Digital Voting System

Design

1. **Structural Design**
   1. **Design**

System design is the process of defining elements of a system like segments, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing system which satisfies the specific needs and requirements of a project or business.

Design phase is measured as one of the most critical and vital stage of software development. The design phase uses all our gathered requirements and information which were created during the system analysis phase to transform them into a blueprint for the future system. The design phase controls how the new system will operate and determines the overall system architecture which includes a set of physical processing components, software, user, hardware, communication and the backend processing which if put together are hopeful to satisfy the future system requirements as proposed by the initial or planned requirement. For making the diagrams for this project called ‘digital voting system’ I must use software **StarUML** and for the prototype i must use **balsamiq Mockups 3** software**.**

**Some points that helps to achieve a good quality design.**

1. Making a required software and hardware selection.
2. Designing input and output flow.
3. Converting raw data into physical model.
4. Use simple content to make user friendly.
5. Involved user in every part of the design.

This project use above all the points to make system more efficient. The design phase also the includes how the system interacts with other sub-system, components and units. Therefore, this project follows the following tools for the design of the Digital Voting System.

**Tools used for this project:**

**XAMPP** for make a localhost in the computer include in MYSQL.

**Subline text-editor** in which write all the php, html code.

**Design library: - CSS, Font awesome, JS Query** and **bootstrap 4.3.1** for the design system interface.

**StarUML** software to make **class diagram**, **activity diagram** and **use case diagram**.

* 1. **Notation**

* 1. **Class diagram**

A class diagram is a type of diagram and part of a unified modeling language (UML) that defines and provides the overview and structure of a system in terms of classes, attributes and methods, and the relationships between different classes. Class diagram is a static diagram so, it represents the static view of an application.

I must use some notations for making class diagram these are given below table.

**Notations of class diagram:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. N** | **Notations** | **Name of Notation** | **Descriptions** |
| 1 |  | Class | Class is like a container in which store all the behavior attributes and essential operations of the system. |
| 2 |  | Association | Association can be represented by a line between the classes **with an arrow indicating the navigation direction**. In case arrow is on the both sides to communicate each other. |
| 3 |  | Aggregation | Aggregation implies a relationship where the child can exist independently of the parent. Example: Class (parent) and Student (child). Delete the Class and the Students still exist. |
| 4 |  | Composition | Composition is a special type of aggregation that denotes a strong ownership between Classes. Illustrate composition with a filled diamond. Example: House (parent) and Room (child). Rooms don't exist separate to a House. |
| 5 |  | Dependency | **Dependency** is often confused as Association. Dependency is normally created when you receive a reference to a class as part of a particular operation / method. Dependency is represented by a dashed arrow starting from the dependent class to its dependency |
| 6 |  | Interface | interfaces in class diagram to specify a contract between the interface and the classifier that realizes the interface. Each interface specifies a well-defined set of operations that have public visibility. |

