Dharmsinh Desai University, Nadiad

Faculty of Technology

Department of Computer

Engineering B. Tech. CE Semester-IV

Subject:-Software Engineering Project

Project title: I-VOTING SYSTEM

By:

1)Sagar Vala, Roll no: CE146, Id: 19CEUOG081

2)Prince Vanani, Roll no: CE147, Id: 19CEUES108

3) Meet Vaishnani, Roll no: CE145, Id: 19CEUEG165

Guided by: Prof. Pinkal C. Chauhan.

Contents: -

1.)Abstract :

online voting system that facilitates user(voter), candidate and administrator (who will be in charge and will verify all the user and information) to participate in online voting. our online voting system is highly secured, and it has a simple and interactive user interface. The proposed online portal is secured and have unique security feature such as unique id generation that adds another layer of security (except login id and password) and gives admin the ability to verify the user information and to decide whether he is eligible to vote or not. It also creates and manages voting and an election detail as all the users must login by user name and password and click on candidates to register vote.

2.) Introduction:

In online voting system a voter can use his/her right online without any difficulty .provides secure voting system.he/she has to be registerd first for him/her to vote.Registration is done by the system admin for security reasons.After registration,the voter is assigned username,password with which he/she can use to log into the system and enjoy services provided by the system such as voting ,result.if invalid/wrong details are submitted ,then the voter is not registerd to vote. If user vote already then again vote it will be not registerd to vote and give error.

3.) Software Requirement Specification

I-Voting System:-

1.Student voter

Description: -in this module we manage student voter detail. student can log-In, sign-up, change password, vote in election. student enter record while sign-up

store into department database. student updated data also be overlap old data.

R.1.3 sign up

Description: if student is new for this system then first need sign-up for log-in.

input: student fill all information.

output: successfully signup message.

R.1.2 login

Description: Login can be done if student's information is stored in the system.

input: enter login detail username and password

output: Successfully login

status: failure if invalid detail entered.

R.1.3 Forgot password

Description: if any student forgot/change password.

input: enter new password.

output: successfully change password.

R.1.4 Update student profile

Description: if any student wants to update his/her profile.

input: enter new profile details.

output: change has been successfully stored.

R.1.5 vote in election

Description: in this function student can vote to any candidate when election conduct.

Input: select candidate.

Output: vote for selects candidate will be counted.

Status: if student invalid than vote not be counted.

2. Election details

Description: in this module we can see election detail like election result, election calendar. election result conduct current year result. also we can show past year result.

R.2.1 display election result

Description: in display result shows who win election.

input: click on result button.

output: show result of all election which conduct in current year.

R.2.1.1 voting statistics

Description: how many percentage voting and how many votes get all candidate of selected election.

input: select voting statistics

output: show voting percentage.

R.2.3 Election schedule

Description: it is show when the election will take place.

input: select or click.

output: show election schedule.

3.host

Description: -in this module host can create election. election detail send to all eligible student. host also can select student candidate. host also display election result. election result store into database.

R.3.1 select candidate

Description: host will select candidate for election. if he/she fill candidate form.

input: add/update candidate information.

output: message all candidate who selected.

R.3.2 send election detail

Description: -host send election details like election id, candidate name to all eligible student.

Input: -select student record.

Output: -student get email for election.

R.3.3 update election result

Description: host will update after voting is over and declare winner.

input: update election result

output: successfully updated

R.3.3.1 update voting statistic

Description: host will update after voting is over. how many votes each candidate got and what percentage of votes were cast etc...

input: update voting statistics

output: successfully updated

R.3.4 store election record.

Description: - in this function host will store election record into database

input: -select election record.

Output: -election record will be store.

4.candidate

Description: -in this module candidate can fill candidate form if selected to become candidate, update their profile.

R.4.1 fill candidate form.

Description: -in this function candidate enter their detail.

Input: enter valid all information.

Output: -get conformation message.

R.4.2 update candidate profile.

Description: -in this function candidate can update their profile.

Input: -enter new detail.

Output: -profile updated.

5.admin

Description: -in this module system admin provide different accessibility for host and student voter. host can create election but student cannot create election. admin also can delete any student or host record if needed.

R.4.1 delete student or host record.

Description: -in this function admin can delete any record.

Input: -enter detail of record

Output: -record detail

status: -failed if record not found.

R.4.2 create election calendar.

Description: -in this function admin create election calendar.

Input: enter election detail.

Output: -election calendar will be display.

