

Hybrid Centralized Voting System



Submitted By:

M. Fawad Awan
20584

Supervised By:

Mr. Syed Aizaz Ul Haq
Lecturer

Department of Computing
Abasyn University, Peshawar

October-2023

Department of Computing Project Proposal Submission Form

| | | | | |
|---------------------------------------|--|---------------------------------------|--|------------------|
| Project Title: | Hybrid Centralized Voting System | | | |
| Date of Submission | | | | |
| Project Status (Tick any one) | <input type="checkbox"/> New Proposal | <input type="checkbox"/> Modification | <input type="checkbox"/> Re-Submission | |
| Students' Information | | | | |
| | Student-1 | Student-2 | Student-3 | Student-4 |
| Reg. No. | 20584 | - | - | - |
| Name | M. Fawad Awan | - | - | - |
| Contact No. | +92 3169400169 | - | - | - |
| email | fawadawan183@gmail.com | - | - | - |
| Signatures | | | | |
| To be filled by the supervisor | | | Specify the tools used | |
| 1. | Which method(s) used for requirement gathering? (Sec-2.1) | | | |
| 2. | Which tool(s) used for requirements analysis process? (Sec-2.2 & 2.3) | | | |
| 3. | Which software development model to be followed? (Sec-3) | | | |
| 4. | Which tool(s) will be used for system design? (Sec-4) | | | |
| 5. | Which technologies will be used for the system development? (Sec-5) | | | |
| 6. | Which tool(s) is used for Work Breakdown Structure (WBS)? (Sec-7) | | | |
| 7. | What is each student's contribution in terms of man month as mentioned in WBS? (Sec-7) | | | |
| Name of Supervisor | | Supervisor Signature | | |

Table of Contents

| Topics | Page No. |
|---|----------|
| 1. Introduction and Background of the Project | 1 |
| 2. Requirement Analysis Process | 2 |
| 3. Software Development Process Models | 3 |
| 4. System Design | 4 |
| 5. Technology Requirements | 6 |
| 6. Project Deliverables | 7 |
| 7. Work Breakdown Structure | 9 |
| 8. References | 11 |
| 9. Proposal Evaluation Form | 12 |

1. Introduction and Background of the Project

Elections and voting are an important part of our life, ranging from the opinion gathering via survey forms or a voting over a certain topic in social media sites such as WhatsApp, Facebook, LinkedIn, Reddit etc. to larger firms and corporate organize elections to gather a democratic opinion to come at a consensus over a certain matter/opinion/result.

There're organizations who still follows the manual/paper-stamp method; inevitably to be having an inaccurate, faulty or at worst rigged elections; which I've been a witness of in my time in City University of Science and Information Technology (CUSIT) as a student before migrating here; when the election for students societies president were commenced, the elected student would be president of all student societies, when election's polling started; which was manual, you've to mark your desired person's election symbol; in order to get more votes a candidate managed to bribe the person on the ballot by financial favors; and that person marked two or three votes per every vote casted, thus resulting in that candidate's victory.

To tackle the above-mentioned scenario, we may use a digital approach. **POLYAS** is a platform that can be used for handling corporate level elections. It is a SAAS based online voting solution which includes:

- Online dedicated ballot
- Live voting
- Nomination-based candidates

POLYAS is useful for elections and the result's efficiency but here're its cons:

- There's no direct organizational level monitoring over the election organizer on this platform.
- There is no validation for the voters or candidates being of the organization for which the elections initiates, election entity's identity has to be verified manually by election organizer; thus, opening the gateway for the imposters or, if the election organizer has been bribed, is coerced or forced into doing someone's bidding then he/she may add imposters as voters for the candidate who coerced or forced the election organizer.
- In the nomination-based candidates, the criteria of selection can be vague or ambiguous because the election organizer is selecting the candidates; thus, resulting in a non-transparent process with severe lack of trust by the employees who're voters.
- The election proceeds on a centralized database, which may be subject to some serious data breaches, hacking attacks; and once an attempt is successful, all election data can either be lose, altered or used for extortion.