Figure 1 notation of class diagram

**Class Diagram**

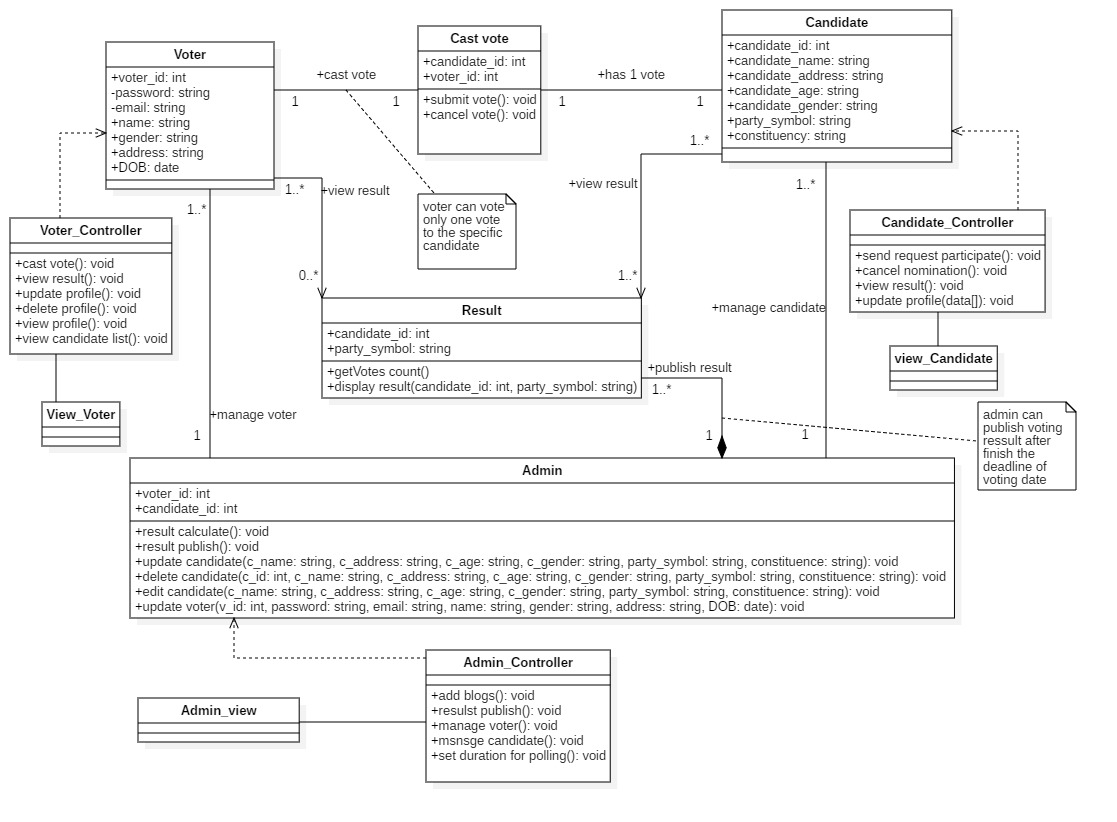


Figure 2 final class diagram of digital voting system

**Justification:**

The above class diagram illustrated that each class has each controller that is directly associated with database. Result is fully dependent in admin class so composition relation is established and admin class is control by admin controller. Result is directly viewed by voter and candidate so direct association relationship is established.

* 1. **Data Flow Diagram (DFD)**

Data flow diagrams (DFDs) shows the relationships among and between the various components in a program or system and data is processed within a system based on inputs and outputs. DFDs are a vital technique for modeling a system’s high-level detail by showing how input data is transformed to output results through a sequence of functional transformations.

This projects also use the notation for the making a DFDs diagram for the system called digital voting system.

DFDs consist four major components:

1. Entities
2. Processes
3. Database (stores data)
4. Data flows

DFDs notations are used to represent the flow of information, data sources and destinations, and where data is stored.

**Notations of Data Flow Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. N** | **Diagrams** | **Diagrams Name** | **Description** |
| 1 |  | Entity | An actor that contains the data that flows (input and output) between the entity and the system being diagrammed. |
| 2 |  | Data flow | This notation denotes the movement of data between external entities, process and data stores is represented with an arrow symbol which indicates the direction of flow. |
| 3 |  | process | In this notation an activity that changes or transforms data flows. Since they transform incoming data to outgoing data, all processes must have inputs and outputs on a data flow diagram. It handles the functions of the system. |
| 4 |  | Database | This notation only stores the data of the whole system it does not generate any operations but simply holds data for later access. |

Figure 3 notation of data flow diagram

**Data flow diagram:**

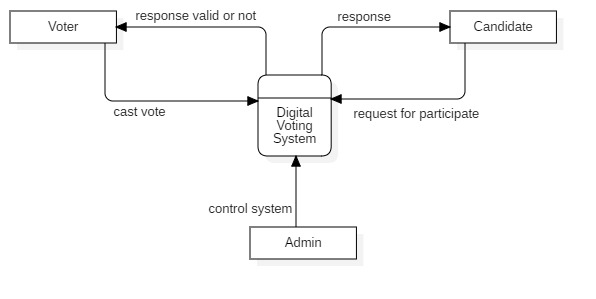


Figure 4 0-level data flow diagram

**Justification:**

The above diagram shows the 0-level data flow diagram. Some function are shows with the related directly to the system.

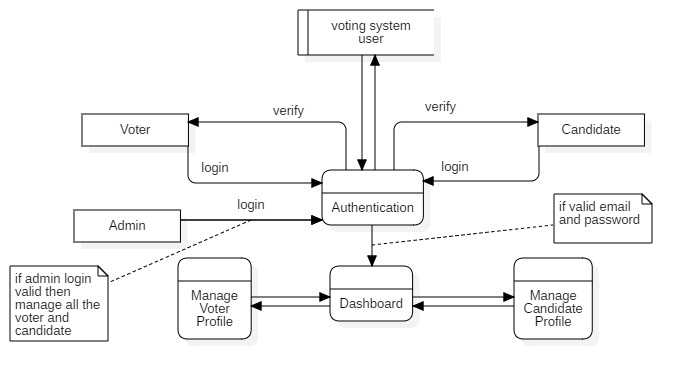


Figure 5 level-1 data flow diagram

**Justification:**

The above data flow diagram is a 1-level diagram of digital voting system. It shows the login system of voter and candidate. After login voter and candidate, they can access their dashboard in which they can manage their profile. Admin can also in that dashboard and his/her personal dashboard and manage all the voter and candidate and many other more functions.

