

**E-Voting Portal**

Software Architecture and Design Document

**Group Members -**

**1.G Srisha Anagh (IIT2016030)**

**2.Sai Charan Teja (IIT2016039)**

**3.Anmol Singh Sethi (IIT2016040)**

**4.Manavdeep Singh (IIT2016042)**

**5.Nairit Banerjee (IIT2016505)**

Table of Contents

1. **Introduction**
   1. [Purpose](https://docs.google.com/document/d/1yjqFpE4YH0lF5BYiOd3ItK2Xt6zYPHNpQ_BG3FDisdE/edit#heading=h.j0iwe8bu959p)
   2. [Scope](https://docs.google.com/document/d/1yjqFpE4YH0lF5BYiOd3ItK2Xt6zYPHNpQ_BG3FDisdE/edit#heading=h.il8pe19p68wh)
   3. [Intended Audience](https://docs.google.com/document/d/1yjqFpE4YH0lF5BYiOd3ItK2Xt6zYPHNpQ_BG3FDisdE/edit#heading=h.p6zyx2bt7uxa)
   4. [Document Overview](https://docs.google.com/document/d/1yjqFpE4YH0lF5BYiOd3ItK2Xt6zYPHNpQ_BG3FDisdE/edit#heading=h.skrszm4hjgjk)
2. **Architectural Design**
   1. System Components
      1. User Component
      2. Admin Component
   2. Component Interface
      1. User Interface
      2. Admin Interface
   3. Architectural Model
3. **Component and Detailed Design**
   1. Technologies Used
   2. Use-case Diagram
   3. Class Diagram
   4. Activity Diagram
   5. Object Diagram
   6. Sequence Diagram
   7. State Chart Diagram
   8. Component Diagram
   9. Collaboration Diagram
   10. Deployment Diagram
   11. E-R Diagram
   12. Database Design
   13. References
4. Introduction

## 1.1. Purpose

“E-Voting Portal” is a software which ideally conducts voting in a completely digital manner.

This is a portal which is mobile in sense of it’s usage, i.e. it can be used anywhere by hosting it on a system and a wide range of people can use it to solve the purpose of standing in long queues to cast their vote when the same can be done from the comfort of their homes in a single click. Also, no lengthy forms have to be filled for contesting the elections as nominations can also be done online.

This software design specification document is created with the purpose to give an insight of the basic architecture and design of the system “E-Voting Portal”, which will guide the developer to fulfill the requirements of the user effectively and efficiently. The document provides different views of the system to enhance developer’s understanding of the overall functioning of and user expectations from the system.

## 1.2. Scope

The software design specification document will demonstrate the design as specified by the functional and non-functional requirements in the Software Requirement Specification(SRS) document. It will give an idea about the components, architecture and database design of the “E-Voting Portal” using the design patterns, sequence diagrams, class diagrams, etc.

The “E-Voting Portal” is a system in which a database will be created to store all the information entered by the user. Once verified, the person can nominate himself to be a candidate in the elections. Other general public must then log in using their aadhaar card details and vote for the suitable candidate. The winner is then announced in the results. The details are very securely stored using SHA256 which is a very popular encryption tool used in cryptography and blockchain.

## 

## 1.3. Intended Audience

This document is mainly for the developers to understand the basic requirements of the user and how the system should work. And for any user who intends to use the system, can consult the document and update his/her requirements from the system if needed.

## 1.4. Document Overview

The next sections of the Software Design Specification Document include the Architectural Design, Components and Detailed Design of the system which will be depicted through Use Case Diagram, Class Diagram, Activity Diagram and ER Diagram.

1. Architectural Design

# 2.1. System Components

## 2.1.1. User Component

This component is responsible to take information from user as input and storing that information correctly in database. Also for the candidates a dashboard should be displayed with all their information duly displayed.

## 2.1.2. Admin Component

This is a background component which is not visible to system users. This component is responsible for continuously querying with the server and verifying the candidates. Also, it is responsible to select election officers in a particular constituency.

# 2.2. Component Interface

## 2.2.1. User Interface

* Verify() - If all the data of the user is correct according to standards set then enter into the database.
* withdraw() - withdraw the nominee from candidature.
* login() - This directs to the html page which allows entries like aadhaar number and password and redirects towards dashboard.

## 

## 2.2.2. Admin Interface

* Remove\_electionofficer() - Delete the entry of the election officer from the database. Also can view the number of voters registered etc.
* Show\_Hideresults() - This would enable/disable the view Results option for voters and candidates.

# 2.3. Architectural Model

The proposed architectural model that we are going to implement in our application is Client-Server Model.

**Client:** The Client side in our model is User and Admin where user inputs the data about him/herself. If the user wishes to be a candidate, they have to upload an image etc required. Admin verifies candidates and also has the powers to start/stop elections.

