VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

"Android Based Online Voting System"

Thesis submitted in partial fulfillment of the curriculum prescribed for the award of the degree of Bachelor of Engineering in Computer Science & Engineering by

1CR14CS126	Santosha Bhat
1CR14CS154	Vageesh K C
1CR14CS162	Vishwaradhya
1CR14CS167	Yashwanth R K

Under the Guidance of

Mr. Shivaraj V B
Assistant Professor
Department of CSE, CMRIT, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
#132, AECS LAYOUT, IT PARK ROAD, BENGALURU - 560037

2017-18

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

CELEBRATING 25 YEARS
CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Certificate

This is to certify that the project entitled **"ANDROID BASED ONLINE VOTING SYSTEM"** is a bonafide work carried out by **Santosha Bhat** bearing **USN:1CR14CS126** , **Vageesh K C** bearing **USN:1CR14CS154**, **Vishwaradhya** bearing **USN:1CR14CS162** and **Yashwanth R K** bearing **USN:1CR14CS167** in partial fulfillment of the award of the degree of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belgaum, during the year 2017-18. It is certified that all corrections / suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

-------------------------	-------------------------	-------------------------
Signature of Guide	Signature of HoD	Signature of Principal
Mr. Shivaraj V B	**Dr. Jhansi Rani P**	**Dr. Sanjay Jain**
Assistant Professor	Professor & Head	Principal
Department of CSE	Department of CSE	CMRIT,
CMRIT, Bengaluru - 37	CMRIT, Bengaluru - 37	Bengaluru - 37

External Viva

	Name of the Examiners	Institution	Signature with Date
1.	-------------------------	----------------------------	----------------------------
2.	-------------------------	----------------------------	----------------------------

Acknowledgement

We take this opportunity to thank all of those who have generously helped us to give a proper shape to our work and complete our BE project successfully. A successful project is fruitful culmination efforts by many people, some directly involved and some others indirectly, by providing support and encouragement.

We would like to thank **Dr. SANJAY JAIN** , Principal , CMRIT , for providing excellent academic environment in the college.

We would like to express our gratitude towards **Dr. JHANSI RANI** , Professor & HOD , Dept of CSE , CMRIT , who provided guidance and gave valuable suggestions regarding the project.

We consider it a privilege and honour to express our sincere gratitude to our Internal Guide **Mr. Shivaraj V B** , Asst. Professor , Department of Computer Science & Engineering , CMRIT , for her valuable guidance throughout the tenure of this project work.

Santosha Bhat
Vageesh K C
Vishwaradhya
Yaswanth R K

Table of Contents

List of Figures

Abstract

In the traditional system there was a need to go on the voting booth and cast a vote. People from distinct places who did not have their voting cards cannot cast their votes. Also authentication of the user was not good and appropriate. There was a lot of paper work which was very time consuming. The results needed to be calculated manually which was very time consuming process. Therefore the proposed system is developed to remove the efforts needed in the traditional voting process.

The proposed system has an application developed on android phone via which the user can cast his vote from anywhere on the face of the globe. The user registers by giving his personal details which gets stored in the database at the server side. After the voting date is fixed the user gets notification on the android phone via GCM (Google Cloud Messaging). After that the user opens the application. The finger print authentication is done and then the system will allow user to get into home screen. Then user opens the voting form he casts his vote and then click on submits button and then logout. On the server side we can check the results. The GCM is which sends notification to user's android phone. The sqlite is the local database of the user's phone. If his internet connection is off then the notification and other details get stored on his local server. When he starts his internet connection then this message are retrieved from the GCM that is the local database of the android phone.

Chapter 1

PREAMBLE

1.1 Introduction

ANDROID BASED ONLINE VOTING SYSTEM is an online voting technique. In this system people who have citizenship of India and whose age is above 18 years of age and any sex can give his vote online without going to any physical polling station. There is a database which is maintained in which all the names of voters with complete information is stored.

In ANDROID BASED ONLINE VOTING SYSTEM a voter can use his voting right online without any difficulty. He has to be registered first for him to vote. Registration is mainly done by the system administrator for security reasons. The system Administrator registers the voters on a special site of the system visited by him only by simply filling a registration form to register voter. Citizens seeking registration are expected to contact the system administrator to submit their details. After the validity of them being citizens of India has been confirmed by the system administrator by comparing their details submitted with those in existing databases such as those as the Registrar of Persons, the citizen is then registered as a voter.

After registration, the voter is assigned a secret Voter ID with which he/she can use to log into the system and enjoy services provided by the system such as voting. If invalid/wrong details are submitted, then the citizen is not registered to vote.

1.2 Existing System

In existing system of voting, voter go on voting booth on the day of voting. There is no any centralized system, where we can cast our vote from remote location. There is no proper authentication present of an individual. There is manual work involved, so it may generate errors in the system. The Traditional approach is very time consuming and have got manual errors.

1.3 Problem Statement

- **Expensive and Time consuming:** The process of collecting data and entering this data into the database takes too much time and is expensive to conduct, for example, time and money is spent in printing data capture forms, in preparing registration stations together with human resources, and there after advertising the days set for registration process including sensitizing voters on the need for registration, as well as time spent on entering this data to the database.

- **Too much paper work:** The process involves too much paper work and paper storage which is difficult as papers become bulky with the population size.
- **Errors during data entry:** Errors are part of all human beings; it is very unlikely for humans to be 100 percent efficient in data entry.
- **Loss of registration forms:** Some times, registration forms get lost after being filled in with voters details, in most cases these are difficult to follow-up and therefore many remain unregistered even though they are voting age nationals and interested in exercising their right to vote.
- **Short time provided to view the voter register:** This is a very big problem since not all people have free time during the given short period of time to check and update the voter register.
- Above all, a number of voters end up being locked out from voting.

1.4 Objective of Project

The specific objectives of the project include:

- Reviewing the existing/current voting process or approach in India.
- Coming up with an automated voting system in India.
- Implementing a an automated/online voting system.
- Validating the system to ensure that only legible voters are allowed to vote.

1.5 Proposed System

This system was proposed to eliminate the trouble of people to go and vote at the voting booth. Whenever schedule date notification is get on user android device, user can cast their vote from anywhere and at anytime. For casting the vote particular user should be authorized so the proposed system will done authentication of voter. With the help of mobile finger print device.

This proposed system is based on fingerprint authorization.During registration the proposed system will get all user details.And while user is signing in to the application this will authorizes the user with the help of fingerprint device. In proposed system advancement in android device is very easy and secure voting. The proposed system provides the specification and requirements for E-Voting using an Android platform.

1.6 Scope of Study

It is focused on studying the existing system of voting in India and to make sure that the peoples vote is counts, for fairness in the elective positions. This is also will produce:

- Less effort and less labor intensive, as the primary cost and focus primary on creating, managing, and running a secure web voting portal.
- Increasing number of voters as individuals will find it easier and more convenient to vote, especially those abroad.

Chapter 2

LITERATURE SURVEY

2.1 Introduction

Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

2.2 Literature Survey

To make the voting process very easy and efficient wireless and web technologies are used. The online-voting system has the possibility of secure, easy and safe way to capture and count the votes in the election. The proposed system provides the specification and requirements for Online-Voting using an Android platform. The Online-voting means the voting process in election by using mobile phone. The android platform is used to develop an Online-voting application. Through a general diagram the introduction of the system is presented. The proposed Online-voting system will be presented with the obtained results.The proposed system also described how the android mobile phones are efficient. The android platform is used to develop a reliable and efficient application. Using the face-book APIs provided by the android SDK (software development kit) the login can be done very efficiently.