R.4.3 add host for any faculty

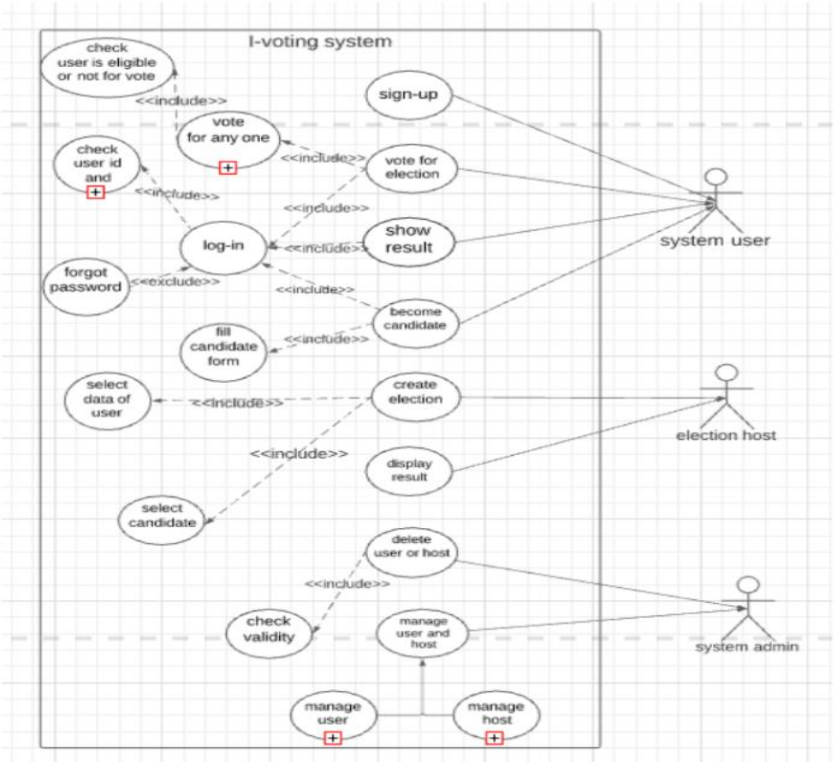
Description: - in this function admin can add host for faculty.

Input: host detail enters.

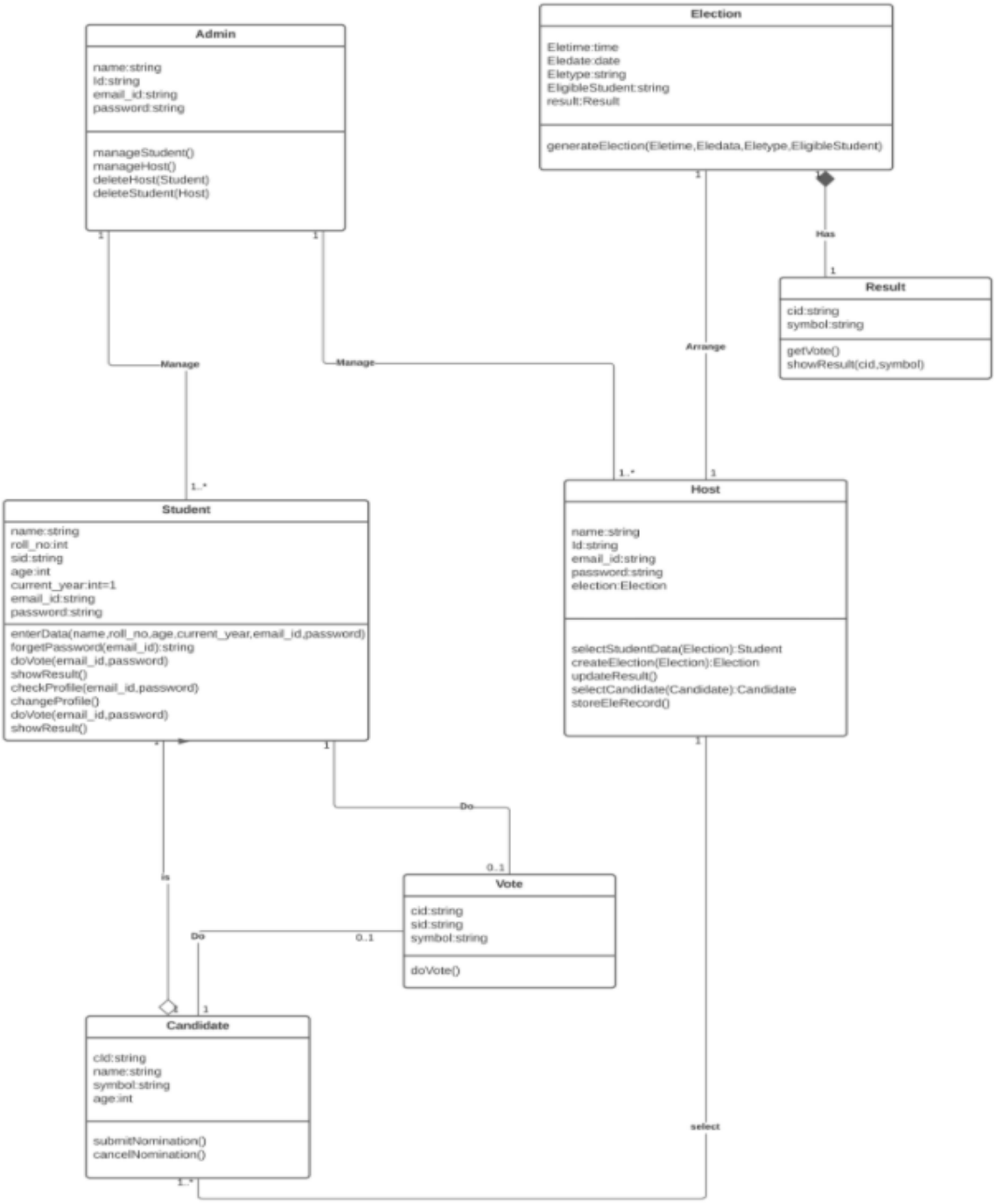
Output: host become successful.