Due to all above mentioned reasons, there has to be a system which tackle them all, that's why I came up with the idea of a system which will have the following features:

- Organization level auth i.e., the authentication of the person using this system will be based on the person's existence in the database.
- Role-based apps (Voting, Candidates stats, Election Data Admin, System Admin).
- The employees as candidates or voters will be added from the organization's DB by the system, based on a certain criteria/criterion.
- Once vote casted, the voter's voting is done, preventing from re-voting for someone again or undoing the casted vote.
- If a voter/candidate left the organization then he/she won't be the part of the system.
- The voting will be done using blockchain to prevent the tampering or cyber-attack on the voters, their casted votes or on the candidates and their all data including but not limited to their acquired votes [1].
- The organization's database should be able to interact with the blockchain to determine only the candidate who is within the organization should be able to remain a candidate whenever there is election, and voter should only be the employee who is within the organization, of he/she left; then they'll be removed as the voter from the organization's DB and after that, from the blockchain as the voter as well.

This project of mine is a revolutionizing step towards changing the perspectives of the employees, workers and rest of the stakeholders of a company/firm/organization which will enable them all to empower their beneficiaries, employees, employers to get the most transparent yet democratic opinion for the acquisition of a central or a principal administrative role by the most competent person.

I've created the selection criteria for employer as either a candidate or a voter based on his/her attendance, which is the most basic selection criteria; ***this system is a proof of the concept of the system with the above-mentioned features/qualities and is not dedicated or made for a specific organization/firm/company but rather fake data entered in a cloud-database will be used.***

2. Requirement Analysis Process

The requirements are important to be collected properly in order for any system or software to be made, the requirements of this system are corporate oriented, including but not limited to transparency, authenticity and integrity; of the system, its stakeholders and users.

2.1 Gather the requirements

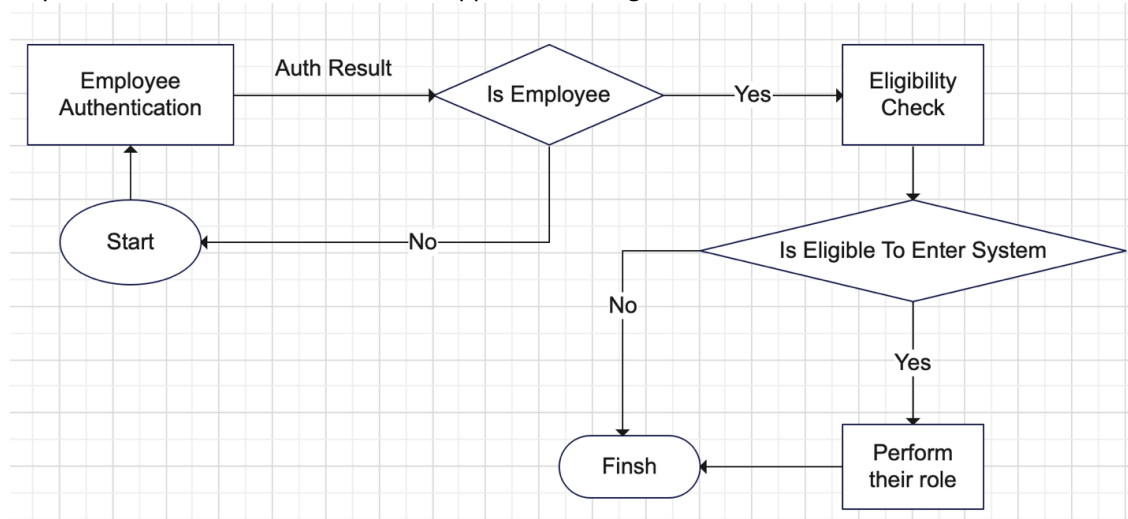
I gathered my requirements via observatory method which I observed first hand in my time at the CUSIT, and some by self-brainstorming. The requirements gathered are:

- Voter and candidate should be able to enter the system, only if they're authorized as the employee and fulfill the condition of entering and using the system.
- The voter after voting will not be able to re-vote; preventing from any rigging scenario.
- The system admin will perform the critical tasks related to system.

- The EDA (Election Data Admin) will be responsible for performing the vetting process of employees to be considered as candidates, and some other peripheral tasks.
- The candidate will be able to see their results only.