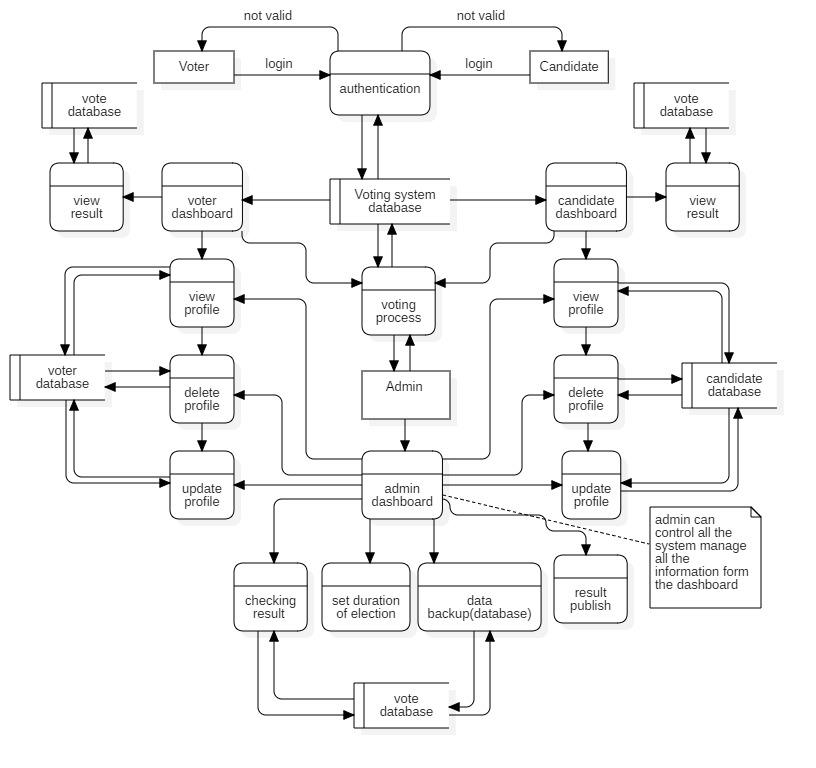


Figure 6 level-2 data flow diagram

**Justification**

In the above diagram I must create diagram as level 0, 1 and 2 data flow diagram. Activities that perform in this project like voters can vote and candidate send request to the admin for the participation of nominee. And admin can control the entire candidate and voters also update their profiles, add blogs, publish results of voting and results also viewed by voters and candidate directly in the system.

1. **Behavior Design**
   1. **Activity Diagram**

Activity diagram is an important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. Some notations of activity diagram are used for making diagram for this project. Notations are given below table.

**Activity diagram:**

1. Draw the activity flow of a system.
2. Describe the sequence from one activity to another.
3. Describe the parallel, branched and concurrent flow of the system.

**Notations of activity diagram:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. N** | **Notations** | **Notation Name** | **Descriptions** |
| **1** |  | **Initial node** | A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram |
| **2** |  | **Action** | An action state represents the non-interruptible action of objects. Actions and the operations of the system. |
| **3** |  | **Control flow** | Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line. |
| **4** |  | **Decision** | A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. |
| **5** |  | **Fork node** | A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram. |
| **6** |  | **Join node** | A join node joins multiple concurrent flows back into a single outgoing flow. |
| **7** |  | **Swimlane** | Swim-lanes group related activities into one column. |
| **8** |  | **Final node** | An arrow pointing to a filled circle nested inside another circle represents the final action state. |

Figure 7 notations of activity diagram

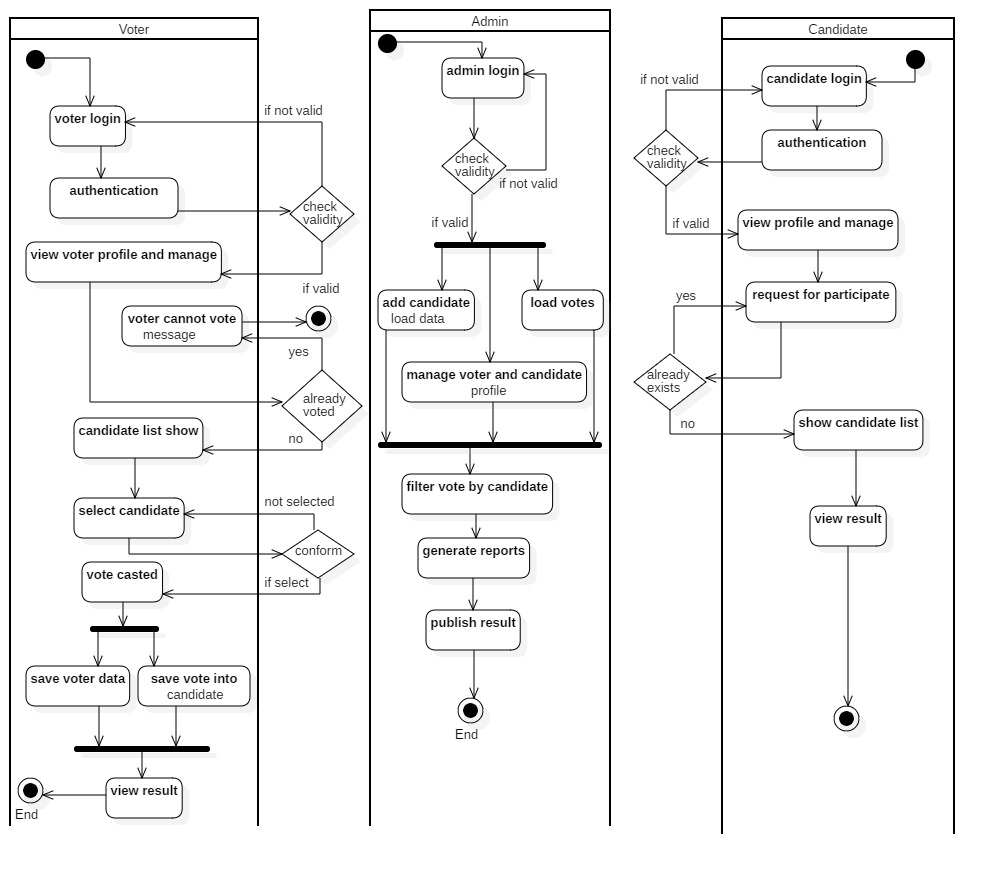


Figure 8 activity diagram of digital voting system

**Justification:**

In the first activity diagram shows the activity of the voters. Voters can at first login to the system and access their dashboard and manage all the information. They can cast vote in that form after generating voter id and then input some information about your self and select candidate and cast vote to the specific candidate.