2.3 Review Concept

- **Efficient E-voting Android Based System:** The advancement in the mobile devices the wireless sensor network and web technologies that may given rise to the new applications which will make the voting process very easy and efficient.The research project provides the specifications and requirements for the E-Voting using an Android Platform. The E-Voting means the voting process in election by using electronic device. E-Voting is not more secure as possible than the paper-ballot system. Electronic failures might occur with such a system.

- **Web-Based Voting System Using Fingerprint:** The problem of voting system is still critical in terms of safety and the security. The design and development of the web based voting system using fingerprint verification in order to provide a high performance with high security to the voting system also the user can use the web technology to make the voting system more practical. The proposed architecture presents the module for election system for selecting the president in university. The Electronic voting System allows the voters to scan their fingerprint and then it will matched with an already saved image with in a database. Web based Voting System using Fingerprint Recognition. The online system has provided an efficient way to cast their votes, free of fraud, and make the system more trust, economic and fast. It may have used based on Minute based fingerprint identification and matching with high accuracy. More number of peoples were participate in the election to vote their candidate for choosing the majority votes. When limited number of people were register in the system means that particular candidate only allowed for voting and the remaining peoples may not allowed for the voting .

2.4 Survey Papers

- **Title:** Election voting system using mobile (m-voting)-(2013)
  Context:
  Biometric identification of voter by using iris pattern scanning and otp of each individual voter for validation and authorization purpose.

- **Title:** Transforming voting paradigm the shift from inline through online to mobile voting -(2014)
  Context:
  A model for m-voting that utilizes biometric (voice recognition), nfc (used to store secret question and answer), mobile phone and location based technology.bring your own devices concept introduced.

- **Title:** Event based application of voting system for mobile devices-(2015)
  Context:
  An objective was to focus on implementing mobile voting application for measuring the cognitive competencies of social group, particularly in the field of education.

- **Title:** Mobile based facial recognition using otp verification for voting system -(2015)
  Context:
  Biometric identification of voter by using face recognition and otp of each individual voter for validation and authorization purpose.

Chapter 3

THEORETICAL BACKGROUND

3.1 Android Operating System

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The operating system has since gone through multiple major releases, with the current version being 8.1 ”Oreo”, released in December 2017. The core Android source code is known as Android Open Source Project (AOSP), and is primarily licensed under the Apache License.

Android is also associated with a suite of proprietary software developed by Google, including core apps for services such as Gmail and Google Search, as well as the application store and digital distribution platform Google Play, and associated development platform. These apps are licensed by manufacturers of Android devices certified under standards imposed by Google, but AOSP has been used as the basis of competing Android ecosystems, such as Amazon.com’s Fire OS, which utilize its own equivalents to these Google Mobile Services.

Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2017, it has over two billion monthly active users, the largest installed base of any operating system, and as of 2017, the Google Play store features over 3.5 million apps.

- **Interface** Android’s default user interface is mainly based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a virtual keyboard.Game controllers and full-size physical keyboards are supported via Bluetooth or USB.The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user.

- **Applications** Applications (”apps”), which extend the functionality of devices, are written using the Android software development kit (SDK)and, often, the Java programming language.Java may be combined with C/C++, together with a choice of non-default runtimes that allow better C++ support. The Go programming language is also supported, although with a limited set of application programming interfaces (API). In May 2017, Google announced support for Android app development in the Kotlin programming language.

- **Memory Management** Since Android devices are usually battery-powered, Android is designed to manage processes to keep power consumption at a minimum. When an application is not in use the system suspends its operation so that, while available for immediate use rather than closed, it does not use battery power or CPU resources.Android manages the applications stored in memory automatically: when memory is low, the system will begin invisibly and automatically closing inactive processes, starting with those that have been inactive for the longest amount of time. Lifehacker reported in 2011 that third-party task killer applications were doing more harm than good.

- **Hardware** The main hardware platform for Android is ARM (the ARMv7 and ARMv8-A architectures), with x86, MIPS and MIPS64, and x86-64 architectures also officially supported in later versions of Android. The unofficial Android-x86 project provided support for x86 architectures ahead of the official support.The MIPS architecture was also supported ahead of Google's support. Since 2012, Android devices with Intel processors began to appear, including phones and tablets. While gaining support for 64-bit platforms, Android was first made to run on 64-bit x86 and then on ARM64. Since Android 5.0 "Lollipop", 64-bit variants of all platforms are supported in addition to the 32-bit variants.
  Requirements for the minimum amount of RAM for devices running Android 7.1 range from in practice 2 GB for best hardware, down to 1 GB for the most common screen, to absolute minimum 512 MB for the lowest spec 32-bit smartphone. The recommendation for Android 4.4 is to have at least 512 MB of RA while for "low RAM" devices 340 MB is the required minimum amount that does not include memory dedicated to various hardware components such as the baseband processor. Android 4.4 requires a 32-bit ARMv7, MIPS or x86 architecture processor (latter two through unofficial ports),together with an OpenGL ES 2.0 compatible graphics processing unit (GPU).Android supports OpenGL ES 1.1, 2.0, 3.0, 3.1 and as of latest major version, 3.2 and since Android 7.0 Vulkan (and version 1.1 available for some devices). Some applications may explicitly require a certain version of the OpenGL ES, and suitable GPU hardware is required to run such applications.

- **Linux Kernel** Android's variant of the Linux kernel has further architectural changes that are implemented by Google outside the typical Linux kernel development cycle, such as the inclusion of components like device trees, ashmem, ION, and different out of memory (OOM) handling. Certain features that Google contributed back to the Linux kernel, notably a power management feature called ”wakelocks”, were initially rejected by mainline kernel developers partly because they felt that Google did not show any intent to maintain its own code. Google announced in April 2010 that they would hire two employees to work with the Linux kernel community, but Greg Kroah-Hartman, the current Linux kernel maintainer for the stable branch, said in December 2010 that he was concerned that Google was no longer trying to get their code changes included in mainstream Linux. Google engineer Patrick Brady once stated in the company’s developer conference that ”Android is not Linux”, with Computerworld adding that ”Let me make it simple for you, without Linux, there is no Android”. Ars Technica wrote that ”Although Android is built on top of the Linux kernel, the platform has very little in common with the conventional desktop Linux stack”.

- **Technical Security Feuture** Android applications run in a sandbox, an isolated area of the system that does not have access to the rest of the system’s resources, unless access permissions are explicitly granted by the user when the application is installed, however this may not be possible for pre-installed apps. It is not possible, for example, to turn off the microphone access of the pre-installed camera app without disabling the camera completely. This is valid also in Android versions 7 and 8.

  Since February 2012, Google has used its Google Bouncer malware scanner to watch over and scan apps available in the Google Play store. A ”Verify Apps” feature was introduced in November 2012, as part of the Android 4.2 ”Jelly Bean” operating system version, to scan all apps, both from Google Play and from third-party sources, for malicious behavior. Originally only doing so during installation, Verify Apps received an update in 2014 to ”constantly” scan apps, and in 2017 the feature was made visible to users through a menu in Settings.

3.2 SDK and API for Fingerprint Integration

Top technology corporations have been focusing on building ecosystem of integrated services, which can communicate with each other and data is seamlessly shared among them. It requires engagement of a large community of developers to develop apps and service and their integration with other services. Tech firms try their best to persuade developers to build for their ecosystems, so that the more and more services could be integrated to make the ecosystems grow. They provide training material, videos, SDKs, APIs, tools and help for developers. The same applies on biometric software solution firms. Biometric software firms want developers to use their SDK and APIs to grow their market presence and revenue.