2.2 Analyze the requirements

The requirements are to be considered for every app included in the system. I've generalized the requirements for candidate and voter apps as following:

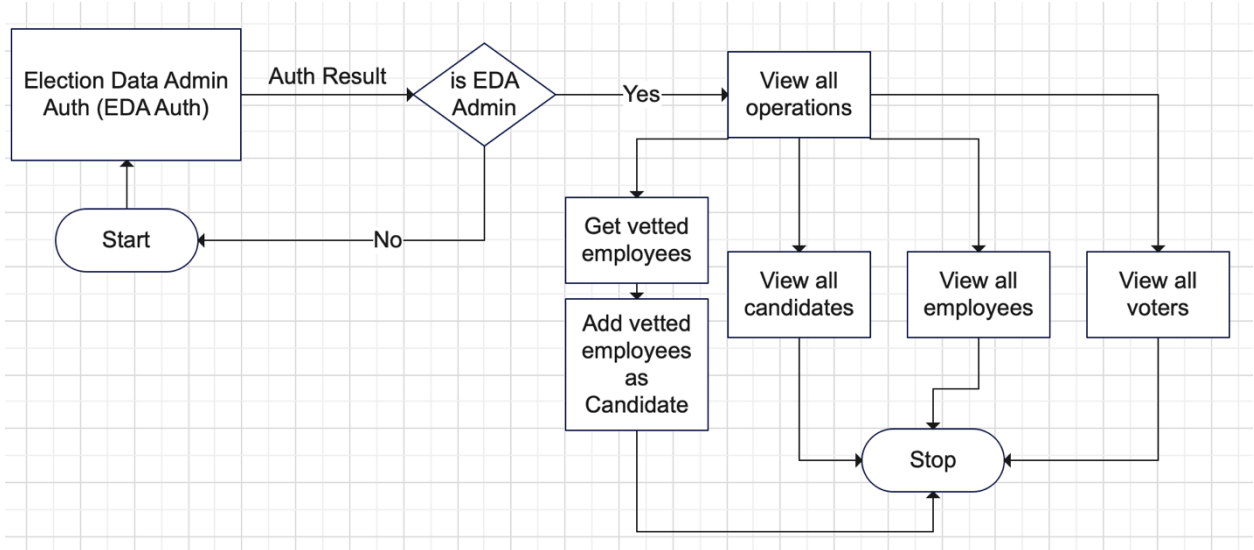


In the above screenshot, you're able to see the following features which are very basic requirements of system:

- Employee Auth & validation
- Eligibility check & validation
- Performing their role
- Finish

Here the validation refers to the bool checks. The process "Perform their role" refers to the roles of the both Voter and Candidate, the prior's role is to cast vote and wait for result and post one's role is to wait for result until election is done.

The requirements, and based on it the workflow of the Election Data Admin is as under:

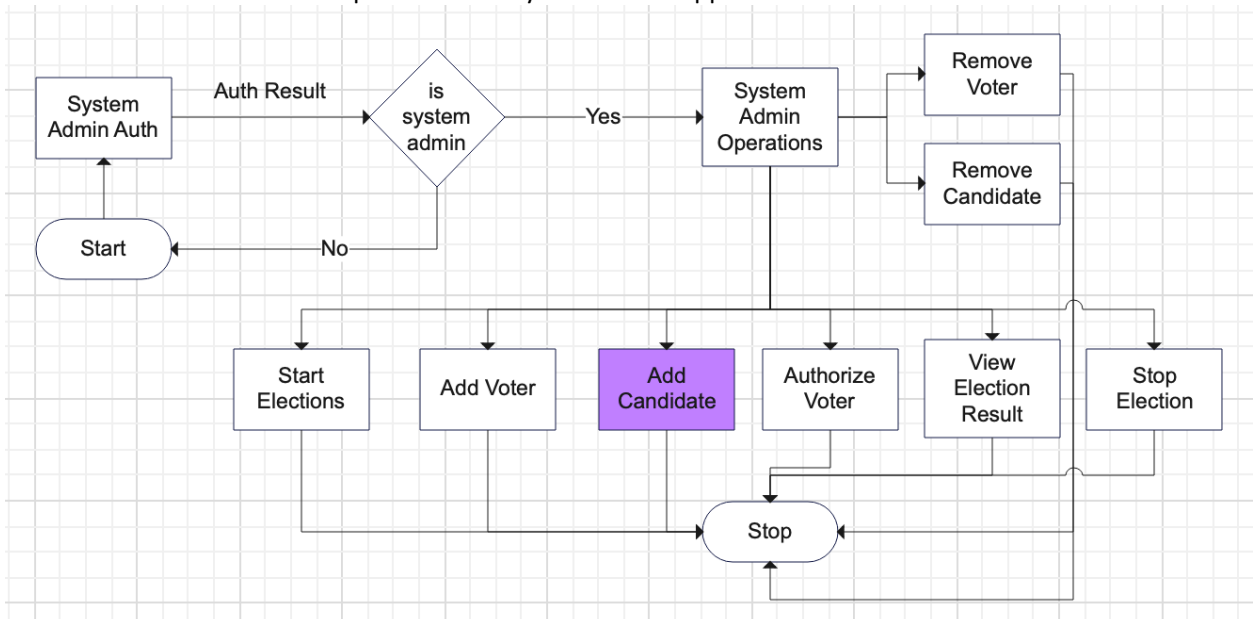


The main task here is to do the vetting of the employees to be considered fit as candidates. The peripheral tasks are:

- View all Candidates
- View all employees
- View all Voters

The significance of the periphery tasks is to make sure everything is going all right.

The workflow based on the requirements of System Admin app is as under:



If you see both EDA and System Admin apps, I haven't added the employee checking auth; because the EDA and system admin are part of the organization but not employees, they're both registered in the organization's DB as EDA and System Admin respectively; and more importantly, both are two different roles.

3. Software Development Process Models

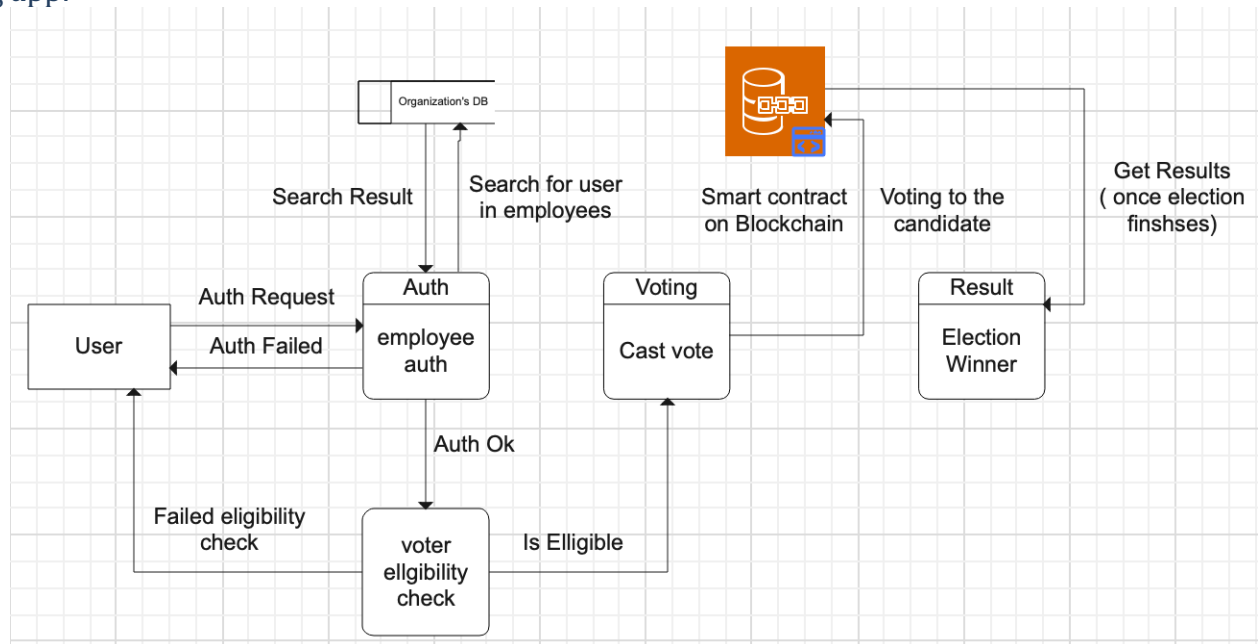
I am using waterfall method, because it goes very well with my project's nature; which is a non-critical project and is based on a non-existing organization's DB, based on the typical entities and their attributes as found in every organization.