In the second activity diagram shows the activity of the admin. Whole system controlled by admin after he/she login simply input default email and password. Admin can manage all the voters and candidate. He/she add blogs, publish result in the system.

In the third activity diagram shows the activity of the candidate. Candidate at first login to the system and access their dashboard in which they manage their all information and request for participation for the election. They can also view the result of voting.

* 1. **Sequence diagram**

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. Notations that are used in this project for sequence diagram are given below table.

**Sequences diagram benefits:**

1. Represents the details of a UML use case.
2. Model the logic of a sophisticated procedure, function, or operation.
3. See how objects and components interact with each other to complete a process.
4. Plan and understand the detailed functionality of an existing or future scenario.

**Notations of sequences diagram:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. N** | **Notations** | **Notation Name** | **Descriptions** |
| **1** |  | Lifeline | Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. |
| **2** |  | Combined fragment | This notation contains interactive elements of the diagram. Also known as a frame, this rectangular shape has a small inner rectangle for labeling the diagram. |
| **3** |  | Message initialize | This notation is the initialize point of the message of any system. |
| **4** |  | Synchronous message | This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply. |
| **5** |  | Self-message | A Self-Message reflects a new process or method invoked within the calling lifeline's operation. It is a specification of a Message with its self. |
| **6** |  | Reply message | Represent by a dashed line with a lined symbol arrowhead, these messages are replies to call. |

Figure 9 notations of sequence diagram

**Sequence diagram**

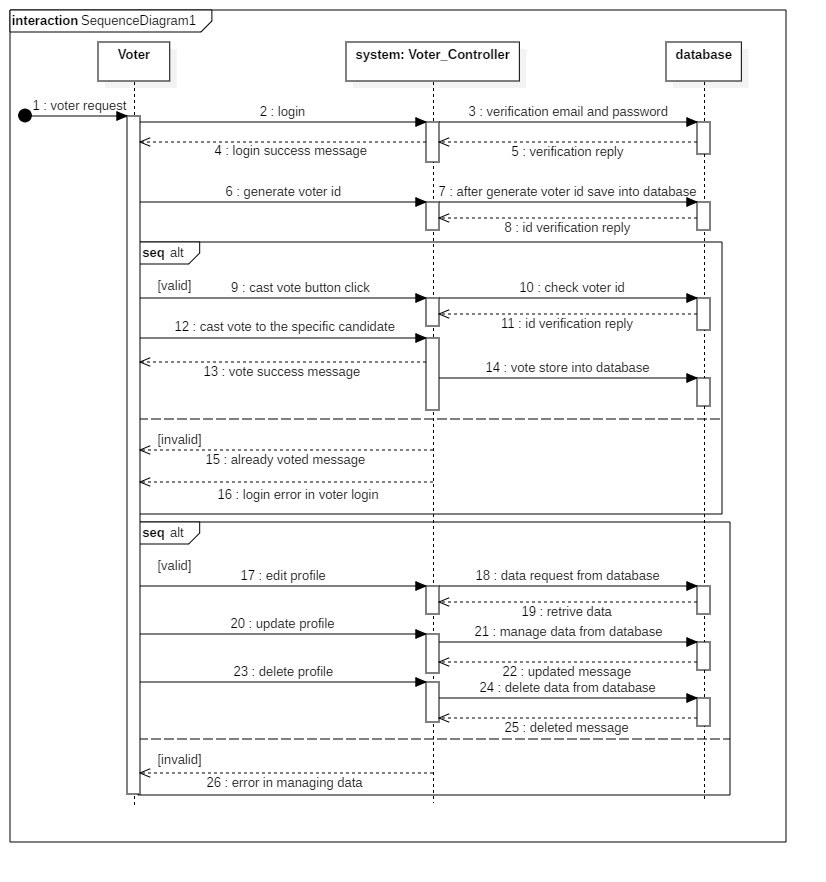


Figure 10 sequence diagram of voter activity

**Justification:**

In the above diagram shows the sequence diagram for voter activity in the system. Voter can send login request message with email and password to the system and system will send message to the database for authentication of email and password. If email and password is invalid then in combined fragment shows the error message. Another combined fragment shows all the activity perform by the voters. Voters can cast vote they can input voter id and then system send to the database for checking its authentication. And then database send a message of valid or invalid from combined fragment and then process of voting will be start.