To integrate fingerprint hardware, developers would require fingerprint SDK that will allow them to access fingerprint hardware features and API, which can communicate with other software or services. For example, to integrate fingerprint hardware with an Android app, a developer needs to get the Android SDK and target API level. API level is determined by the Android versions the app wants to have compatibility with. SDKs and APIs are often available on software/service providers website and can be downloaded by anyone. In cloud biometrics, applications integrated with fingerprint sensors need to communicate with remote servers on each request. This communication is made possible by APIs. Biometrics as a Service or cloud biometrics service providers also provide APIs that can communicate with external services and software.

Some device hardware manufacturers like Samsung provide SDK and API for their devices to securely integrate apps with fingerprint hardware. User can use Androids generic API or devices manufacturers APK as required. Apple also provides iOS SDK and Touch ID API for fingerprint authentication.

3.3 Rules in Voting

- **Majority-Rule** The most common way of tallying all the votes is the majority rule, which is take the alternative that is preferred by a majority of the voters rank it first, placing the other second. With only two alternatives, this works perfectly. However, when there are more than two options in the vote, a very famous Condorcet Paradox can potentially occur and cause incoherence.

- **Positional Voting** Positional voting is another common voting system. Being different from building up ranked list with pairwise comparison and majority rule votes aggregation, it produces a group ranking directly from the individual ranking. In this type of system, each alternative receives a weight based on its position in the preference list. For example, in a voters ranking with k alternatives, the first-ranked alternative receives a weight of k-1, the second-ranked alternative receives a weight of k-2, and etc. Then, the last ranked alternative will receive a weight of zero. Aggregating all the weights assigned to each alternative and ordering the alternatives by the aggregated weights produce the group-ranking list.

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

4.1 Introduction

This chapter describes about the requirements. It specifies the hardware and software requirements that are required in order to run the application properly. The Software Requirement Specification (SRS) is explained in detail, which includes overview of dissertation as well as the functional and non-functional requirement of this dissertation.

A SRS document describes all data, functional and behavioral requirements of the software under production or development. SRS is a fundamental document, which forms the foundation of the software development process. Its the complete description of the behavior of a system to be developed. It not only lists the requirements of a system but also has a description of its major feature. Requirement Analysis in system engineering and software engineering encompasses those tasks that go into determining the need or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirement Analysis is critical to the success to a development project. Requirement must be documented, measurable, testable, related to in identified business needs or opportunities, and defined to a level of detail sufficient for system design.

The SRS functions as a blueprint for completing a project. The SRS is often referred to as the ”parent” document because all subsequent project management documents, such as design specifications, statements of work, software architecture specification, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and non-functional requirements only.

Thus the goal of preparing the SRS document is to

- To facilitate communication between the customer, analyst, system developers, maintainers.
- To serve as a contrast between purchaser and supplier.
- To firm foundation for the design phase.
- Support system testing facilities.
- Support project management and control.
- Controlling the evolution of the system.

4.2 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

- Input test case must not have compilation and runtime errors.
- The application must not stop working when kept running for even a long time.
- The application must function as expected for every set of test cases provided.
- The application should generate the output for given input test case and input parameters.
- The application should generate on-demand services.

4.3 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

- Product Requirements
- Organizational Requirements
- User Requirements
- Basic Operational Requirements

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. This should be contrasted with functional requirements that define specific behaviour or functions. The plan for implementing non-functional requirements is detailed in the system architecture. Broadly, functional requirements define what a system is supposed to do and non- functional requirements define how a system is supposed to be. Functional requirements are usually in the form of system shall do ¡requirement¿, an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of system shall be ¡requirement¿, an overall property of the system as a whole or of a particular aspect and not a specific function. The systems' overall properties commonly mark the difference between whether the development project has succeeded or failed.

4.3.1 Product Requirements

- Portability: Since the system is designed to run using Android, the system is portable.
- Correctness: It follows a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.
- Ease of Use: The front end is designed in such a way that it provides an interface which allows the user to interact in an easy manner.
- Modularity: The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product.
- Robustness: This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness.

whereas evolution quality involves testability, maintainability, extensibility or scalability.

4.3.1.1 Organizational Requirements

Process Standards: IEEE standards are used to develop the application which is the standard used by the most of the standard software developers all over the world. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

4.3.1.2 User Requirements

The user requirements document (URD) or user requirements specification is a document usually used to software engineering that specifies the requirements user expects from software to be constructed in software project. Once the required information is completely gathered it is documented in a URD, which is meant to spell out exactly what the software must do and becomes part of the contractual agreement. A customer cannot demand feature not in the URD, whilst the developer cannot claim the product is ready if it does not meet an item of the URD. The URD can be used as a guide to planning cost, timetables, milestones, testing etc. The explicit nature of the URD allows customers to show it to various stakeholders to make sure all necessary features are described. Formulating a URD requires negotiation to determine what is technically and economically feasible. Preparing a URD is one of those skills that lies between a science and economically feasible. Preparing a URD is one of those skills that lies between a science and an art, requiring both software technical skills and interpersonal skills.

4.3.1.3 Basic Operational Requirements

Operational requirement is the process of linking strategic goals and objectives to tactic goals and objectives. It describes milestones, conditions for success and explains how, or what portion of, a strategic plan will be put into operation during a given operational period, in the case of, a strategic plan will be put into operation during a given operational period, in the case of commercial application, a fiscal year or another given budgetary term. An operational plan is the basis for, and justification of an annual operating budget request. Therefore, a five-year strategic plan would typically require five operational plans funded by five operating budgets. Operational plans should establish the activities and budgets for each part of the organization for the next 1-3 years. They link the strategic plan with the activities the organization will deliver and the resources required to deliver them. An operational plan draws directly from agency and program strategic plans to describe agency and program missions and goals, program objectives, and program activities. Like a strategic plan, an operational plan addresses four questions:

- Where are we now?
- Where do we want to be?
- How do we get there?

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points:

- Mission profile or scenario: It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.

- Performance and related parameters: It points out the critical system parameters to accomplish the mission
- Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.
- Operational life cycle: It defines the system lifetime

4.4 Hardware Requirements

- Processor: 800MHz Intel Pentium III or equivalent
- Memory: 512 MB
- Disk space: 750 MB of free disk space

  For Mobile:
- Running Device : Android.
- RAM : 512 MB minimum.
- Internal storage :1 GB.
- Internet access : Yes.
- Sensors : fingerprint device.

4.5 Software Requirements

- Operating System : Windows 10 and Ubuntu.
- Coding Language : Java.
- Tools : Eclipse and Netbeans IDE.
- Library support : SDK,JDK,NDK.

Chapter 5

SYSTEM ANALYSIS

5.1 Introduction

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

5.2 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- Operational Feasibility
- Economical Feasibility
- Technical Feasibility
- Social Feasibility

5.3 Operational Feasibility

Some national Governments in the world are looking for the change in the current E-voting system.This needs to be developed in suc a way that the government can abble to fund the all amount for developing, designing and maintaining the E-voting system.And they should make sure that they can handle all interuption,faults in the proposed voting system.

5.4 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

5.5 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.6 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Chapter 6

SYSTEM DESIGN

6.1 Introduction

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

6.2 System Development Methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

6.2.1 Model Phases

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

- **Requirement Analysis:** This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.
- **System Design:** Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:-algorithm, data structure, software architecture etc.
- **Coding:** In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code.

- **Implementation:** The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation
- **Testing:** In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.
- **Maintenance:** The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

6.2.2 Reason For Choosing Waterfall Model As Development Method

- Clear project objectives.
- Stable project requirements.
- Progress of system is measurable.
- Strict sign-off requirements.
- Helps you to be perfect.
- Logic of software development is clearly understood.
- Production of a formal specification
- Better resource allocation.
- Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.
- Less human resources required as once one phase is finished those people can start working on to the next phase.

