SECURE E VOTE

Anytime | Anywhere

Participants :

Shivam Marathe	[111708035]
Utkarshbhanu Andurkar	[111708073]

Guides :

Shirish Gosavi Sir
Tanuja Pattanshetti Ma’am

Work proof:

You can check live website here
You can check github repo here

Problem Statement

Create a secure and efficient online voting system.

- In India, there are institutes, colleges and organizations where *Democracy* exists.
- Thus our vision is to provide them the online E voting system which can help them to make required *decisions* and take off the *overhead of arrangement of elections*.
- Even when on-going situation occurs like *corona virus*, online voting systems are spot on solutions.
- Officials can monitor the election using their desktops.

SECUREEVOTE

SCOPE

- This project is developed for voting inside *organizations* / institutions only.
- This is a *three layer architecture* where it involves Admin, Polling Officers and Voters. Cryptography is the basis and domain of this project.
- This is a *complete full stack* application. Our audience / clients will enjoy neat, clean and responsive UI alongside with *secure, robust* and simple procedure to conduct elections.

INTRODUCTION

- This project aims to develop a secure e-voting system to create *any-time, anywhere* voting possible.
- This is a *web based application*.
- The concern of trust deficit of people in online voting is addressed by providing security and confidentiality through *Steganography* and *Visual Cryptography.*
- Voter's *privacy* and individual *verifiability* is at heart of our project

REQUIREMENTS

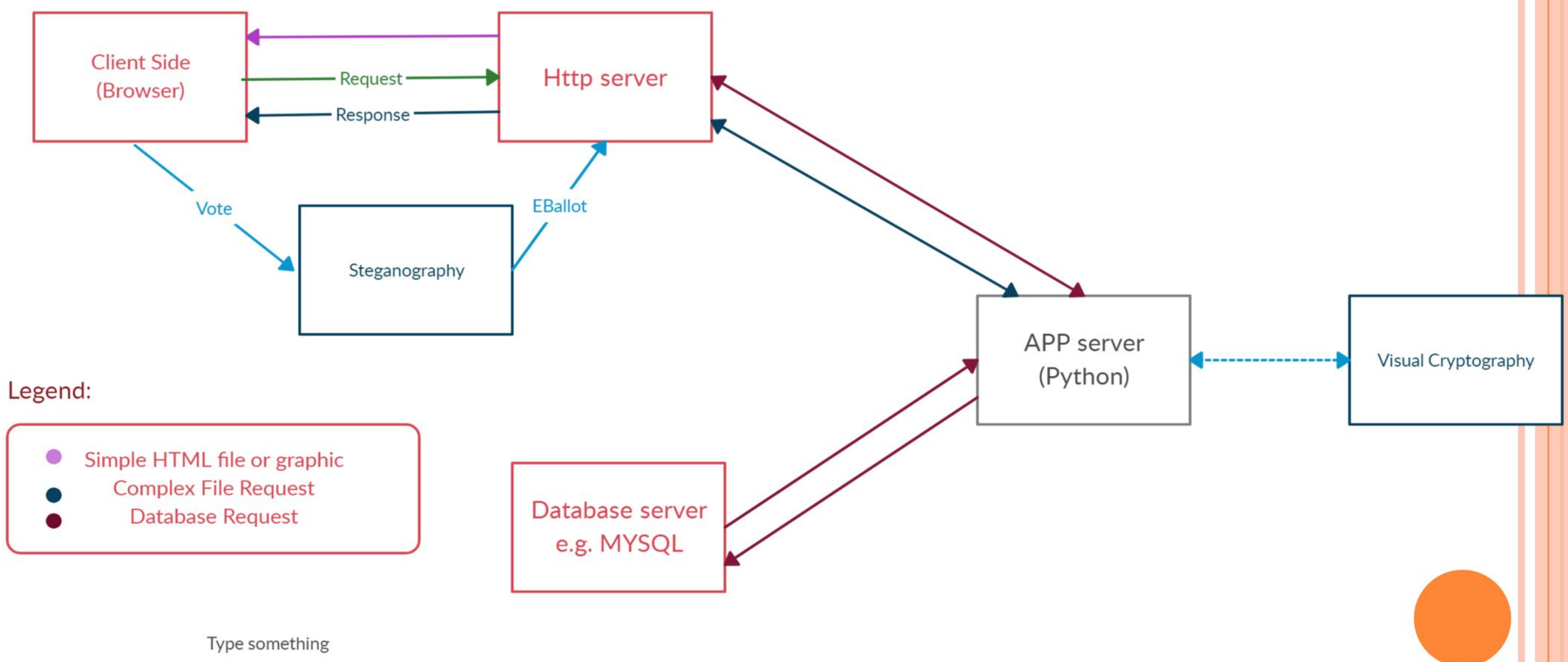
- Anytime Anywhere voting is possible.
- Information regarding voter and his vote should be kept *confidential*.
- Administrator should have lists of voters, P.O's before the election gets registered with SEV.
- Make vote coercion nearly impossible.
- Provide easy to use and elegant *user interface*.
- Declaration of election results should be robust and accurate.
- Voter can keep share of his vote to keep allegations away and so is Admin and each P.O with the use of VC and TC
- REST API

