conjunction to the biometric authorization, the usage of thermal printer brings transparency and reliability to the overall voting process.

REFERENCES

[1] Anis, M. A., Rahman, H., Alam, J. S., Nabil S. I. and Hasan, S. M. 2014 Development of Electronic Voting Machine with the Inclusion of Near Field Communication ID Cards, Biometric Fingerprint Sensor and POS Printer: http://dspace.bracu.ac.bd/bitstream/handle/10361/3967/ThesisReportFinalv1.pdf?sequence=1

[2] The Importance of Voting in a Truly Democratic Society. 2010: http://vibeghana.com/2011/01/28/the-importance-of-voting-in-a-truly-democratic-society/

[3] Jones, D.W. 2001. Problems with Voting Systems and the Applicable Standards: http://homepage.cs.uiowa.edu/~jones/voting/congress.html

[4] Ford, M. 2014. Indian Democracy Runs on Briefcase-Sized Voting Machines: http://www.theatlantic.com/international/archive/2014/04/indian-democracy-runs-on-briefcase-sized-voting-machines/360554/

[5] International IDEA , Voter turnout data for Bangladesh: http://www.idea.int/vt/countryview.cfm?CountryCode=BD

[6] Smart National ID Cards: http://hifipublic.com/2015/01/21/smart-national-id-cards-for-voters-from-march/

[7] Digital Signature on Smart ID Cards: http://bdnews24.com/bangladesh/2016/02/26/digital-signature-to-be-on-smart-nid-cards-soon

[8] EVM Vote Manipulation: https://www.statslife.org.uk/politics/2288-how-trustworthy-are-electronic-voting-systems-in-the-us

[9] 2004 US Election Vote Rigging Controversy: https://en.wikipedia.org/wiki/2004_United_States_election_voting_controversies#Voting_machines

[10] ADH Tech GT-511C3 Fingerprint Scanner: http://www.adh-tech.com.tw/?22,gt-511c3-gt-511c5-%28uart%29

[11] JST SH Jumper 4 Wire Assembly: https://www.sparkfun.com/products/10359

[12] Arduino Mega 2560: https://www.arduino.cc/en/Main/ArduinoBoardMega2560

[13] Sparkfun Thermal Printer (CSN-A2-T-0): https://www.sparkfun.com/products/10438

[14] Mazumdar, S. and Dhulipala, V. "Biometric Security Using Finger Print Recognition", University of California, San Diego.

[15] Physicaloid Library: https://github.com/ksksue/PhysicaloidLibrary

[16] Zip4J: http://www.lingala.net/zip4j/

[17] Thakkar, D. 2016 "False Acceptance Rate (FAR) and False Recognition Rate (FRR) in Biometrics": "https://www.bayometric.com/false-acceptance-rate-far-false-recognition-rate-frr/"

[18] ESC/POS at Epson: content.epson.de/fileadmin/content/files/RSD/downloads/escpos.pdf

[19] Microsoft Windows Bitmap: Summary from the Encyclopedia of Graphics File Formats “http://www.fileformat.info/format/bmp/egff.htm”