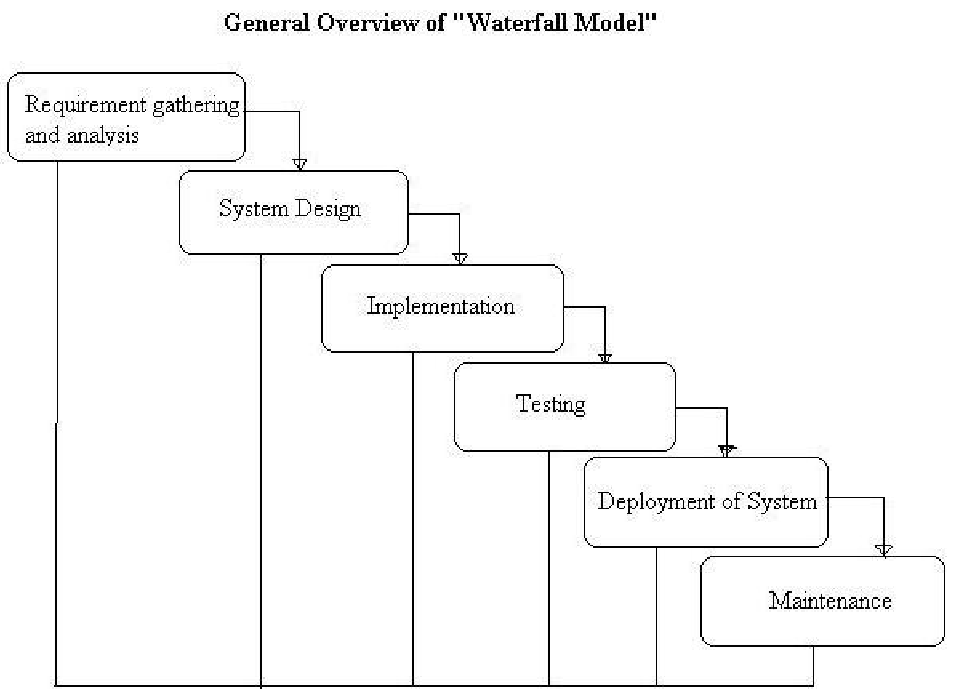

General Overview of "Waterfall Model"
Requirement gathering and analysis
System Design
Implementation
Testing
Deployment of System
Maintenance

Figure 6.1: Waterfall Model

6.3 Design Using UML

Designing UML diagram specifies, how the process within the system communicates along with how the objects with in the process collaborate using both static as well as dynamic UML diagrams since in this ever-changing world of Object Oriented application development, it has been getting harder and harder to develop and manage high quality applications in reasonable amount of time. As a result of this challenge and the need for a universal object modeling language every one could use, the Unified Modeling Language (UML) is the Information industries version of blue print. It is a method for describing the systems architecture in detail. Easier to build or maintains system, and to ensure that the system will hold up to the requirement changes.

6.4 Data Flow Diagram

A data flow diagram (DFD) is graphic representation of the ”flow” of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. DFDs show the flow of data from external entities into the system, how the data moves from one process to another, as well as its logical storage. There are only four symbols: 1. Squares representing external entities, which are sources and destinations of information entering and leaving the system. 2. Rounded rectangles representing processes, in other methodologies, may be called 'Activities', 'Actions', 'Procedures', 'Subsystems' etc. which take data as input, do processing to it, and output it. 3. Arrows representing the data flows, which can either, be electronic data or physical items. It is impossible for data to flow from data store to data store except via a process, and external entities are not allowed to access data stores directly. 4. The flat three-sided rectangle is representing data stores should both receive information for storing and provide it for further processing.

6.5 Class Diagram

UML class diagram shows the static structure of the model. The class diagram is a collection of static modeling elements, such as classes and their relationships, connected as a graph to each other and to their contents. The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed.

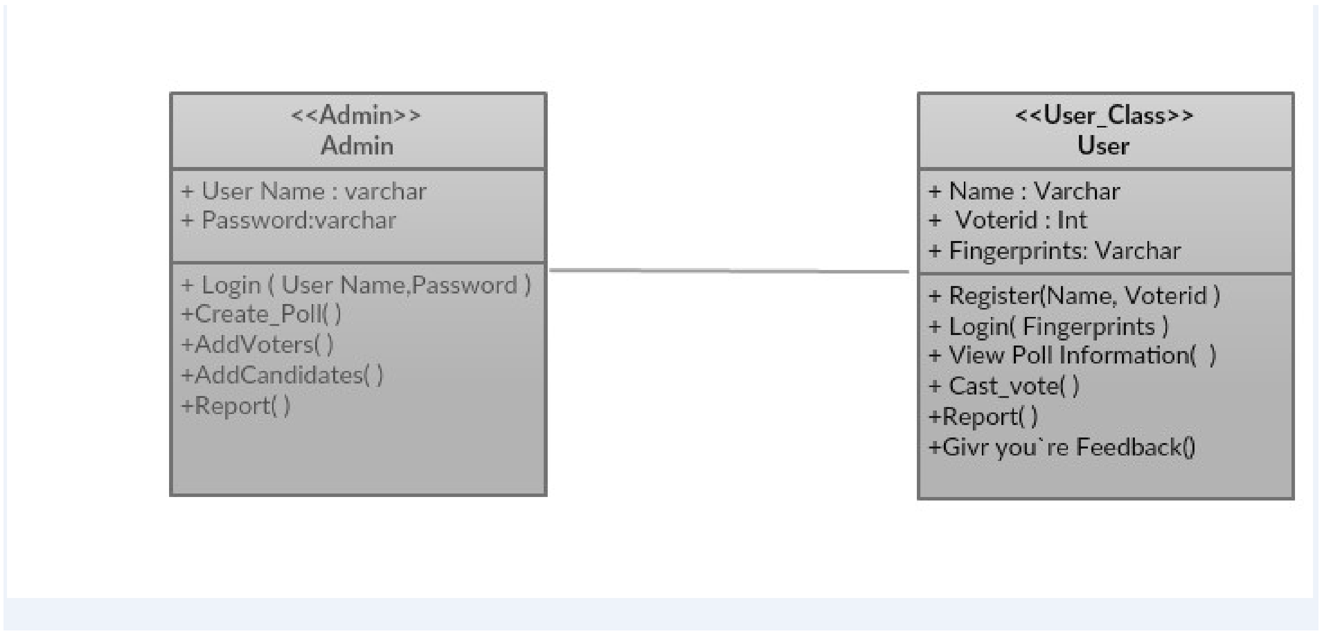


Figure 6.2: Class Diagram Of System

6.6 Use Case Diagram

A use case defines a goal-oriented set of interactions between external entities and the system under consideration. The external entities which interact with the system are its actors. A set of use cases describe the complete functionality of the system at a particular level of detail and it can be graphically denoted by the use case diagram.