DESIGN

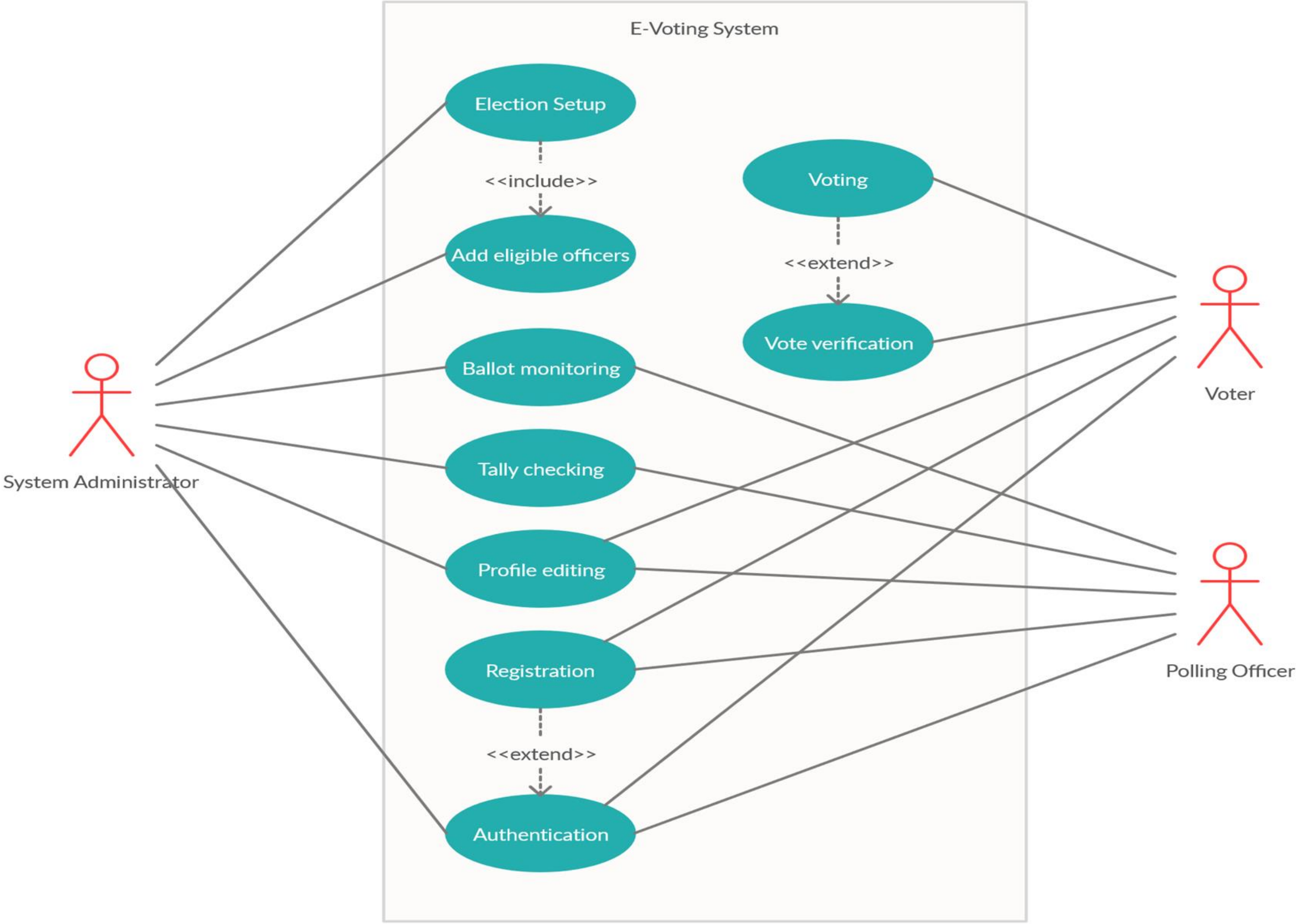
- We used *iterative waterfall* model to design this system.
- We tried to keep this system as *less complicated* as possible.
- How we have used Iterative Waterfall model:
  - Customer *Interaction* at the start only.
  - *Feedback Path* handles errors consciously.
  - *Simple* in Practical Software Projects. For Eg. One team member *can't change* requirements when it is distant project development. Ref: Lockdown.
- Following couple of slides contain use case diagram and state diagram which will help use understand design of Secure E-Vote[SEV] easily.
- Software used for all project diagrams : creately.com

Design schema 1:

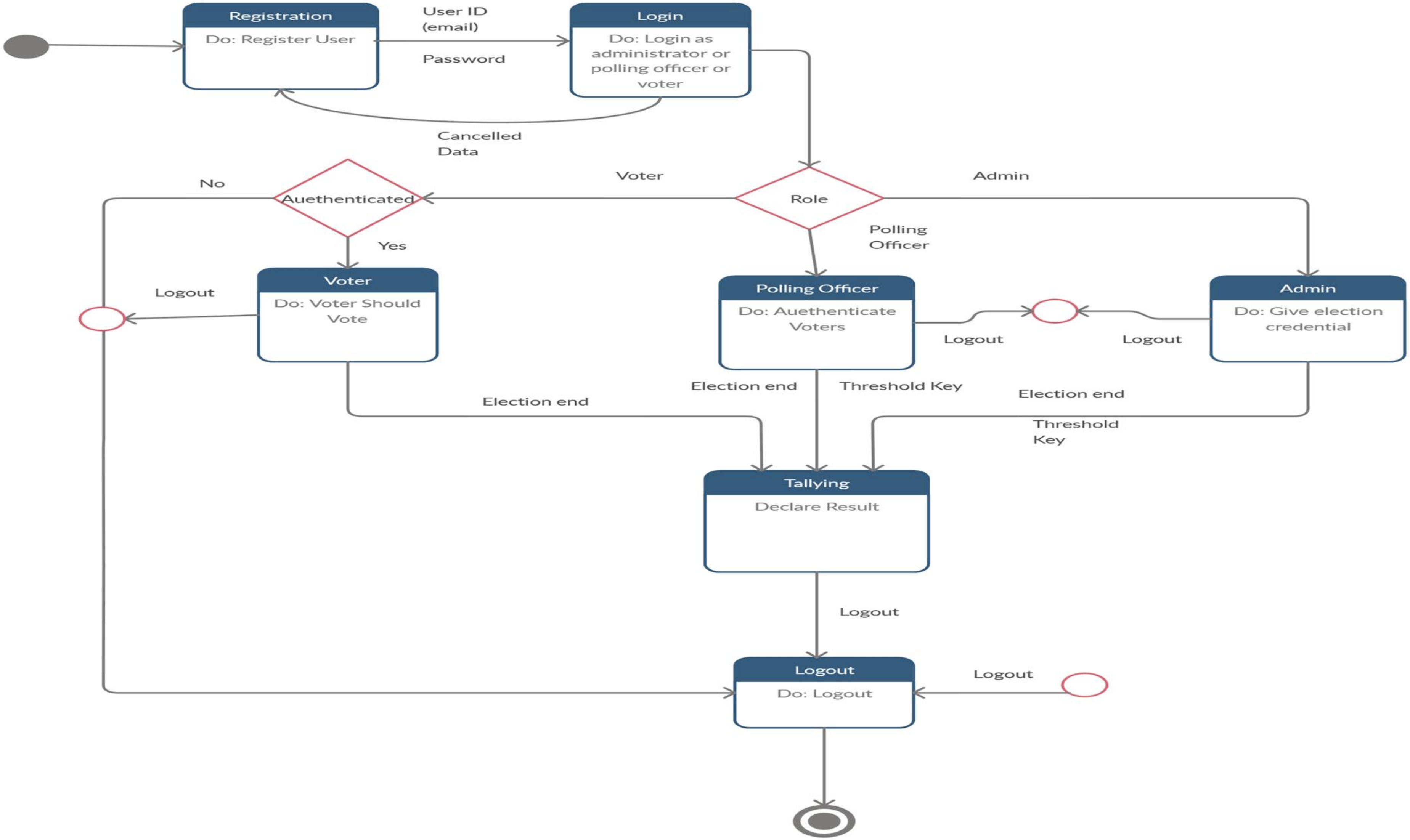
PROPOSED SYSTEM ARCHITECTURE



DESIGN SCHEMA 2



DESIGN SCHEMA 3



SECUREEVOTE

IMPLEMENTATION

Front End	Back End	Database	API
HTML 5		SQLite	REST API
CSS 3	django		B
JS			AJAX

LEARNING OUTCOMES AND CHALLENGES

Outcomes	Challenges Faced
We were unknown about the SDLC cycle. We learned it through Theory lectures	Deadline management. Due to lockdown we weren't able to contact that frequently. (8 days delay to 11 April)
We used Latex for professionals for writing report / thesis	JavaScript and Ajax use and its integration, Django use cases.
We learned technology stack and full stack development, Image processing and Cryptography,	Latex workshops, Managing Requirements.
Algorithms, Team skills and distant communication	Keeping Repository up to date. Test cases take a lot of time.

SECUREEVOTE

CONCLUSION

- Secure E-Vote system shows that it is possible to provide a secure *voting service using Internet*.
- Secure E-Vote has potential to reduce *expenses* for elections. This include both money and human resource.
- Fair and limited use of resources by this system proves it’s *efficiency*.
- Secure E-Vote System provided *elegant* yet easy to use voting service.
- *Tutorials* and *FAQ*s provided with the system tries to answers most queries asked by users. Result of ” *User acceptance*” testing support these conclusions.
- The scope of this system is limited to elections within an organization. Using this system in large elections like elections in state/country may cause scalability issues.
- It *generates report* which is demand for every system.

SECUREEVOTE

FUTURE SCOPE

- We are thinking to use *BlockChain* technology to make *database* more secure and reliable.
- The *scalability* of secure E-Vote is also a major concern. We would like to improve scalability of this system so that it can be used in large scale elections.
- To improve scalability we are thinking of using *client side computation*.
- More *attributes* and *features* to this prototype can be added.