4. System Design

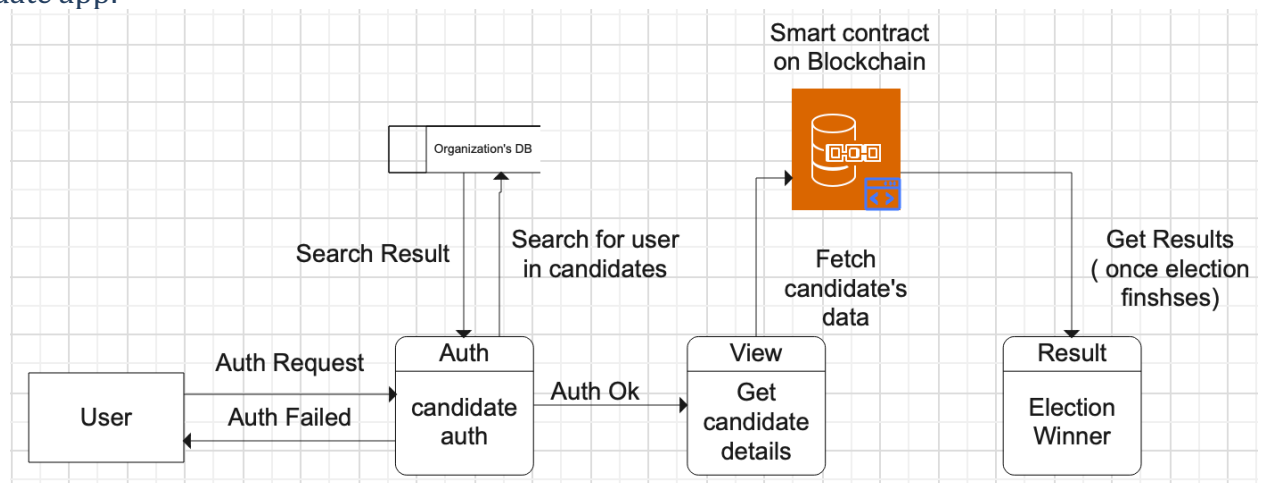
The system processes, data and their relation level views are yielded by DFD, ERD of all apps of the system proposed.

DFD diagrams of all the apps of the system

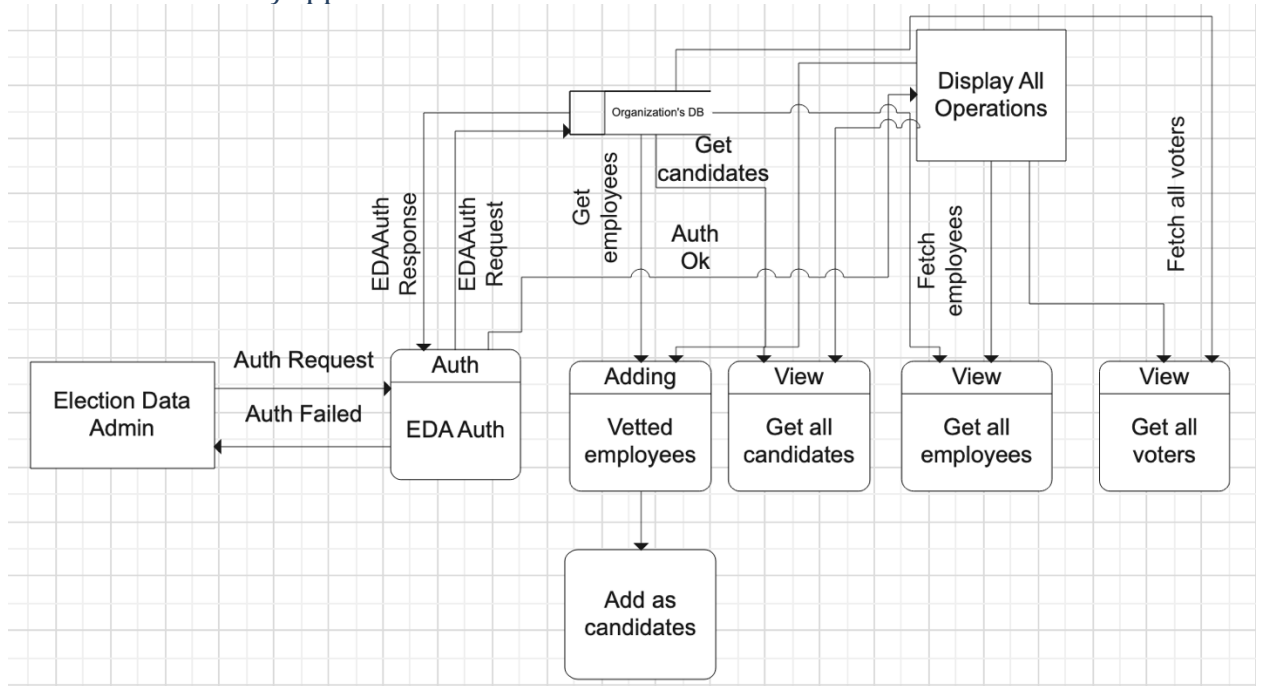
Voting app:



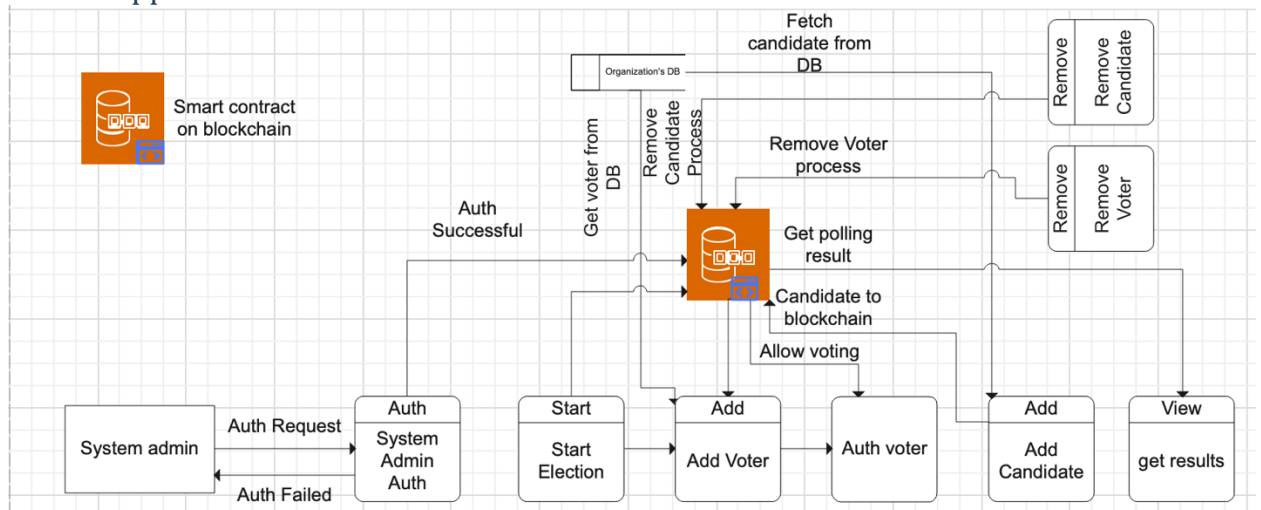
Candidate app:



EDA (Election Data Admin) app:

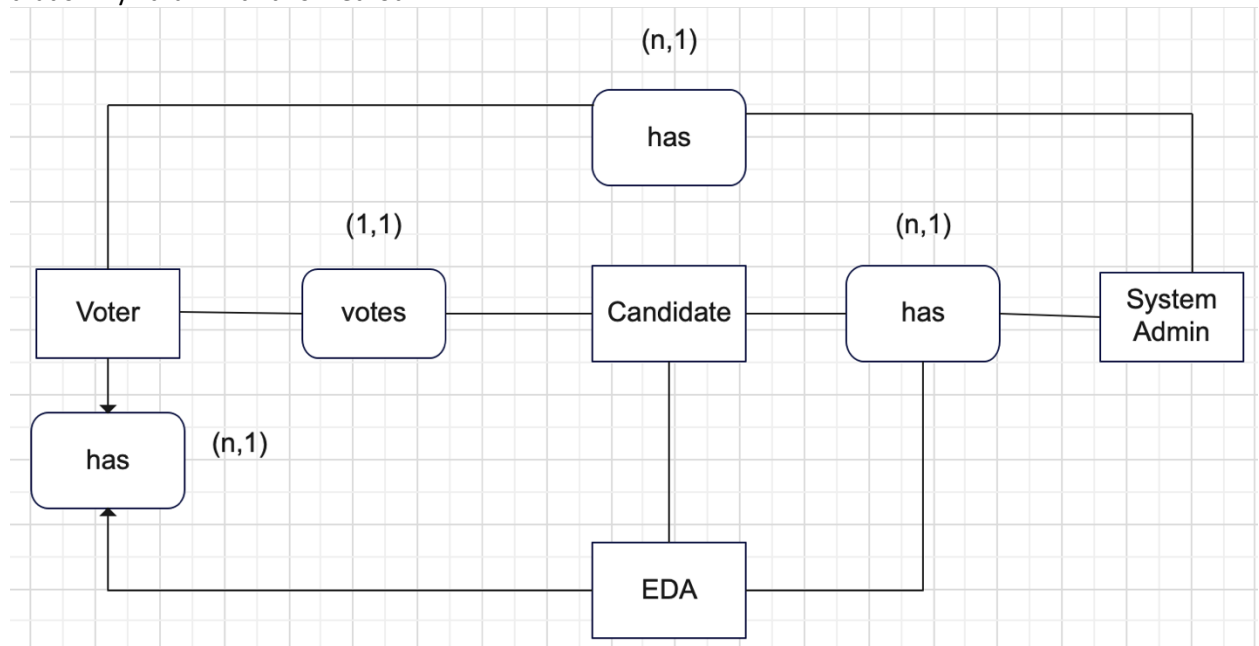


System Admin app:



The ERD to show the relationship among the entities of the system proposed is as under:

NOTE: The crow's foot notation of ERD is now only in premium version of Edraw max, that's why I draw with this method.



- A voter votes for a candidate.
- Many candidates and voters have a system admin
- Many candidates and voters have an EDA (Election Data Admin).

5. Technology Requirements

The system is a set of 4 mobile applications and will be build using flutter framework primarily, the other required technologies are:

- **Dart:** Flutter's default programming language.
- **Node js:** For writing the migrations to deploy the smart contract on local blockchain of Ethereum, provided by the Truffle Ganache.
- **Truffle (node library):** A node library installed globally, to facilitate web3 development and providing local solidity compiler to write the solidity smart contracts.
- **Truffle Ganache (a local blockchain):** A local ethereum blockchain to deploy and test the solidity smart contracts before deploying them in a production level blockchain(s).
- **VSCode:** For flutter app development, developing local smart contract using solidity
- **Solidity 0.8.19:** The version of solidity, which is used in smart contract's developing which is deployed on the sepolia test net. Solidity is the programming language with which you can interact with the ethereum based blockchains.
- **Sepolia Ethereum Faucet:** To get test ethers, which've no market value, are not actual ethers, can be acquired free of cost, and are to be used as gas fee of the smart contract deployment over the sepolia blockchain.

- [Sepolia Ethereum Test-net](#): A sandbox blockchain made for testing of a smart contract, created by Ethereum, where contract deployment occurs, without the actual ethereum(s) deduction as gas fee from your crypto-wallet
- [Remix IDE](#): For solidity compiling and smart contract deploying over sepolia test-net.
- [Metamask \(A crypto wallet\)](#): For storing sepolia test ethers and to deploy the smart contract over the sepolia testnet, by adding it to the remix IDE as an ether wallet
- [Firebase](#): As a central database for the non-existing organization's data storage.
- [Wondershare EdrawMax](#): For UML diagrams creation.

Note: The space is left empty intentionally, in order for the project deliverables and WBS screenshot to be elegantly understandable.