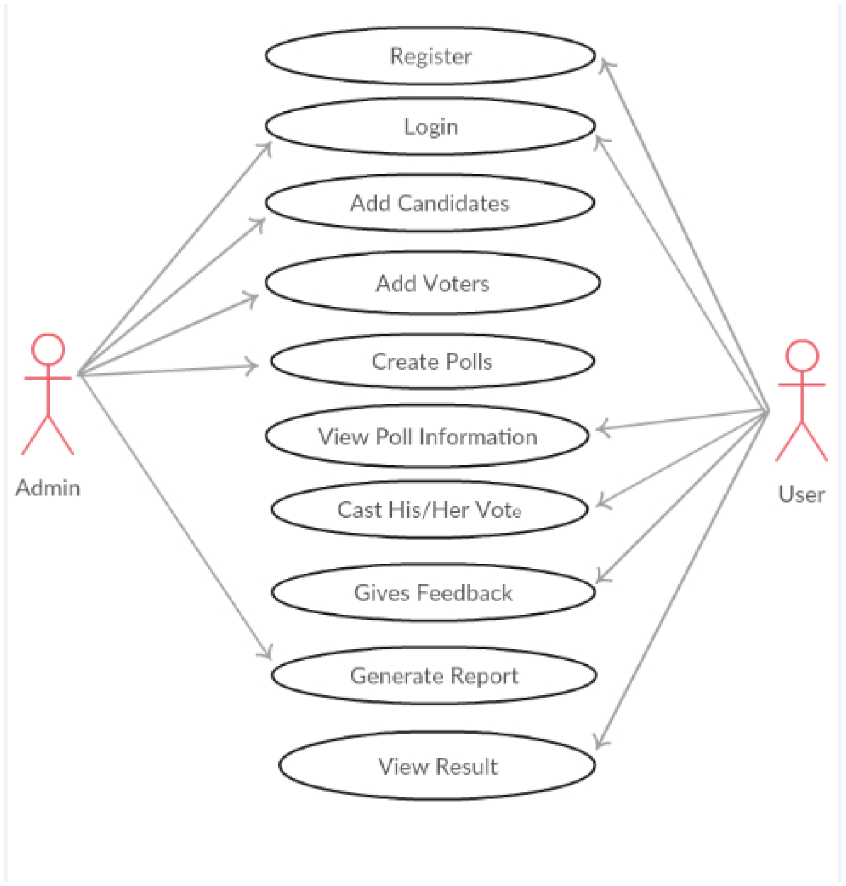


Figure 6.3: Use Case Diagram

6.7 Activity Diagram

An activity diagram shows the sequence of steps that make up a complex process. An activity is shown as a round box containing the name of the operation. An outgoing solid arrow attached to the end of the activity symbol indicates a transition triggered by the completion.
Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.
Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

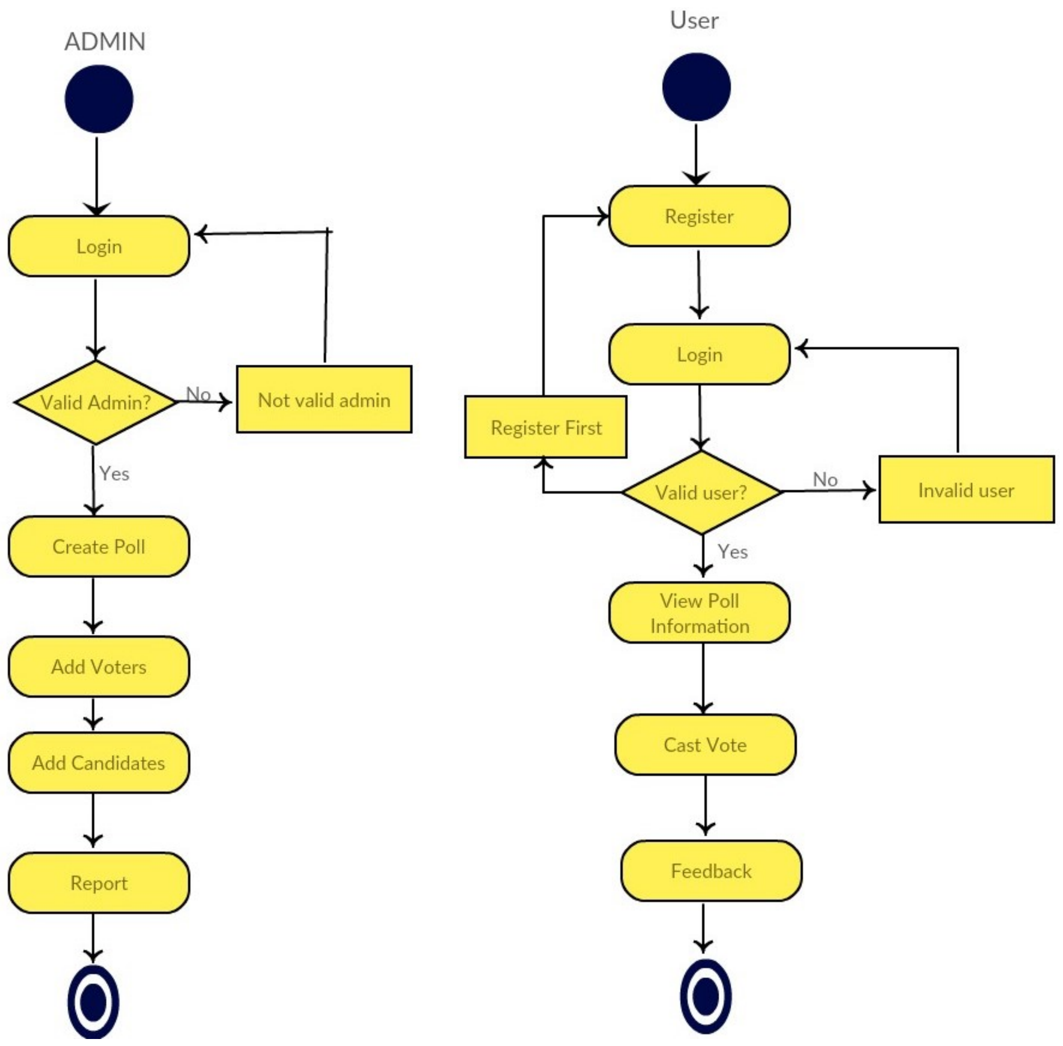


Figure 6.4: Activity Diagram

6.8 Sequence Diagram

Sequence diagram are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and the environment. A sequence diagram shows an interaction arranged in a time sequence. A sequence diagram has two dimensions: vertical dimension represents time, the horizontal dimension represents the objects existence during the interaction. **Basic elements:**

- **Vertical rectangle:** Represent the object is active (method is being performed).
- **Vertical dashed line:** Represent the life of the object.
- **X:** represent the life end of an object. (Being destroyed from memory)
- **Horizontal line with arrows:** Messages from one object to another.