**Server:**The Server side in our model is implemented by a database which stores all the information of the Candidates, users etc and is interfaced with Admin Component in order to respond to any information query.

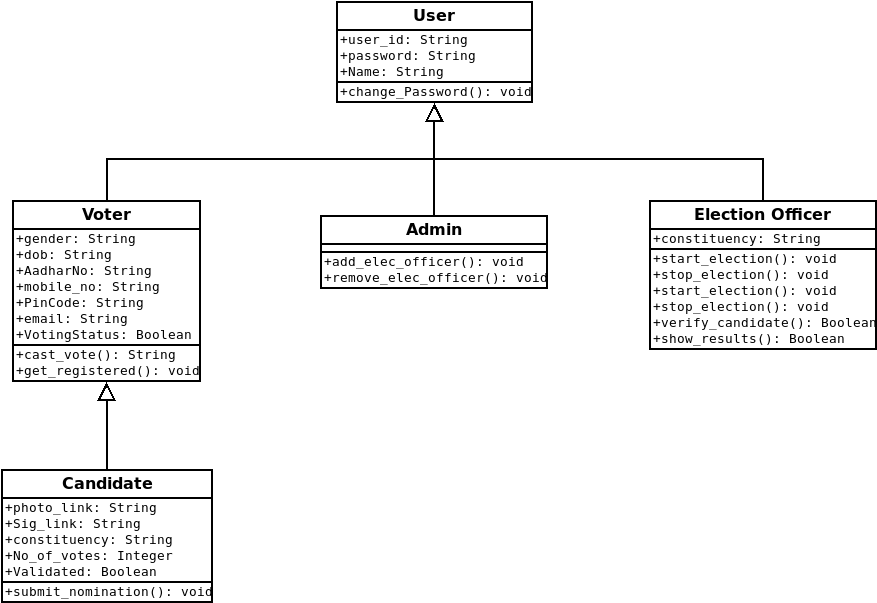
1. Component and Detailed Design

## 3.1. Technologies used

* **Front end**:HTML, CSS, JavaScript, jQuery
* **Back end**: Flask - a framework of python

## 3.2. Use Case Diagram

## 3.3. Class Diagram



## 3.4. Activity Diagram

## 3.5. Object Diagram

## 3.6. Sequence Diagram

## 3.7. State Chart Diagram

## 

## 

## 

## 

## 

## 

## 3.8. Component Diagram

## 

## 

## 

## 

## 3.9. Collaboration Diagram

**VOTER**

|  |  |
| --- | --- |
| **Responsibilities** | **Collaborators** |
| Cast a vote  Get himself/herself registered | Candidate |

**CANDIDATE**

|  |  |
| --- | --- |
| **Responsibilities** | **Collaborators** |
| Submit nomination  Cast a vote | Voter  Election Officer |

