Fingerprint Voting System

CSC-3460/CSC-6160

Prof. Selvanayaki. K.S

Ronit Prasad: crf\_prasadr@cuchicago.edu

December 4th, 2020

Fingerprint Voting System

CSC-3460/CSC-6160

Prof. Selvanayaki. K.S

Ronit Prasad

December 4th, 2020

Table of Contents:

|  |  |
| --- | --- |
| Title Pages | i, ii |
| Table of Contents | iii, iv |
| Project Management Plan |  |
| Overview | 1.1.1-x |
| Scope and Objectives | 1.2.1-x |
| Assignment of Roles and Responsibilities | 1.3.1 |
| Project Schedule | 1.4.1 |
| Requirements Specifications | 2.1-x.x |
| Analysis | 3.1-x.x |
| Design | 4.1-x.x |
| Implementation | 5.1-x.x |
| Test Documentation | 6.1-x.x |
| Appendices |  |
| Glossary | A.x |
| Source Code | B.x |
| References | C.x |
| Progress Report | D.x |
| Time logs | F.x |

Project Management Plan (Document D1)

* **Overview**: The project objective is to develop a project that can recognize users based on their fingerprint patterns for voting. Since finger print authentication is unique for each person, the system can easily figure out who you are naturally. The person who receives the fingerprints can be known as the “Admin” or “System-Manager.” The admin can add all the names and photos of the candidate based on who they picked to vote for the election. The admin has the ability log into the system and also give users their unique ID and password, so you have the ability to personally vote and check your status. The objective of this system is probably to make information-giving a bit lesson the client side. The clients won’t have to give as much information as usual. They don’t have to put their address in, phone number, age, birthday, SSN, and information. They simply just have a unique username/password and their own unique finger-print.
* **Scope and Objective**: However, even though this method is easier and more effective, it still has a problem: even though client-side information is more manageable and easier; the administration-side is a bit more work. They have to gather a lot of information, store the information, access the information, and organize all the information for each person voting in the United States; or possibly in the world.

Even though this is seen as a negative, but it can ultimately be seen as a positive. Since, the administration has a lot of information storing, it’s likely that this can be done by *one* person, therefore we might need to increase the number of administrators. With the increase of employment, this can create new positions for corporate jobs in software engineering and also solves the problem for the lack of admins and solves the problem for the voting system – [flaw 🡪 solution 🡪 positive growth from flaw].

* **Activities and Dependencies**:

**T1**: Advocate for the change of finger-print voting via, Online, E-mail, Postage mail, flyers, social-media (Snapchat, Facebook, Twitter).

**T2**: Once, we’re done advocating and we actually have thousands of people signing up for the finger-print voting through our online website, we can then proceed to the legal interaction. [Information on people from our Website + Government Database]

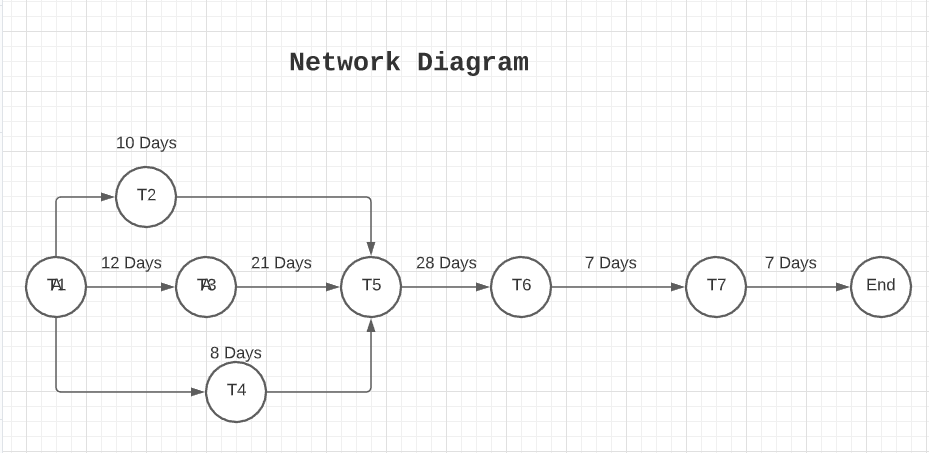
**T3:** If the government allows us to have this personal information of each individual person from their database, based on their finger-print, then we can further continue.