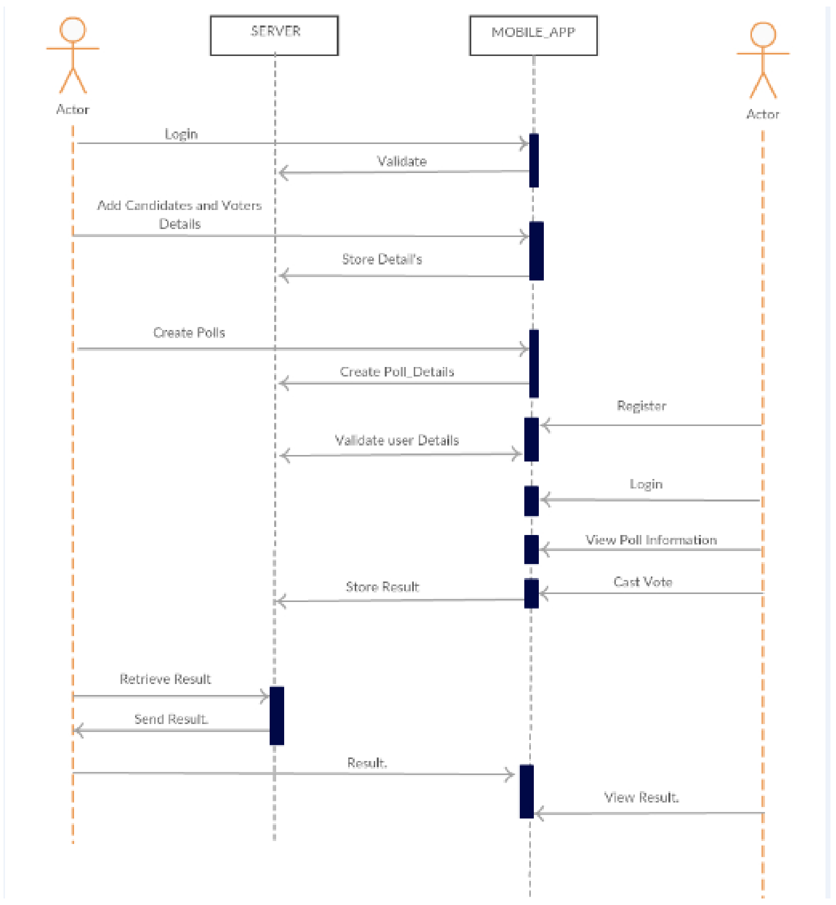


Figure 6.5: Sequence Diagram

Chapter 7

IMPLEMENTATION

7.1 Introduction

The implementation phase of the project is where the detailed design is actually transformed into working code. Aim of the phase is to translate the design into a best possible solution in a suitable programming language. This chapter covers the implementation aspects of the project, giving details of the programming language and development environment used. It also gives an overview of the core modules of the project with their step by step flow. The implementation stage requires the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Evaluation of the changeover method.
- Correct decisions regarding selection of the platform.
- Appropriate selection of the language for application development.

7.2 Fingerprint Authentication

This code shows procedure for fingerprint authentication.

```
package com.sample;

import android.support.v7.app.ActionBarActivity;
import android.speech.tts.TextToSpeech;
import android.support.v7.app.ActionBarActivity;
import android.annotation.TargetApi;
import android.app.Activity;
import android.app.KeyguardManager;
import android.content.pm.PackageManager;
import android.hardware.fingerprint.FingerprintManager;
import android.os.Build;
import android.security.keystore.KeyGenParameterSpec;
import android.security.keystore.KeyPermanentlyInvalidatedException;
import android.security.keystore.KeyProperties;
import android.os.Bundle;
import android.widget.TextView;
import java.io.IOException;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.KeyStore;
import java.security.KeyStoreException;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.UnrecoverableKeyException;
import java.security.cert.CertificateException;
import java.util.Locale;

import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;

import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;


public class FingerprintActivity extends Activity {
private KeyStore keyStore;
Variable used for storing the key in the Android Keystore container
private static final String KEY_NAME = "androidHive";
private Cipher cipher;
private TextView textView;
@TargetApi(23)
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_fingerprint);

 Initializing both Android Keyguard Manager and Fingerprint Manager
KeyguardManager keyguardManager = (KeyguardManager)
getSystemService(KEYGUARD_SERVICE);
FingerprintManager fingerprintManager = (FingerprintManager)
getSystemService(FINGERPRINT_SERVICE);
```

```
textView = (TextView) findViewById(R.id.errorText);
if(!fingerprintManager.isHardwareDetected()){
textView.setText("Your␣Device␣does␣not␣have␣a␣Fingerprint␣Sensor");
}else {
if (!fingerprintManager.hasEnrolledFingerprints()) {
textView.setText("Register␣at␣least␣one␣fingerprint␣in␣Settings");
}else{
// Checks whether lock screen security is enabled or not
if (!keyguardManager.isKeyguardSecure())
textView.setText("Lock␣screen␣security␣not␣enabled␣in␣Settings");
}else{
generateKey();
if (cipherInit()) {
FingerprintManager.CryptoObject cryptoObject = new
FingerprintManager.CryptoObject(cipher);
FingerprintHandler helper = new FingerprintHandler(this);
helper.startAuth(fingerprintManager, cryptoObject);
                        }
                    }
                }
            }
        }

    @TargetApi(Build.VERSION_CODES.M)
    protected void generateKey() {
        try {
            keyStore = KeyStore.getInstance
            ("AndroidKeyStore");
        } catch (Exception e) {
            e.printStackTrace();
        }

        KeyGenerator keyGenerator;
        try {
 keyGenerator = KeyGenerator.getInstance(
 KeyProperties.KEY_ALGORITHM_AES, "AndroidKeyStore");
        } catch (Exception e) {
 throw new RuntimeException
 ("Failed␣to␣get␣KeyGenerator␣instance", e);
        }
        try {
            keyStore.load(null);
            keyGenerator.init(new
            KeyGenParameterSpec.Builder(KEY_NAME,
            KeyProperties.PURPOSE_ENCRYPT |
            KeyProperties.PURPOSE_DECRYPT)
            .setBloc
            kModes(KeyProperties.BLOCK_MODE_CBC)
            .setUserAuthenticationRequired(true)
            .setEncryptionPaddings(
            keyProperties.ENCRYPTION_PADDING_PKCS7)
            .build());
            keyGenerator.generateKey();
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    @TargetApi(Build.VERSION_CODES.M)
    public boolean cipherInit() {
    try {
    cipher = Cipher.getInstance
    (KeyProperties.KEY_ALGORITHM_AES + "/" +
KeyProperties.BLOCK_MODE_CBC + "/" +
KeyProperties.ENCRYPTION_PADDING_PKCS7);
        } catch (Exception e) {
            throw new RuntimeException
            ("Failed␣to␣get␣Cipher", e);
        }

        try {
 keyStore.load(null);
 SecretKey key = (SecretKey) keyStore.getKey(KEY_NAME,
                    null);
            cipher.init(Cipher.ENCRYPT_MODE, key);
            return true;
        } catch (KeyPermanentlyInvalidatedException e) {
            return false;
        } catch (Exception e) {
 throw new RuntimeException("Failed to init Cipher", e);
        }
    }
}
```