**ELECTION OFFICER**

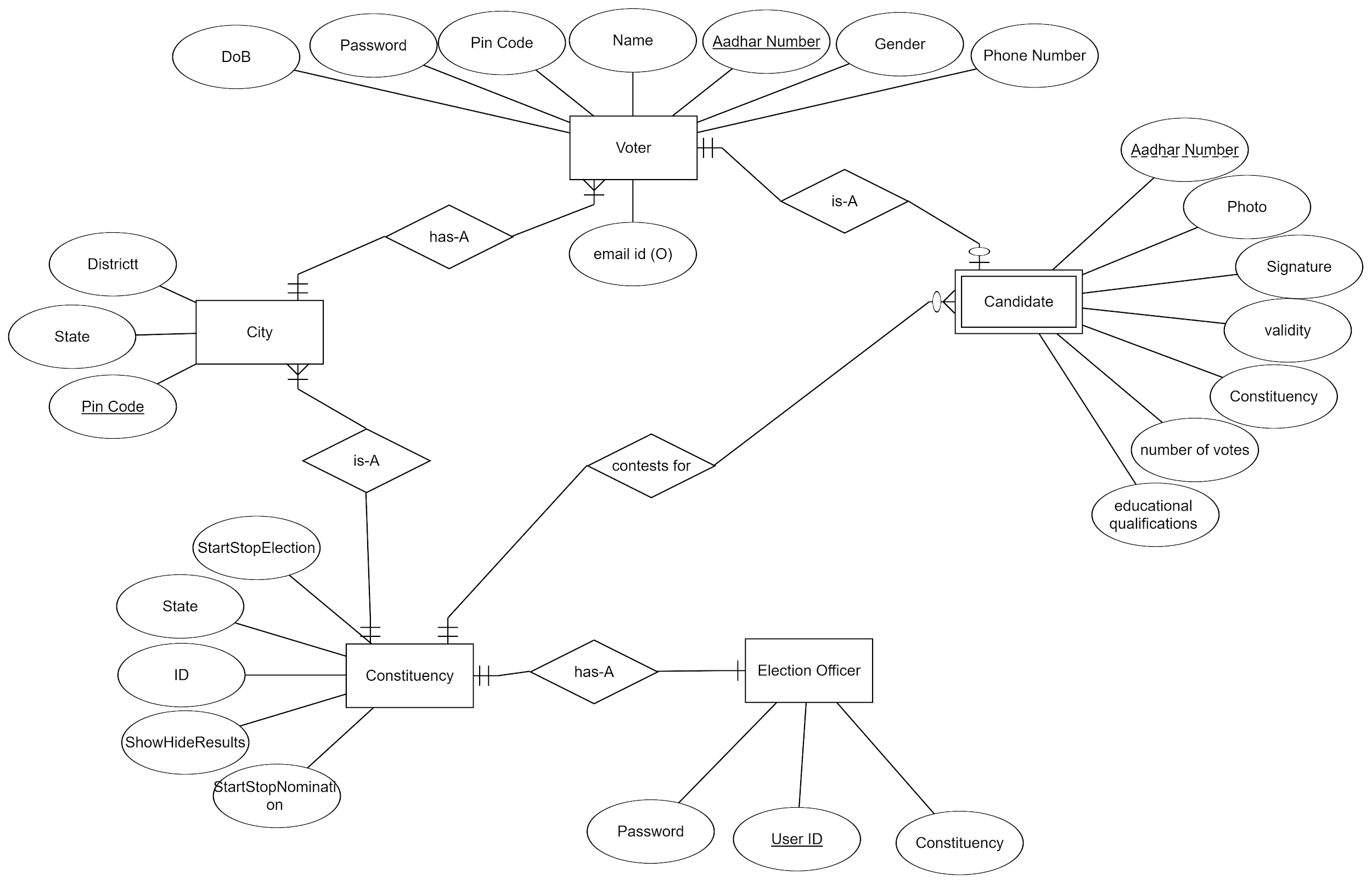
|  |  |
| --- | --- |
| **Resposibilities** | **Collaborators** |
| Start or Stop election  Start or Stop nomination  Verify a candidate nomination  Show election results | Admin  Candidate |

**ADMIN**

|  |  |
| --- | --- |
| **Responsibilities** | **Collaborators** |
| Add election officers  Remove election officers | Election Officer |

## 3.10. Deployment Diagram

## 3.11. E-R Diagram



## 3.12. Database Design

* **Voter Table** :**-**

|  |  |
| --- | --- |
| **Field** | **Datatype** |
| Name | VARCHAR |
| Gender | VARCHAR |
| DateOfBirth | DATE |
| AadhaarNumber | VARCHAR |
| PinCode | VARCHAR |
| MobileNumber | INT |
| EmailId | VARCHAR |
| Password | VARCHAR |
| VotingStatus | VARCHAR |

* **City Table :-**

|  |  |
| --- | --- |
| **Field** | **Database** |
| PinCode | VARCHAR |
| City | varchar |
| State | varchar |

* **Candidate Table :-**

|  |  |
| --- | --- |
| **Field** | **Database** |
| Aadhaar Number | varchar |
| PhotoLink | varchar |
| SignatureLink | varchar |
| EduQua | varchar |
| Constituency | int |
| NumberOfVoters | int |
| Validate | int |

* **ElectionOfficer Table :**-

|  |  |
| --- | --- |
| **Field** | **Datatype** |
| UserID | varchar |
| Constituency | varchar |
| Password | varchar |
| ShowHideResults | int |

* **Constituency Table :-**

|  |  |
| --- | --- |
| **Field** | **Datatype** |
| State | varchar |
| ID | int |
| StartStopElection | int |
| StartStopNomination | int |

## 3.12. References

* + **Software Requirements Specifications (SRS) of E-Voting Portal.**
  + [**https://en.wikipedia.org/wiki/Unified\_Modeling\_Language**](https://en.wikipedia.org/wiki/Unified_Modeling_Language)
  + [**https://en.wikipedia.org/wiki/Client%E2%80%93server\_model**](https://en.wikipedia.org/wiki/Client%E2%80%93server_model)
  + **https://www.tutorialspoint.com/flask - Flask Tutorial**
  + **http://flask.pocoo.org/docs/1.0/tutorial/ - Flask Setup implementation**
  + **https://www.youtube.com/watch?v=MwZwr5Tvyxo&list=PL-osiE80TeTs4UjLw5MM6OjgkjFeUxCYH - Building web Application**
  + [**https://www.youtube.com/watch?v=cp2rL3sAFmI**](https://www.youtube.com/watch?v=cp2rL3sAFmI) **- database links**