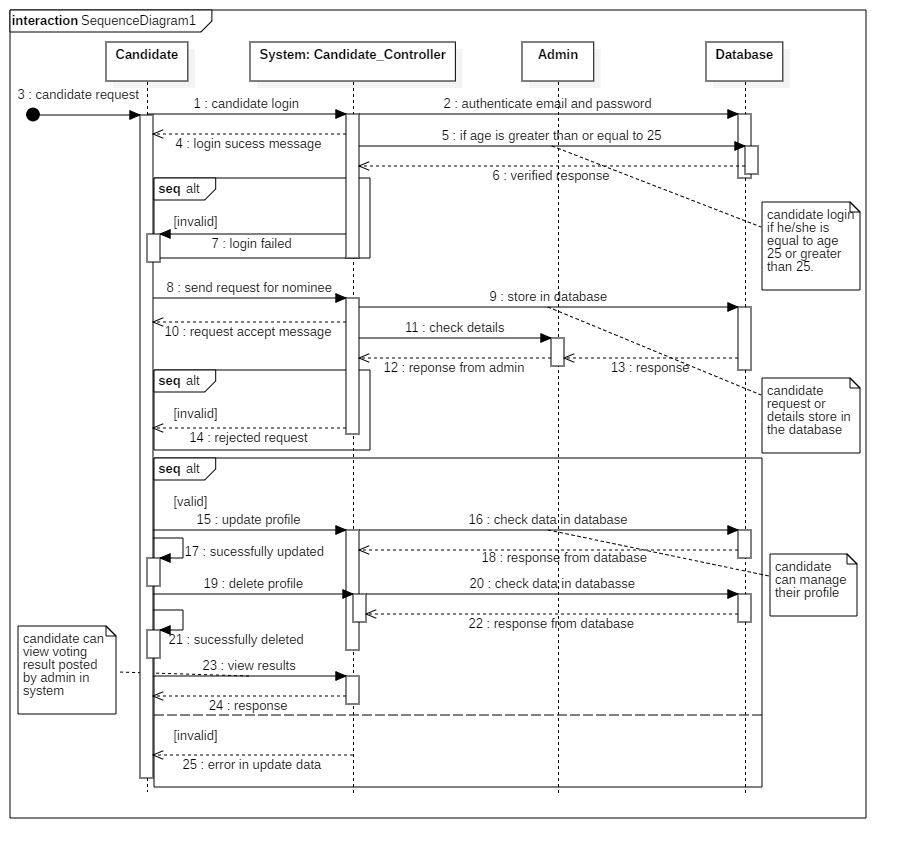


Figure 11 sequence diagram of Candidate activity

In the above diagram shows the sequence diagram for candidate activity in the system. It shows in diagram that candidate sends a login request message with email and password to the system and then system send it to the database for authentication of email and password if not valid error message shown in the combined fragment box. Candidate also send request for nominee to the system and system send it to the database for validation candidate and that data send to the admin and send a success message and rejected message shown in the combined fragment and many other operations perform in the combined fragment by the candidate.

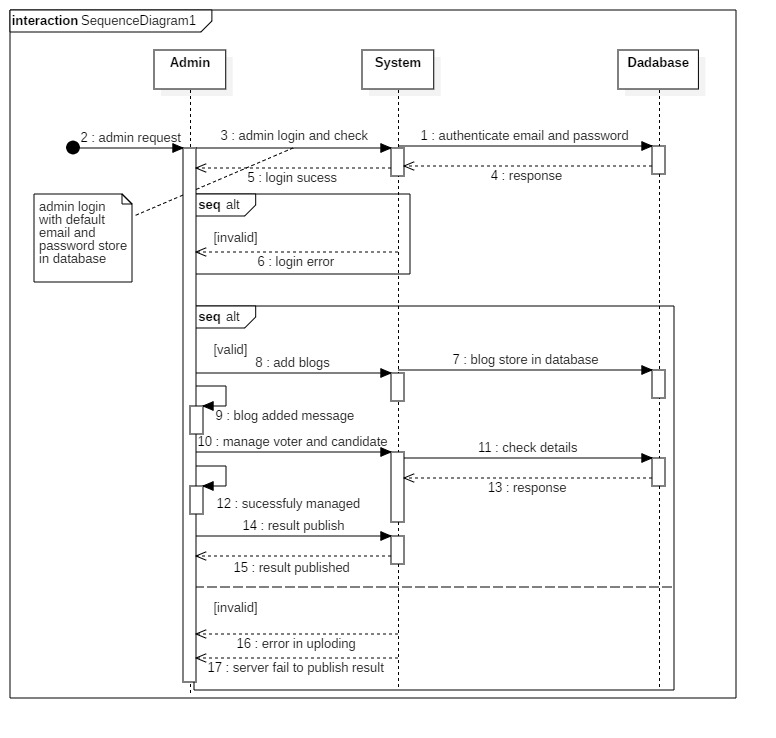


Figure 12 sequence diagram of admin activity

**Justification:**

In the above diagram shows the activity perform by admin in the sequence diagram shows. It shows in diagram that admin directly login into the system by the default email and password. Admin can control all the voter and candidate from their dashboard and add blogs, result publish and set time duration in the system shown in the combined fragment.

1. **Database design**

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database design involves classifying data and identifying interrelationships.

* 1. **Data Dictionary**

a set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.