7.3 Caste Voting

This code shows procedure for Casting vote.

```
package com.sample;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.List;
import java.util.StringTokenizer;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;

import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;


import android.support.v7.app.ActionBarActivity;
import android.R.integer;
import android.app.Activity;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class User_Vote extends Activity {
        TextView t1;
        EditText e1;
        LoginDataBaseAdapter lbd;
        ArrayList<String> al=new ArrayList<String>();
        ListView listView ;
        String stre[]=new String[100];
        public ArrayList<String> list=new ArrayList<String>();
        String vname,vid;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_user__vote);
                lbd=new LoginDataBaseAdapter(this);
                lbd.open();
                t1=(TextView)findViewById(R.id.textView1);
//              e1=(EditText)findViewById(R.id.editText1);
                listView=(ListView)findViewById(R.id.listView1);
                Bundle extras=getIntent().getExtras();
        if(extras!=null)
        {
                String result=extras.getString("result");
//              t1.setText(result);
//              e1.setText(result);
                int size=5;
                int start=0;
                int end=size;
                String str3=result.replaceAll("\\[", "");
                String str34=str3.replaceAll("\\]", "");
```

```
                    String str23[]=new String[100];
                    String str[]=str34.split(",");
                    al=lbd.getUser();
                    vname=al.get(0);
                    vid=al.get(1);
                    sObje s[]= list.toArray();
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, android.R.id.text1, str);

listView.setAdapter(adapter);
listView.setOnItemClickListener(new  OnItemClickListener() {

@Override
public void onItemClick(AdapterView<?> parent,
View view,int position, long id) {
int itemPosition     = position;
String  itemValue    = (String) listView.
getItemAtPosition(position);
Toast.makeText(getApplicationContext(),
"Position␣:"+itemPosition+"␣␣ListItem␣:␣" +itemValue ,
 Toast.LENGTH_SHORT).show();
new UserVote().execute(vname,vid,itemValue);

                    }
             });
          }
          }

          public class UserVote extends AsyncTask<String,
          integer, String>
          {
@Override
protected String doInBackground(String... params) {

                              String s=postData(params);

               return s;
                    }
```

```
private String postData(String[] params) {
        // TODO Auto-generated method stub
//Toast.makeText(AddCandidate.this,
 "user␣date"+params, Toast.LENGTH_LONG).show();
HttpClient httpClient=new DefaultHttpClient();
 HttpPost httppost = new HttpPost
 ("http://192.168.100.19:8084//VoteSample/UserVote");
 String origresponseText="";
  try{
  List<NameValuePair> nameValuePairs =
  new ArrayList<NameValuePair>();
  nameValuePairs.add(new BasicNameValuePair
  ("vname",params[0]));
  nameValuePairs.add(new BasicNameValuePair
  ("vid", params[1]));
  nameValuePairs.add(new BasicNameValuePair
  ("vote", params[2]));

  httppost.setEntity(new
  UrlEncodedFormEntity(nameValuePairs));
  HttpResponse response=httpClient.execute(httppost);
  HttpEntity rp = response.getEntity();
  String t=rp.toString();
origresponseText=readContent(response);
             }
             catch(Exception e)
             {
                 e.printStackTrace();
             }String responseText = origresponseText.
             substring(7, origresponseText.length());

                   return responseText;
                 }
        String readContent(HttpResponse response)
        {
         String text = "";
     InputStream in =null;

  try{
```

```
  in = response.getEntity().getContent();
   BufferedReader reader = new
   BufferedReader(new InputStreamReader(in));
   StringBuilder sb = new StringBuilder();
   String line = null;
   while ((line = reader.readLine()) != null) {
   sb.append(line + "\n");
                        }
                      text = sb.toString();

                      }
 catch (IllegalStateException e) {
 e.printStackTrace();

} catch (IOException e) {
      e.printStackTrace();
                      }
                      finally {
                          try {

                     in.close();
                } catch (Exception ex) {
                        }
                  }

              return text;
                @Override
protected  void onPostExecute(String result) {
 Toast.makeText(User_Vote.this, result,
 Toast.LENGTH_LONG).show();
                      if(result!=null)
                      {
                              String msg=result;
                              try {
Toast.makeText(UserHomeActivity.this, "jsonObject"+
jsonObject,      Toast.LENGTH_SHORT).show();
 startActivity(welcome);

} catch (Exception e) {
```

```
// TODO Auto-generated catch block
        e.printStackTrace();
   }
   }
                              else
                              {
 Toast.makeText(User_Vote.this,
 "present no polls are available",
  Toast.LENGTH_SHORT).show();
                              }
                        }
                  }
}
```

7.4 Interfacing Android Application with Server

As we have seen,the project must consists of two ends.One is Android application which provides interfaces for user/voter, and other is sever side which is used to store all related information of our system.First we need to create admin activity for the system,where admin can login,add candidate,add voters,createpoll.And as well,we need to store the information in server,that server is created in NETBEANS IDE.The server will be having database management system,and all the implementaion fo storing the data are need to be done.AS well it will provide retrieving the existing data for client.The client is nothing but application user interface.Once after this main work is done,we will be able to implement application for user.In which the user can register himself and login to the application after the fingerprint authentication.And the voter will caste his vote.And result calculation implement will be done.

Chapter 8

TESTING AND RESULTS

8.1 Introduction

Testing is an important phase in the development life cycle of the product this was the phase where the error remaining from all the phases was detected. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software. Once the implementation is done, a test plan should be developed and run on a given set of test data. Each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is suppose to do. Testing is the final verification and validation activity within the organization itself. In the testing stage following goals are tried to achieve:-

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as the solution to the original problem.
- To provide operational reliability of the system.

During testing the major activities are concentrated on the examination and modification of the source code. The test cases executed for this project are listed below. Description of the test case, steps to be followed; expected result, status and screenshots are explained with each of the test cases.

8.2 Testing Methedologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important types of testing are:

8.2.1 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level. Using white box testing we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.

- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to assure their validity.

8.2.2 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot see into it. The test provides inputs and responds to outputs without considering how the software works. It uncovers a different class of errors in the following categories:

- Incorrect or missing function.
- Interface errors.
- Performance errors.
- Initialization and termination errors.
- Errors in objects.

Advantages:

- The test is unbiased as the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.
- Test cases can be designed as soon as the specifications are complete.

8.2.3 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. **Test strategy and approach**
Field testing will be performed manually and functional tests will be written in detail.
Test objectives:

- All Components must work properly.
- Proper coordinates should be sent by the Android app to the Arduino
- The entry screen, messages and responses must not be delayed in the Android app.

8.3 Test Case 1

Function: void adminlogin()
Purpose: This function is used for admin to sign in.
Preconditions: the internet should be connected first.
Inputs: login Credentials
Expected Outputs: Should Open The admin home page.
Postconditions: Admin Can do Various Functions.

8.4 Test Case 2

Function: void createpoll()
Purpose: This function is used to create polls by adding candidatess and voters.
Preconditions: Should be connected to Internet and Admin need to be loggedin.
Inputs: All required information of voter and candidate will be given.
Expected Outputs: A successful message on adding candidates and voter.
Postconditions: Voters are able to register themself .

8.5 Test Case 3

Function: void voterreg()
Purpose: This function is used for voter to register themself.
Preconditions: The internet connection should be there.
Inputs: Voter need to give his own credential.
Expected Outputs: He will be registered for election .
Postconditions: Voter can login and do various functions .

8.6 Test Case 4

Function: void voterlogin()
Purpose: This function is used voter to signin to the app.
Preconditions: Internet connection shhould be there and he must have registered.
Inputs: He need to give username and password and get fingerprint authentication.
Expected Outputs: He will get into The home screen.
Postconditions: He will able to see poll information and can vote.

8.7 Test Case 5

Function: void castevote()

Purpose: Used to caste his vote.

Preconditions: The voter should have loggedin to the app.

Inputs: The inputs are taken when he caste his vote.

Expected Outputs: The vote is casted successfully.

Postconditions: The system will generate final result.