6. **Project deliverables**

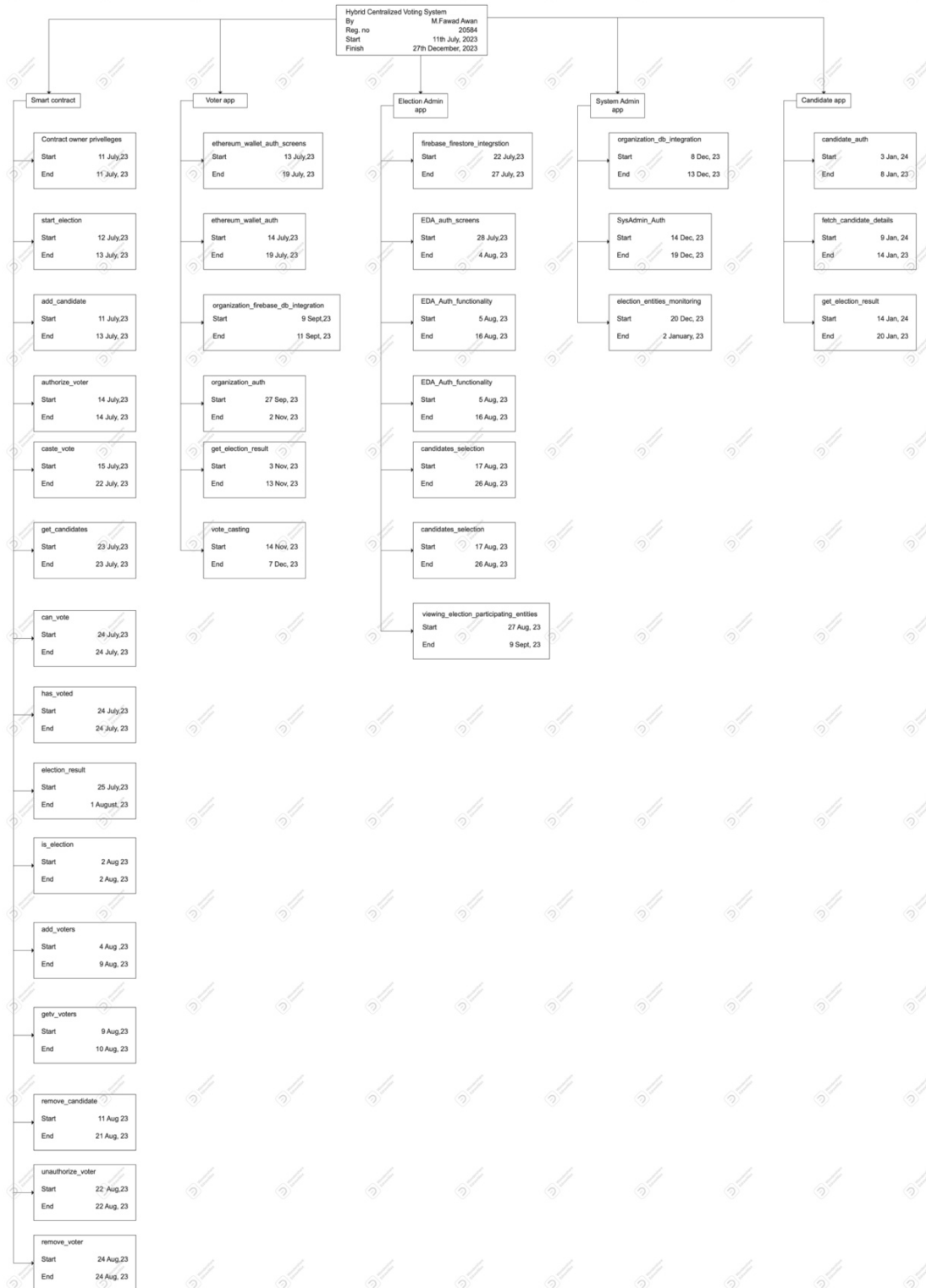
The work breakdown structure is as following:

The project's deliverables along their sub-tasks with their start and end dates are included to give the idea of the work schedule



7. Work Breakdown Structure

The WBS of my FYP is as under:



8. References Bibliography

- [1] M. J. A. A. a. Z. S. Uzma Jafar, "National Center of Medicine," 31 August 2021. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8434614/>.

(For Office Use Only)
Department of Computing
FYP-1 Evaluation Form

Student Name(s):

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

Project Title:

| |
|--|
| |
|--|

Evaluation Committee Member's

| | | |
|--------------------------------------|---|-------------------------------------|
| Remarks: | | |
| | | |
| | | |
| <input type="checkbox"/> Recommended | <input type="checkbox"/> Deferred for modifications | <input type="checkbox"/> Re-Present |
| Name of Member-1 | Signature | Date |
| | | |

| | | |
|--------------------------------------|---|-------------------------------------|
| Remarks: | | |
| | | |
| | | |
| <input type="checkbox"/> Recommended | <input type="checkbox"/> Deferred for modifications | <input type="checkbox"/> Re-Present |
| Name of Member-2 | Signature | Date |
| | | |

| | | |
|--------------------------------------|---|-------------------------------------|
| Remarks: | | |
| | | |
| | | |
| <input type="checkbox"/> Recommended | <input type="checkbox"/> Deferred for modifications | <input type="checkbox"/> Re-Present |
| Name of Member-3 | Signature | Date |
| | | |

Committee Head

| |
|----------|
| Remarks: |
| |

| | | |
|--------------------------------------|---|-------------------------------------|
| | | |
| <input type="checkbox"/> Recommended | <input type="checkbox"/> Deferred for modifications | <input type="checkbox"/> Re-Present |
| Head of Committee Name | Signature | Date |
| | | |