This project (Digital Voting System) is used many tables:

1. Voter table
2. Candidate table
3. Admin table
4. Result table

**Data dictionary of voter table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Null** | **Constraint** | **Description** |
| Voter\_id | integer | 10 | No | Primary key | Primary key of table voter. |
| name | varchar | 100 | Yes | -- | Name of the voter. |
| gender | Varchar | 50 | Yes | -- | Gender of the voter. |
| address | Varchar | 100 | Yes | -- | Address of voter. |
| dob | Date | -- | Yes | -- | Date of birth of the voter. |
| email | Varchar | 100 | Yes | -- | Email of the voter. |
| password | varchar | 255 | Yes | -- | Password of the voter for login. |
| Admin\_id | integer | 10 | No | Foreign key | Foreign key comes from the admin table. |

**Data dictionary of candidate table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Null** | **Constraint** | **Description** |
| Candidate\_id | Integer | 10 | No | Primary key | Primary key of table voter. |
| Name | varchar | 100 | Yes | -- | Name of the candidate. |
| Phone\_No | Varchar | 50 | Yes | -- | Phone number of candidates. |
| Address | Varchar | 100 | Yes | -- | Address of candidate. |
| Dob | Date | -- | Yes | -- | Date of birth of the candidate. |
| Gender | Varchar | 50 | Yes | -- | Gender of the candidate. |
| Party symbol | Varchar | 255 | Yes | -- | Symbol of party in which candidate assign. |
| Email | Varchar | 100 | Yes | -- | Email of the candidate. |
| Password | Varchar | 255 | Yes | -- | Password of the candidate. |
| Admin\_id | Integer | 10 | No | Foreign key | Foreign key references from the admin table. |

**Data dictionary of admin table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **length** | **Null** | **Constraint** | **Description** |
| admin\_id | Integer | 10 | No | Primary key | Primary key of the admin table. |
| Email | Varchar | 100 | Yes | -- | Default email of the admin. |
| Password | Varchar | 255 | Yes | -- | Default password of the admin. |

**Data dictionary of result table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Null** | **Constraint** | **Description** |
| Voter\_id | Integer | 10 | No | Foreign key | Foreign key references from voter table. |
| Candidate\_id | Integer | 10 | No | Foreign key | Foreign key references from candidate table. |
| Admin\_id | Integer | 10 | No | Foreign key | Foreign key references from admin table. |

* 1. **ER-Diagram:**

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

The elements of ER-Diagram:

1. Entities
2. Relationships
3. Attributes

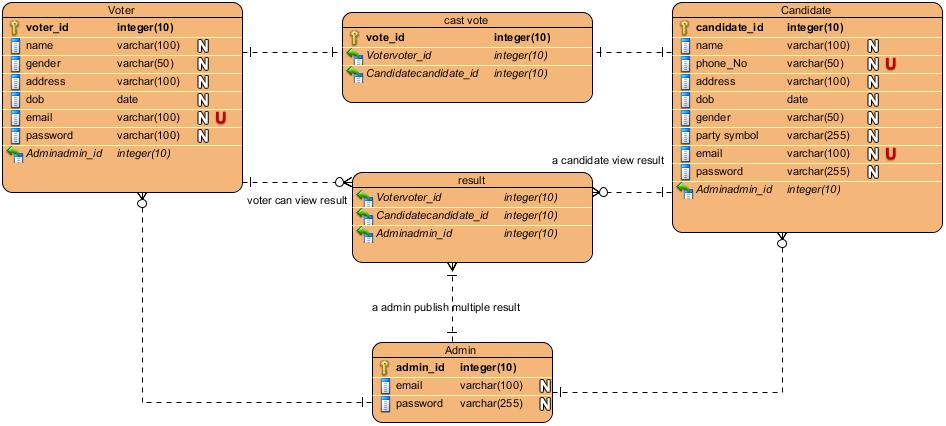


Figure 13 Entity Relationship-Diagram

**Justification:**

In above diagram, it is ER-diagram for digital voting system. It shows the relationship between entities of database. Above ER-diagram is developed for real working database for digital voting system. In the above diagram, one voter can cast maximum one vote only and candidate also. Admin can publish one or more result about voting system. Admin and voter have one to many relationships, voter can control by an admin and also admin and candidate has one to many relationships. Admin can control the candidate also. Admin can manage one or many voter and candidate.

1. **Three Tier System Architecture**

System architecture is the structural design of systems. Systems are a class of software that provide foundational services and automation. It is the conceptual model that describe the structure, behavior and more views of a system.

The system architecture can consist of system components and the sub systems developed, that will work together to implement the overall system.