4.) Design

4.1) Use Case Diagram:

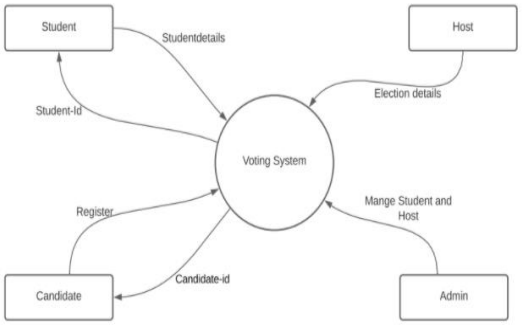


4.2) Class Diagram:

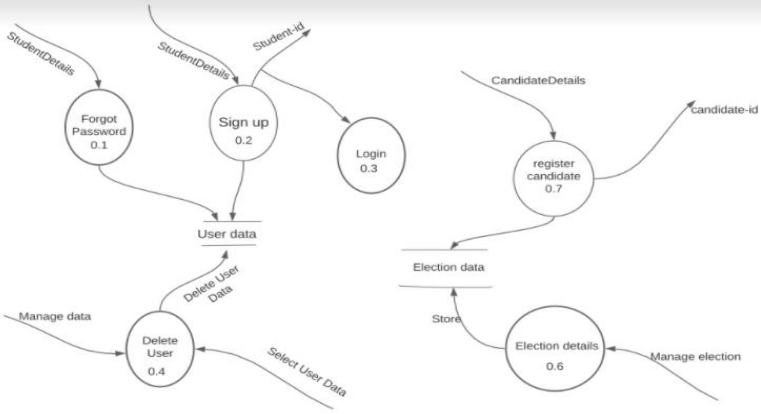

Admin
name:string
Id:string
email_id:string
password:string
manageStudent()
manageHost()
deleteHost(Student)
deleteStudent(Host)
Election
Eletime:time
Eledate:date
Eletype:string
EligibleStudent:string
result:Result
generateElection(Eletime,Eledata,Eletype,EligibleStudent)
Has
Result
cid:string
symbol:string
getVote()
showResult(cid,symbol)
Arrange
Manage
Manage
Student
name:string
roll_no:int
sid:string
age:int
current_year:int=1
email_id:string
password:string
enterData(name,roll_no,age,current_year,email_id,password)
forgetPassword(email_id):string
doVote(email_id,password)
showResult()
checkProfile(email_id,password)
changeProfile()
doVote(email_id,password)
showResult()
Host
name:string
Id:string
email_id:string
password:string
election:Election
selectStudentData(Election):Student
createElection(Election):Election
updateResult()
selectCandidate(Candidate):Candidate
storeEleRecord()
Do
Vote
cid:string
sid:string
symbol:string
doVote()
Do
is
Candidate
cid:string
name:string
symbol:string
age:int
submitNomination()
cancelNomination()
select

4.3) Data Flow Diagram

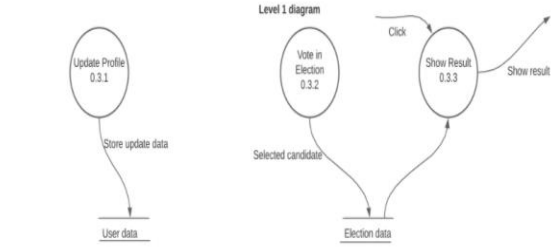
DFD Diagram:

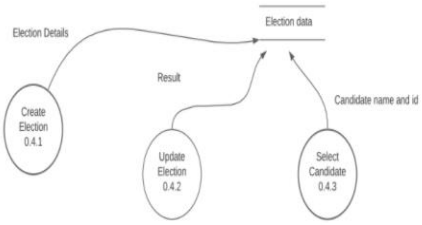


Level 0 / context Diagram



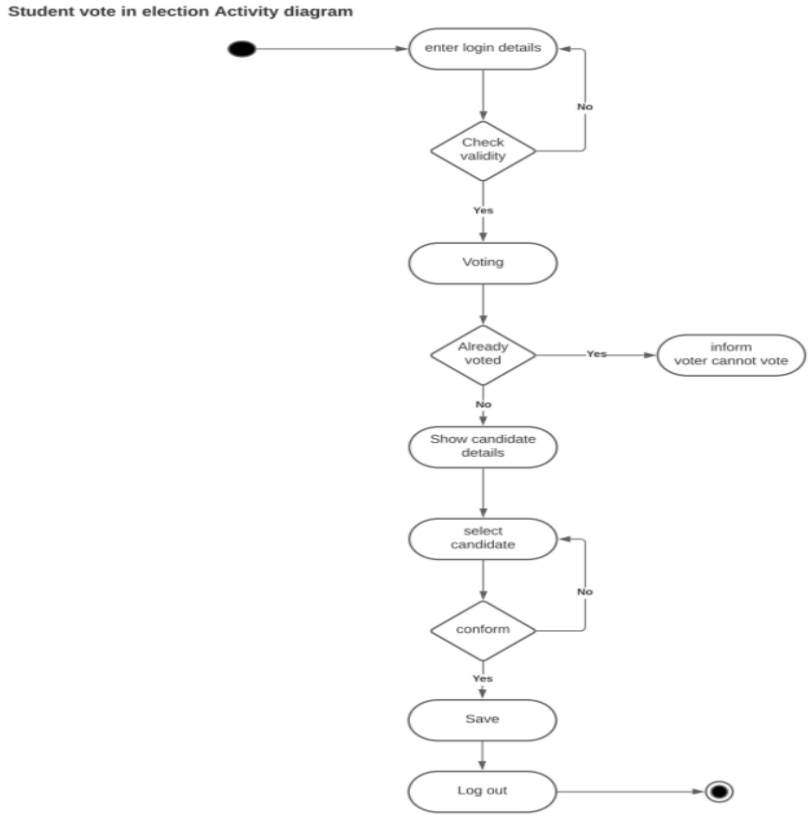
Level 1 diagram

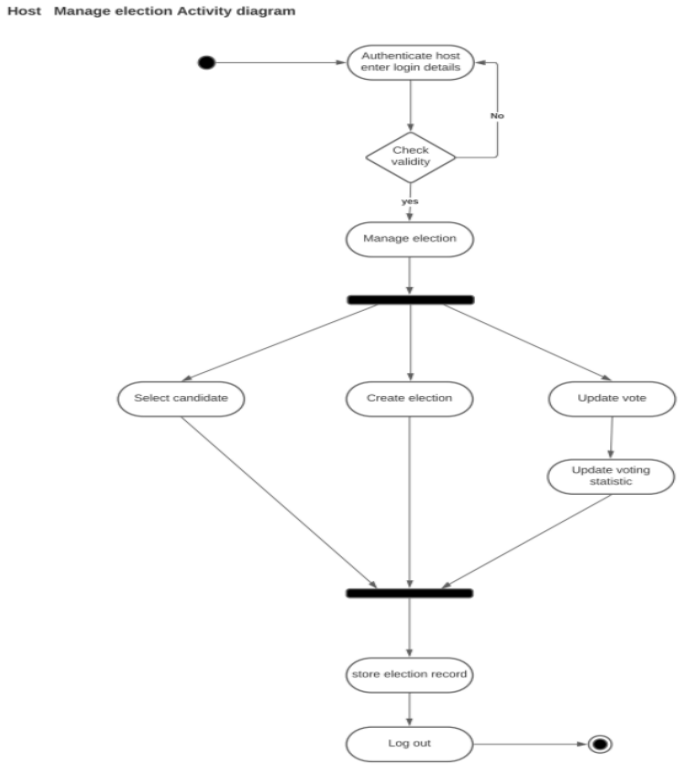




Level 2 diagram

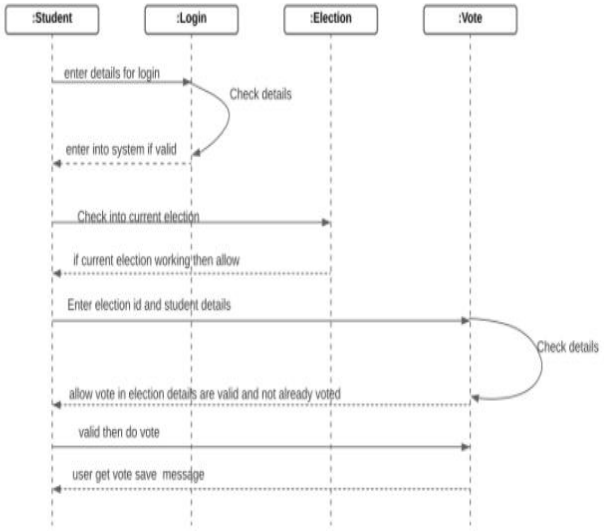
4.4) Activity Diagrams:



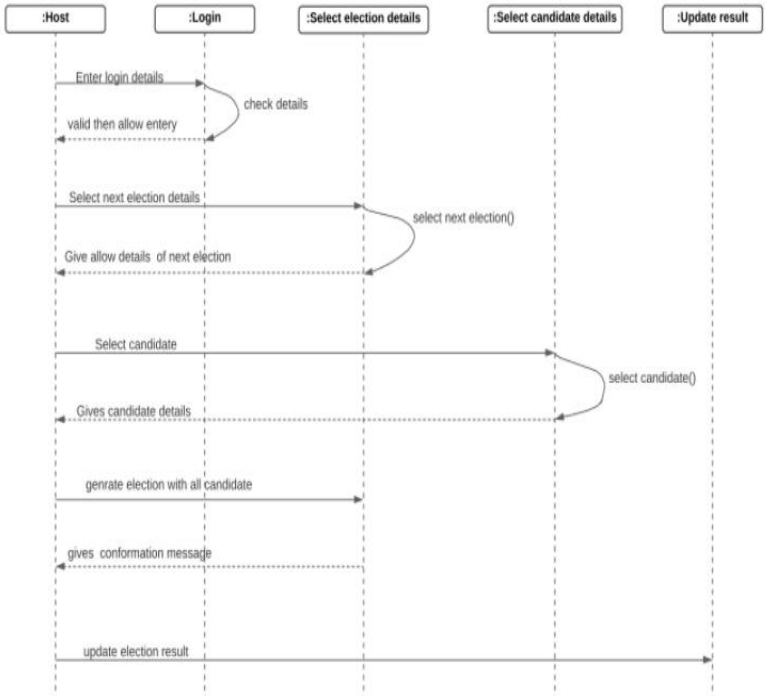


4.4) Sequence Diagram:

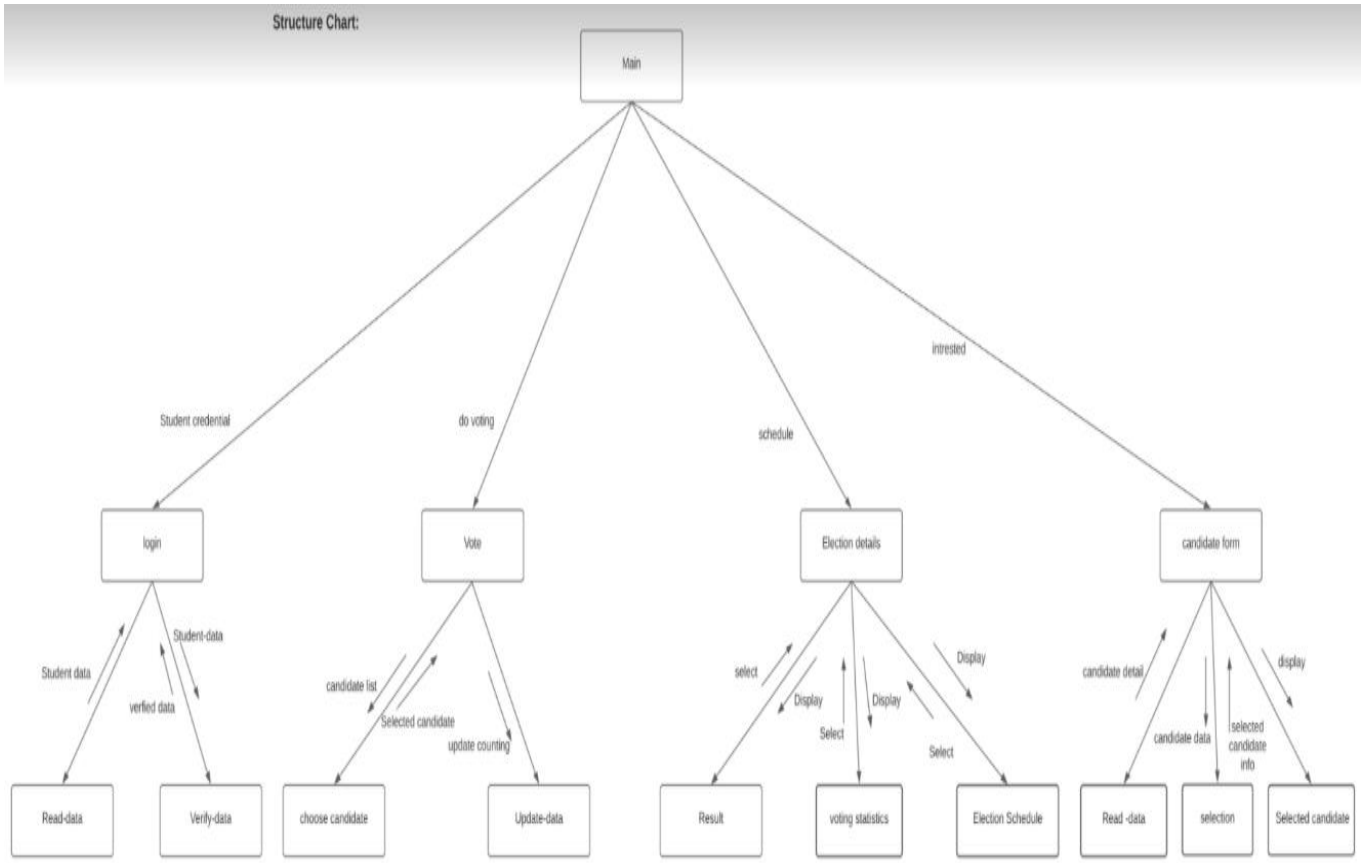
Vote in election sequence diagram



Create election squence diagram



4.5) Structurechart:



4.6) Models

```
class CreateElection(models.Model):
    name=models.CharField(max_length=30)
    type=models.CharField(max_length=30)
    date=models.DateTimeField()

    def __str__(self):
        return f'{self.name} {self.type}'
```

```python
class UserProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    voted = models.BooleanField(default=False)

    def __str__(self):
        return self.user.username
```

```python
class Candidate(models.Model):
    user=models.OneToOneField(User,on_delete=models.CASCADE)
    name = models.CharField(max_length=50)
    Description = models.TextField()
    votes = models.IntegerField(default=0)
    election_type=models.ForeignKey(CreateElection,on_delete=models.CASCADE)
    image = models.ImageField(default='default.jpg', upload_to='profile_pics')

    def __str__(self):
        return f'{self.user.username} Profile'

    def save(self, *args, **kwargs):
        super(Candidate, self).save(*args, **kwargs)

        img = Image.open(self.image.path)

        if img.height > 300 or img.width > 300:
            output_size = (300, 300)
            img.thumbnail(output_size)
            img.save(self.image.path)
# Create your models here.
```

5.) Implementation:

1)Modules

In This Software we mainly use four modules.
1)User Module
2)Election Module
3)Candidate Module

Each module consists of several methods to implement the required functionality. Implementation is done using Django. Database used in these module Sqlite3.

1)Register User Module

Basic information of user is taken by system and stored in database.

2) Login User Module

Users are able to login themselves. System logs user in, then and only then user can use other functionalities of system.

3) Election Module

this module is use for create election and it is main module of our application. mainly use for show result, candidate select, vote etc.

4) candidate Module: -

this module is use for create candidate for election. user which already register will apply for candidate.

2) FuctionPrototype

1.candidate form

```
def home(request):
    return render(request,'election/home.html')

@login_required
def candidate(request):
    if request.user.is_staff == True:
        return HttpResponseRedirect('/NotAllow/')
    if request.method == "POST":
        f=CandidateForm(request.POST)
        if Candidate.objects.filter(user=request.user).exists():
            return HttpResponseRedirect('/Present/')
        if f.is_valid():
            name=f.cleaned_data['name']
            description=f.cleaned_data['Description']
            image=f.cleaned_data['image']
            election_type=f.cleaned_data['election_type']
            can=Candidate(user=request.user,name=name,Description=description,image=image,election_type=election_type)
            can.save()
            return HttpResponseRedirect('/')
    else:
        f=CandidateForm()
    return render(request,'candidate/candidate.html',{'form':f})
```

2. vote view

```
@login_required
def vote(request):
    if request.user.is_staff == True:
        return HttpResponseRedirect('/NotAllow/')
    context = []
    pos = Candidate.objects.all()
    user = User.objects.get(username=request.user.username)
    profile = UserProfile.objects.get(user=user)
    if profile.voted == True:
        return HttpResponseRedirect('/Voted/')
    for c in pos:
       context.append([c,c.image.url])
    if request.method == 'POST':
        selected_candidate = Candidate.objects.get(name=request.POST['candidate'])
        selected_candidate.votes += 1
        selected_candidate.save()
        profile.voted = True
        profile.save()
        return HttpResponseRedirect('/')
    return render(request, 'election/vote.html', {'candidates':context})
```

3. election vote constraint ,selected candidate view

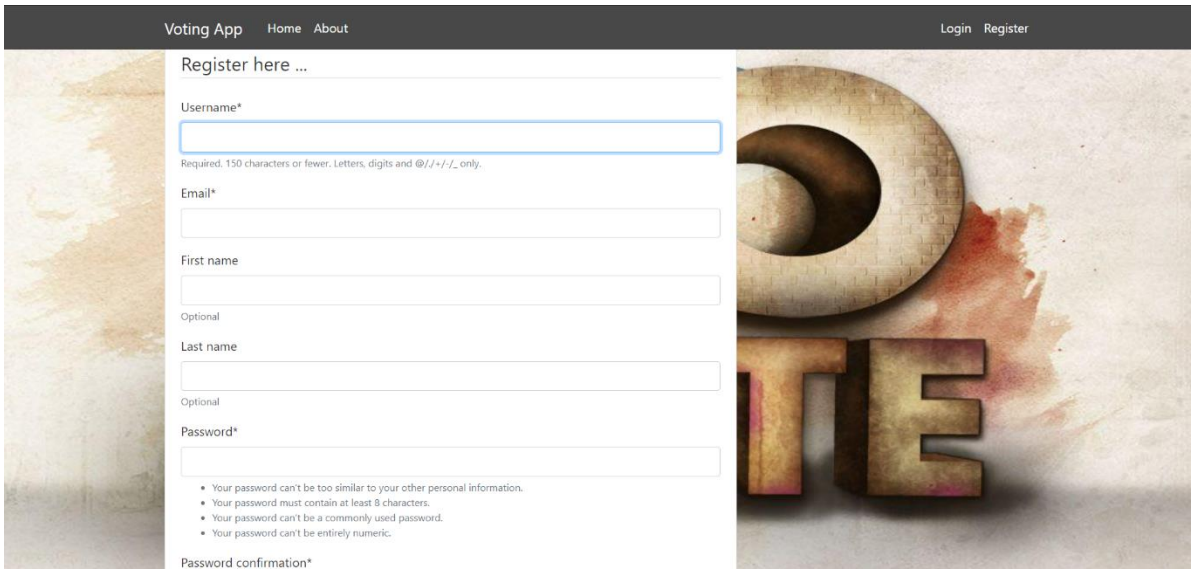
```
@login_required
def voted(request):
    return render(request,'election/voted.html')

@login_required
def notallow(request):
    return render(request,'election/notallow.html')

@login_required
def result(request):
    pos=Candidate.objects.all()
    context = []
    for c in pos:
        context.append([c,c.image.url])
    return render(request,'election/result.html',{'pos':context})

@login_required
def presentcandidate(request):
    return render(request,'election/present.html')
```