8.8 Images Of The Android APP

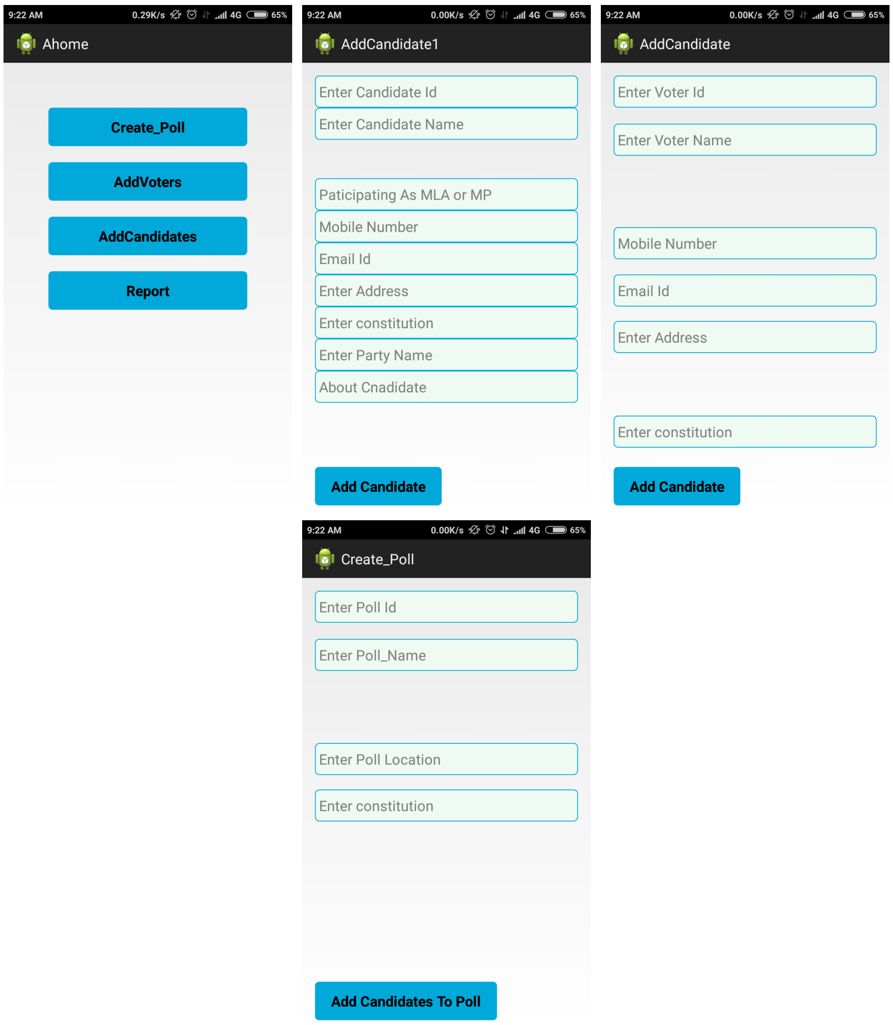
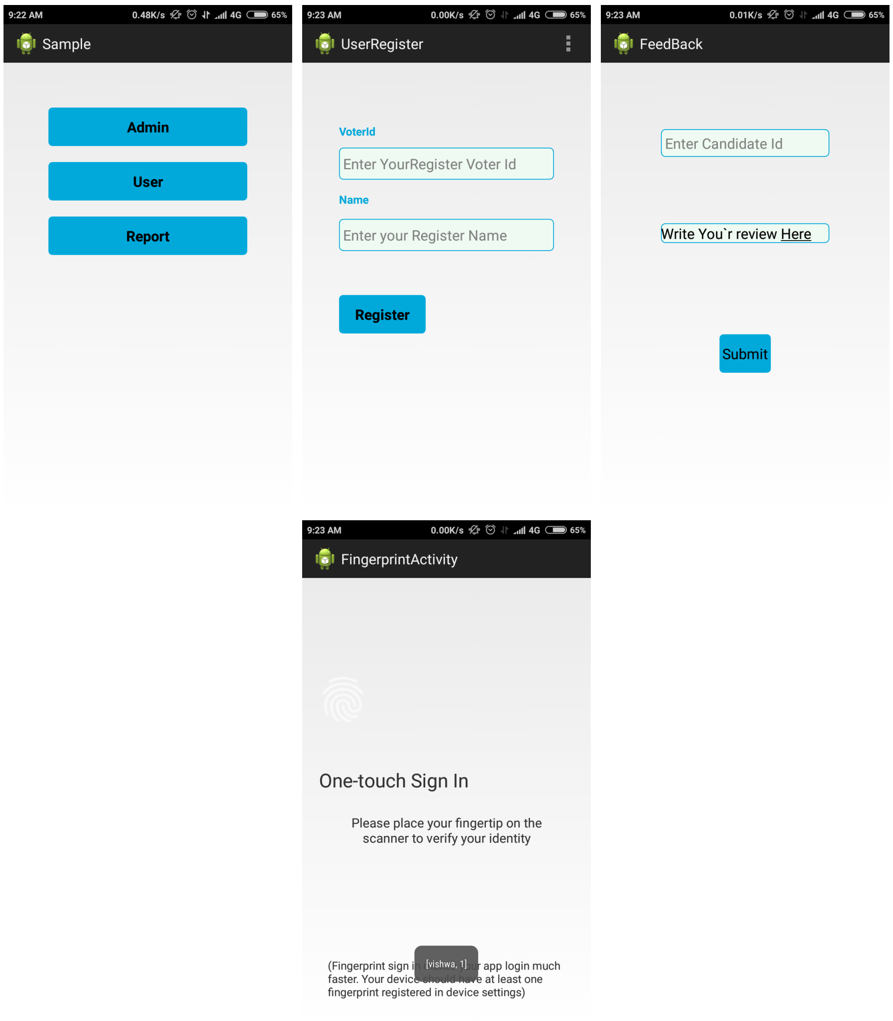


Figure 8.1: App Activities-1(Admin Activity)



Please place your fingertip on the scanner to verify your identity

(Fingerprint sign i r app login much faster. Your device at least one fingerprint registered in device settings)

Figure 8.2: App Activities-2(User Activity)

Chapter 9

CONCLUSION & FUTURE SCOPE

9.1 Conclusion

This Online Voting system will manage the Voters information by which voter can login and use his voting rights. The system will incorporate all features of Voting system. It provides the tools for maintaining voters vote to every party and it count total no. of votes of every party. There is a DATABASE which is maintained by the ELECTION COMMISION OF INDIA in which all the names of voter with complete information is stored.In this user who is above 18 years register his/her information on the database and when he/she want to vote he/she has to login by his id and password and can vote to any party only single time. Voting detail store in database and the result is displayed by calculation. By online voting system percentage of voting is increases. It decreases the cost and time of voting process. It is very easy to use and It is vary less time consuming. It is very easy to debug.

9.2 Future Scope

This project was the first implementation of a system of this nature. We identify that the work done both in terms of analysing and implementing the system is by no means complete. In this section we list the things that were either left open by this project or were opened by the analysis performed and the lessons learned during our interaction with the subject.

9.3 Build a Complete System

The complete system can be established with a proper significance of finger print hardware device.This will bring good security for the system.Since scurity issues are the most important part of any system,We are goin with fingerprint authentication.In addition to this system we can implement same with face authentication algorithms.But it will be difficult to make proper face matching stratgies.

9.4 Research Issues

Analyse and implement different path tracking algorithms to test their effectiveness under different situations.Determine what is the correct defination of pathway quality and how to measure it.Analyse how this system can update the path to be traversed on the go.

9.5 Implementation Issues

We can allow system to use seperate fingerprint hardware.Implememtation procedure involves proper fingeprint API and hardware support.We need to have good internet connection.SDK version to be taken so that it should support fingerprint device.

References

[1] L. Foresti, C. S. Regazzoni, and R. Visvanathan, Scanning the issue/technologySpecial issue on video communications, processing and understanding for third generation surveillance systems, Proc. IEEE, vol. 89, no. 10, pp. 13551367, Oct. 2016.

[2] S.Misra,M. Reisslein, and G. Xue, A survey ofmultimedia streaming in wireless sensor networks, IEEE Communications Surveys and Tutorials, Fourth Quarter, vol. 10, no. 4, pp. 1839, 2015.

[3] Skinner,c.75 of young adults want to vote by SMS in the election.89 expect text voting to be introduced soon.Pcadvisor.February 18, 2010. `http://www.pcadvisor.co.uk/news/index.cfm?newsid=3213010` Accessed in February, 2010

[4] About Android operating system `https://en.wikipedia.org/wiki/Android_(operating_system)`

[5] About Android Studio on Wikipedia `https://en.wikipedia.org/wiki/Android_Studio`

[6] About fingerprint api and integration `https://www.bayometric.com/fingerprint-sdk-api-integration-phones/`

[7] About Sqlite `https://www.youtube.com/watcv=cp2rL3sAFmI&list=PLS1QulWo1RIaRdy16cOzBO5Jr6kEagA07`

[8] About Using fingerprint authentication, `https://www.youtube.com/watch?v=zYA5SJgWrLk`

[9] To know all tool and componets of android development, `https://developers.google.com/`