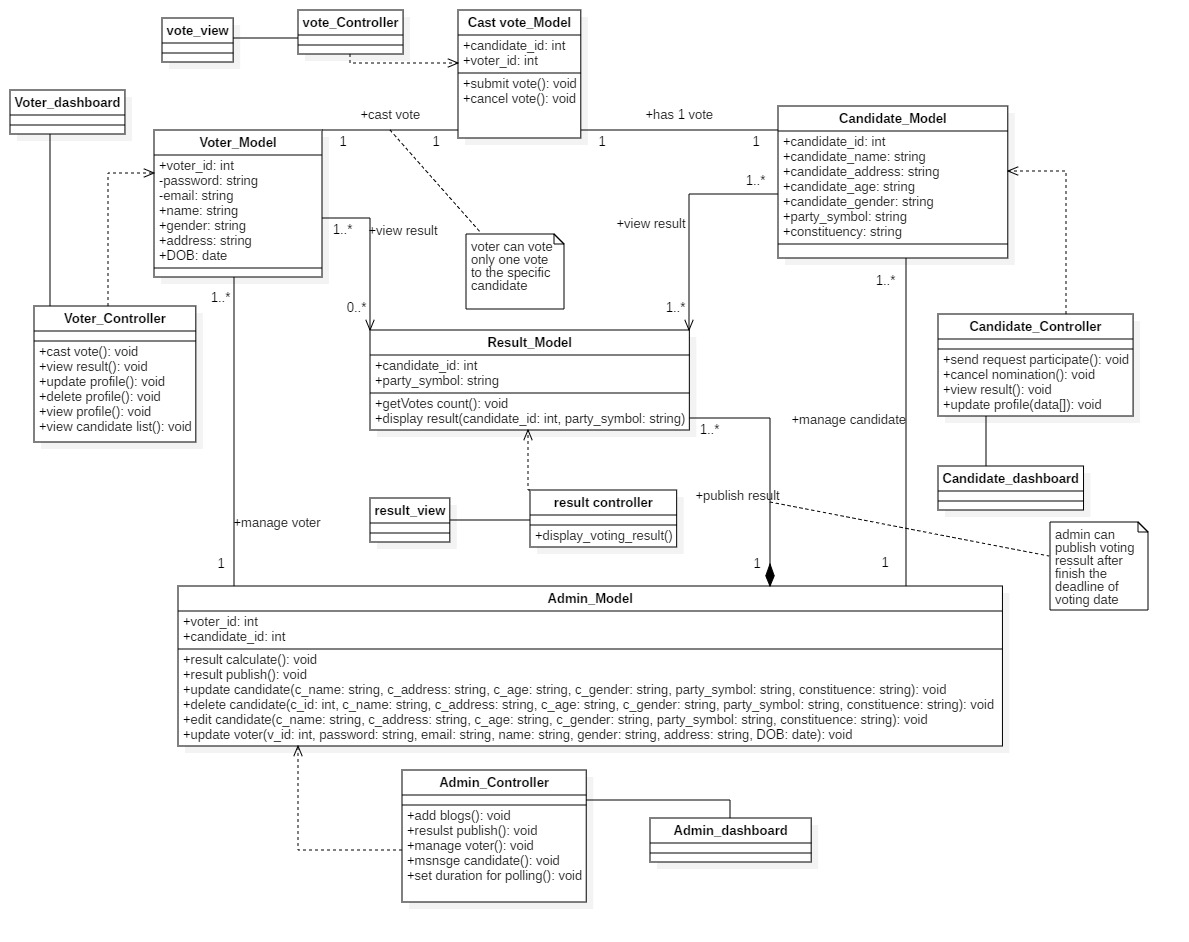


Figure 14 system architecture

**Justification:**

In the above diagram shows the final system architecture. This project uses the three-tier website application layer. Above system architecture use MVC (model view controller) design patter. It shows the relationship between the different classes. Admin class which was controlled all the system. Control all the voters and candidate with their dashboard and publish voting results with the strong relationship. This architecture meet all the three layer tier so this is the well managed web application.

1. **UI (user interface) design:**

User interface design or UI design generally refers to the visual layout of the elements that a user might interact with in a website, or technological product. This could be the control buttons of a radio, or the visual layout of a webpage. User interface designs must not only be attractive to potential users, but must also be functional and created with users in mind.

* 1. **Prototype:**

A prototype is the initial example of a product or program, which acts as a basis for following designs. Prototypes are often made from inexpensive materials and are usually made at a lower level of detail than a final product. In a system design, prototypes can be made that focus on the visual design or the functionality of a program.