**T4**: Once we have this coherent information about the people who registered to vote and also information about people who will vote/are going to vote (Information from government data-base).

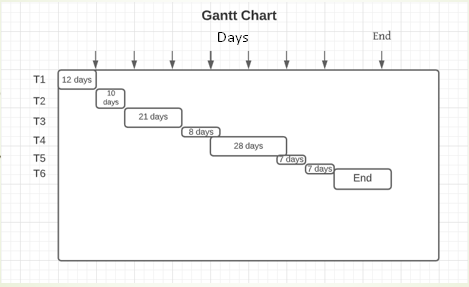
**T5**: The next step would purely be dependent on if there is an election. This would require people to actually attend these stations or hubs, where they will log in into their account and based on their finger-print immediately continue to the next step and just pick who they will vote for.

**T6 + T7**: Repeat process for efficiency between transportation, information, and the other processes.

**Network Diagram**:



**Gantt Chart**:



Requirement Specification / REP (Document D2)

**Requirement Specification:**

* The database information can only be accessed by admins or people who handle the information manipulation.
* The user should be able to pick a party/person to vote and not be able to vote again, since 1 vote = 1 person.
* The information process should be through a set of trials (verification process). If the voter passes the trial, then that person is eligible to be able to vote once and once only. Otherwise, they will need to prove that their information is real and not fake.

**Requirement Engineering Process:**

* Personally, I would document REP (Requirement Engineering Process) to be drafted in four sections: *Elicitation*, *Validation*, *Specification*, and some sort of *Intercession*.

Sections:

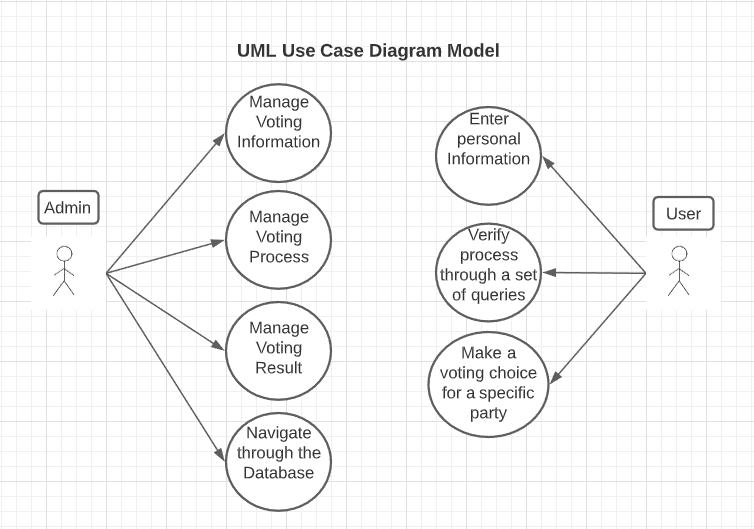
* + *Elicitation*: This would mean how the engineering team talks with the stakeholders and come up with requirements. Whether, something can be usable, readable, deployable, etc.
  + *Validation*: Whether our process, design, or system meets the requirements of the stakeholders. In my case, an example would be, whether the information process achieves everything that both the engineering team wants and the stakeholders.
  + *Specification*: The specifics, processes, and how the processes of the stakeholders interrelate.
  + *Intercession*: The conversation between the stakeholder and the engineering team and the different disagreements that form. Some disagreements could be security problems, how the system interacts with us, or the information process.

Analysis (Document D2)