4.user register view

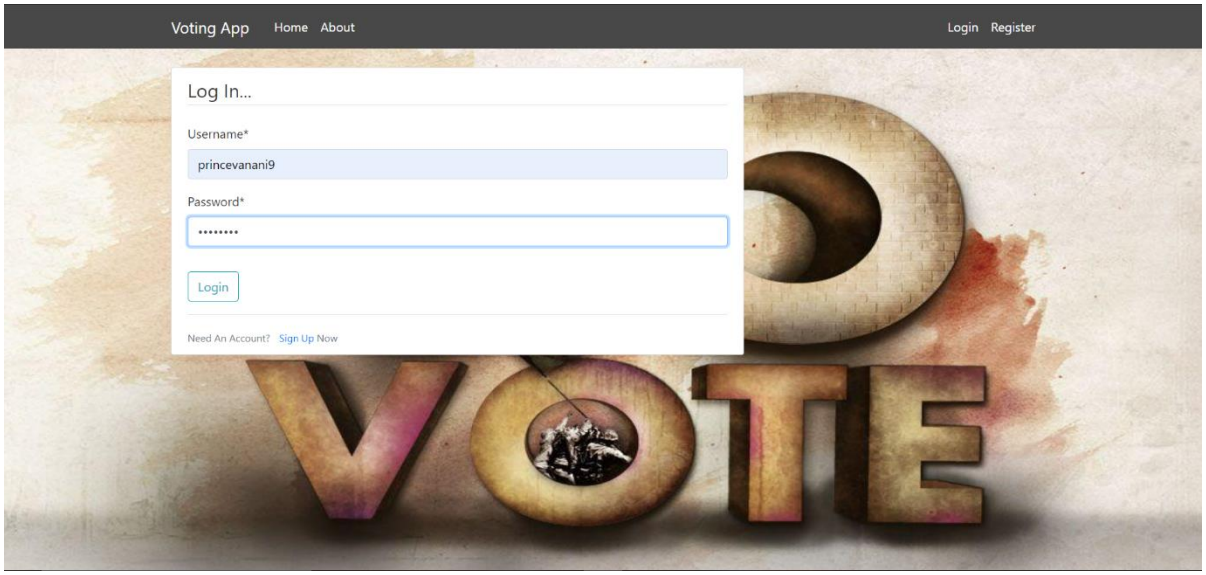
```

# Create your views here.
def register(request):
    if request.method == 'POST':
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get('username')
            messages.success(request, f'Your account has been created! You are now able to log in')
            return redirect('login')
    else:
        form = UserRegisterForm()
    return render(request, 'users/register.html', {'form': form})


```

6.) Screenshot:

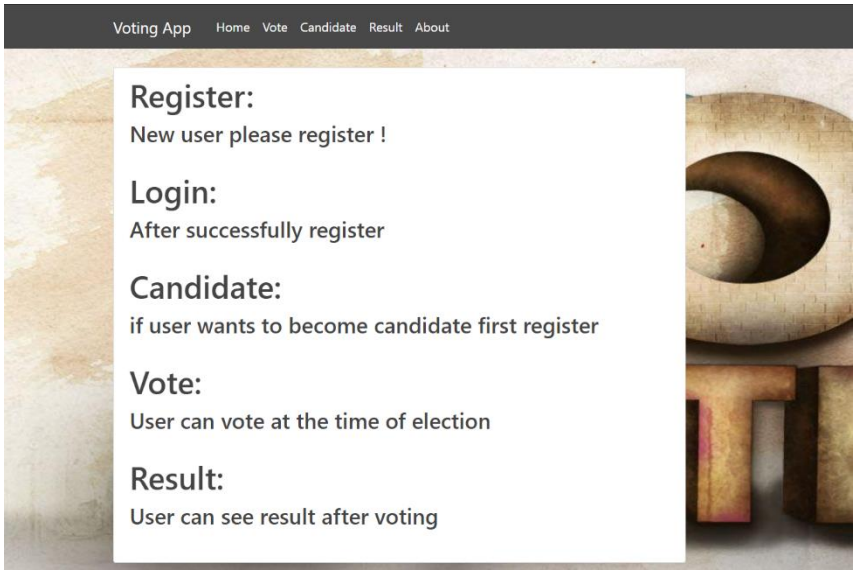
6.1) Register page



6.2) Login page

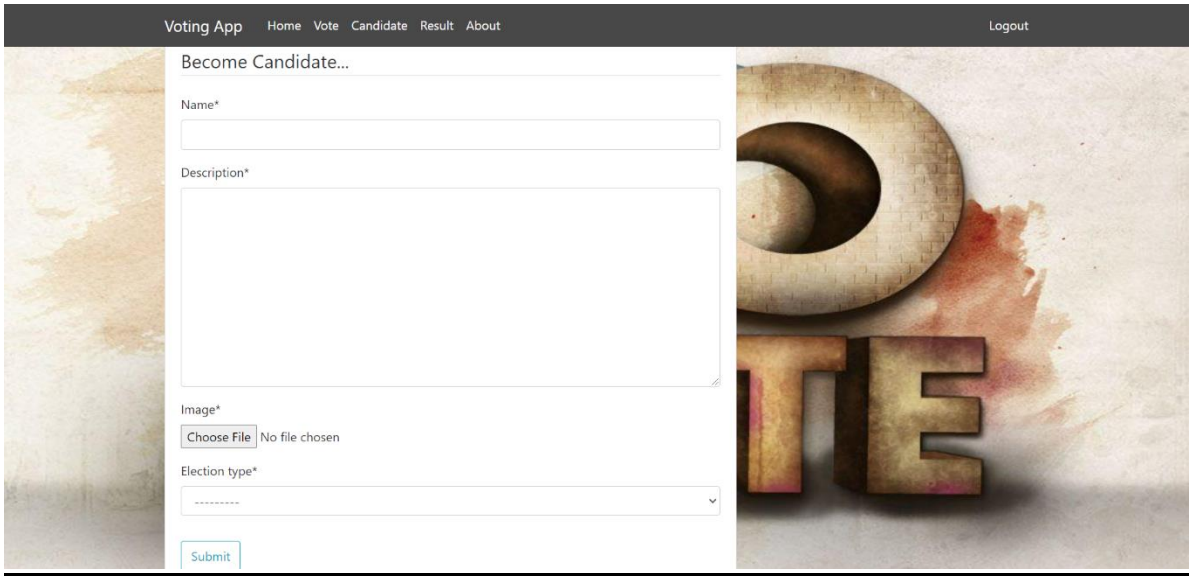


6.3)home page



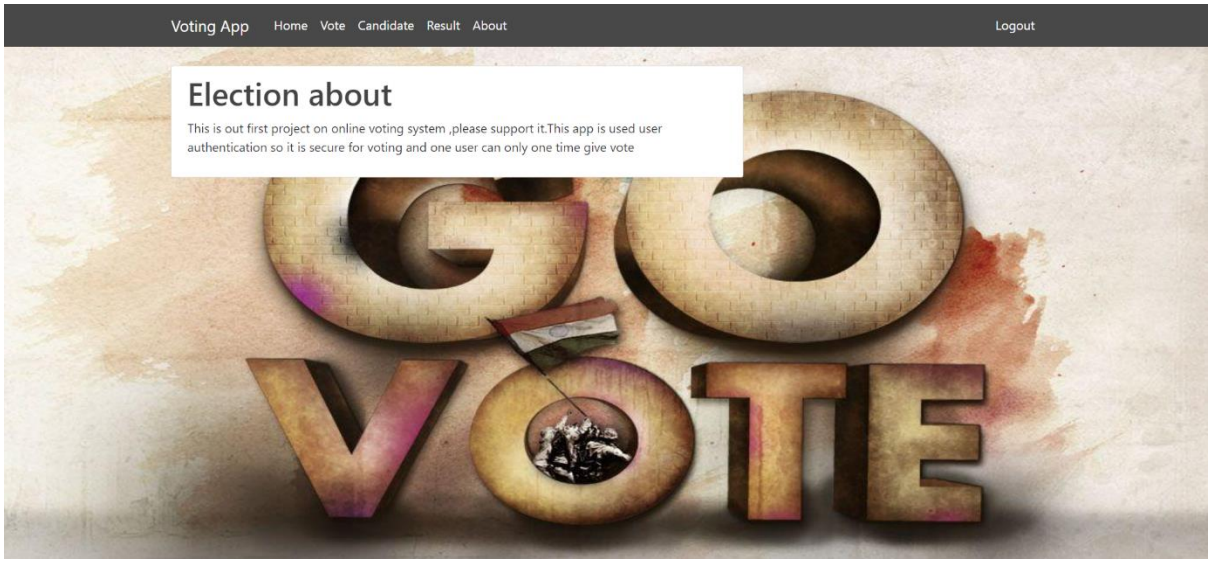
6.4) Vote page

6.5) candidate form



6.6) Result page

6.7)about page



6.8) Logout page

7.) Conclusion:

- This is a paperless work
-A online voting system can save time spent in voting on the booths.
-Online voting reduces paper work.
-It is non expensive and less time consuming, decrease time and cost.
-More secure than conventional voting system.
- It provides accurate information always.
-All gathered and extra information can be saved and can be accessed at any time

8.) Limitations:

-This system cannot work offline. it's must require internet.

-There is risk of cyber-attack from internal or external actors, which may manipulate the vote.

9.) Future extensions:

-We improve our project and overcome from this limitation, add some security and also add more modules finger print based voting system. Voting more functionality in pre-defined module.

10.) Bibliography:

Websites:

https://stackoverflow.com/

https://www.w3schools.com/

https://docs.djangoproject.com/en/3.1/