**Analysis:**

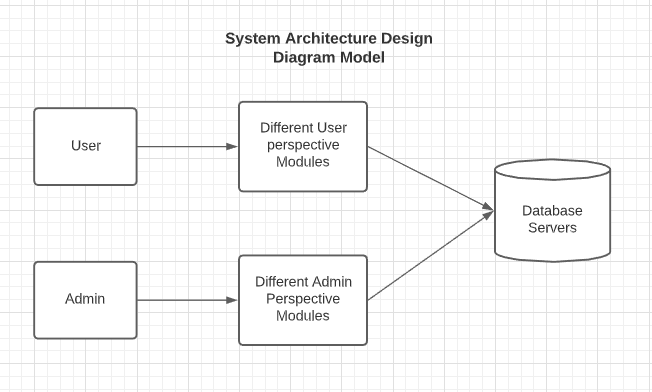
* The voter scans their fingerprint and then all the information gets pulled through our database.
* If the information is false, then we ask them to go through a set of trials for information verification.
* After login, the user will be verified through the database for duplicate information, and then only after that’s cleared, he/she’s vote will be counted as one vote.
  + If the information is a duplicate then we search for any remnant where that person voted multiple times and then clear it as one vote only.
* We then tie up all the votes from everyone who’s voted (*Iff* the votes are cleared to be one vote and one person is associated with one vote).

**UML Use-Case Diagram Model:**



Design (Document D3 and D4)

**Architecture design and interface design**: There are two main modules, the User and the Admin. Both of them have their own specific modules (processes), in which, both of them are directly in link with the Database Servers.

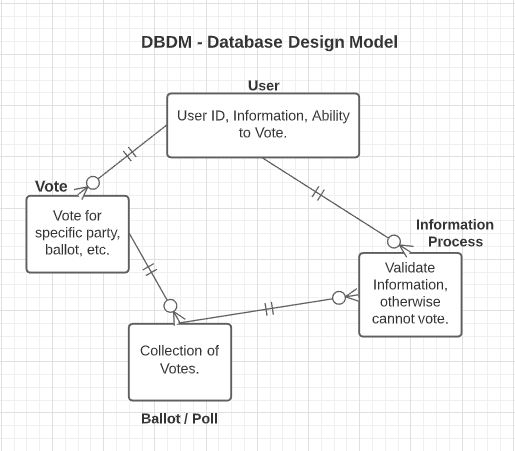


**Database Design + DBDM:** Database Design is how data is organized, stored, and how data elements are correlated to each other. Mostly all DBDM show how they all correlate with each and affect each other.

In my case, the user cannot vote without giving information, but the information has to be processed, and therefore the poll cannot happen and the votes cannot be tallied if we don’t have unique voters.

* *User* – The user would be a string and the user would vote for a specific party, which would be a (string) [letters only / (lowercase/capital letters)].
* *Vote* – Votes would in the form of a number, so it would be an (int).
  1. We would validate the votes based on the person’s information and whether it’s not a duplicate.
* *Ballot –* The collection of votes would be as an (int).
* *Validate Information* – We would check if the person hasn’t voted twice and check if the string matches our constraint.

**Database Design Model:**



Implementation (Document D5)

* **Coding Standards:**
  + Proper declaration of information (public/private), global, local, and other constant variables.
  + Proper header and functions for information – I/O.
  + Syntax Manipulation (indentation, spacing, and proper code etiquette).
  + Debugging. (i.e., Exception Handling).
  + Tracking of modifications. (who does what and why)
  + Testcases (randomly generated or manually).
* **Execution:**
  + I would probably execute step-by-step and learn from each execution and what goes wrong, so that next time that won’t happen.

Test Cases:

* Test Case 1: *Voting for president / nominee*
* Doesn’t Work: Pres9ent Tr3mp
* Works: President Biden

[Should only have Letters (Lowercase/Uppercase Letters)]

* Test Case 2: User Name
* Doesn’t Work: Tes3tUsername3
* Works: TestUser

[Username can only have Lowercase/Uppercase Letters, not numbers as well]

Test Document (Document D6)

**Appendix A:**

* Page 5: Intercession = “Conversation”.
* Page 6: *Iff =* “If and only If”.
* Page 9: I/O = “Input and Output”.

**Appendix B:** N/A

**Appendix C:** Citations:

References

*An Overview of Requirements Elicitation*, www.modernanalyst.com/Resources/Articles/tabid/115/ID/1427/An-Overview-of-Requirements-Elicitation.aspx.

“Biometric Voting System and Voter Registration.” *M2SYS*, 30 July 2020, www.m2sys.com/automated-fingerprint-identification-system-afis-voter-registration/.

**Appendix D:** N/A

**Appendix E:** Time log of project: [~end of October ***<->*** December 3rd, 2020]