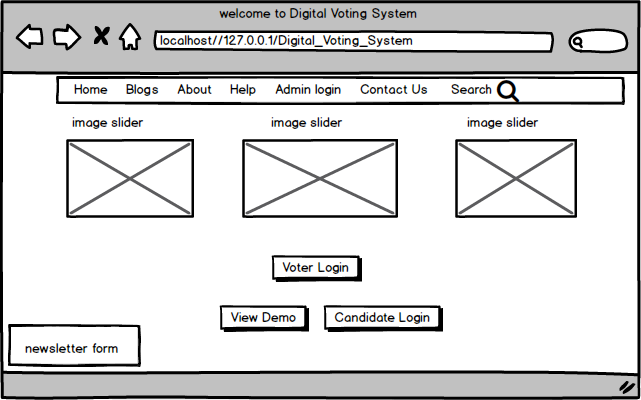


Figure 15 Home page

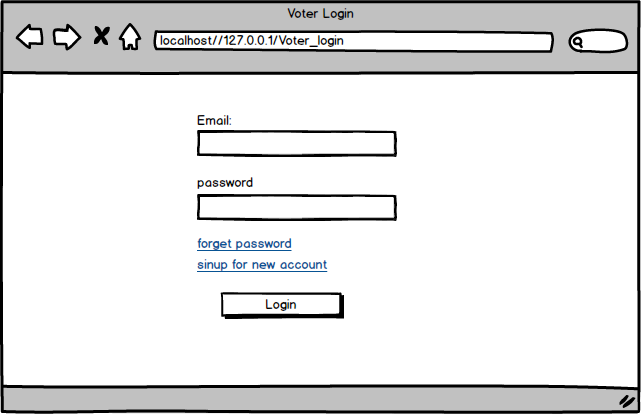


Figure 16 voter login

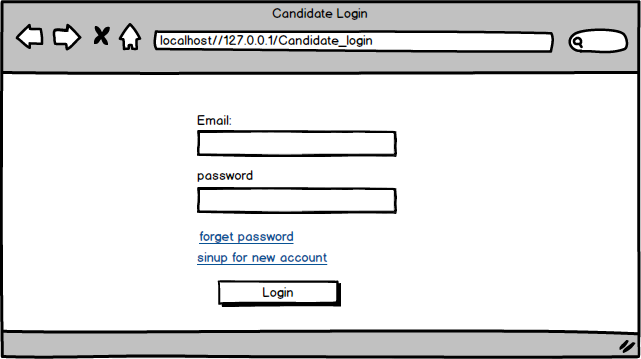


Figure 17 candidate login

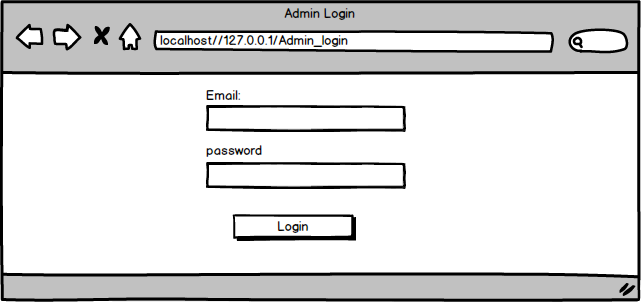


Figure 18 admin login

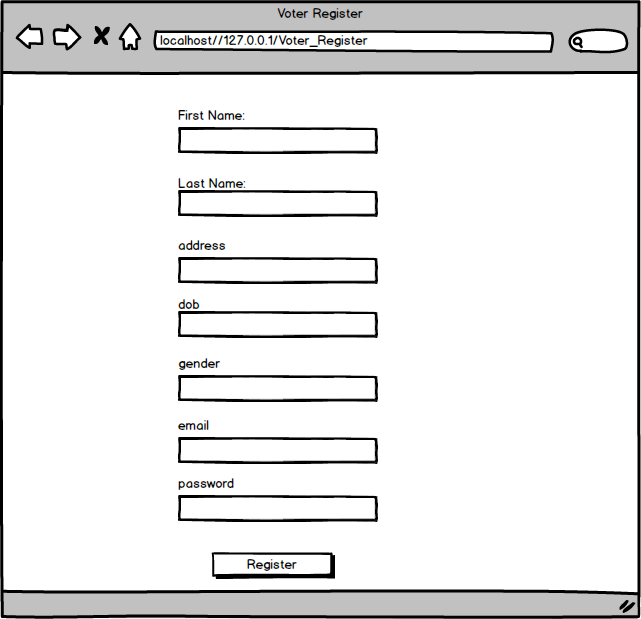


Figure 19 voter register

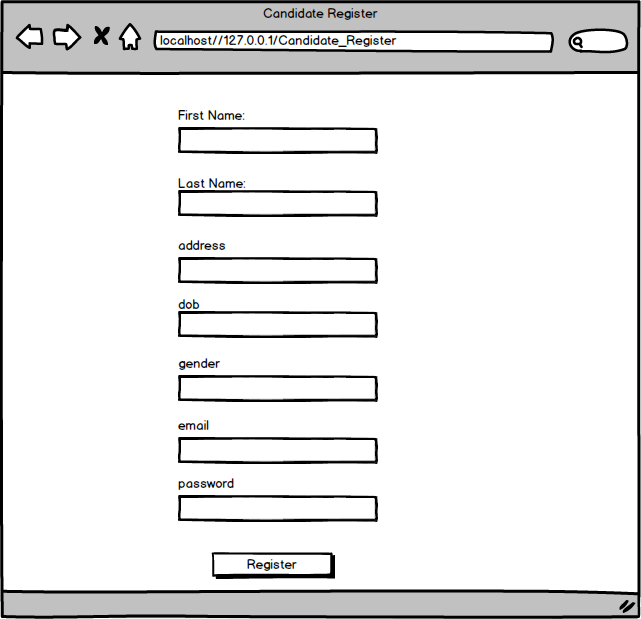


Figure 20 candidate register

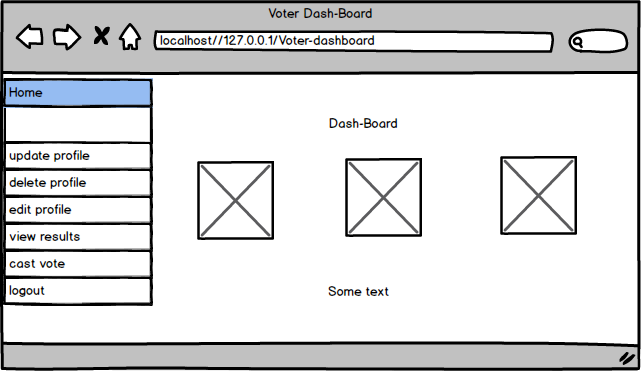


Figure 21 voter dashboard

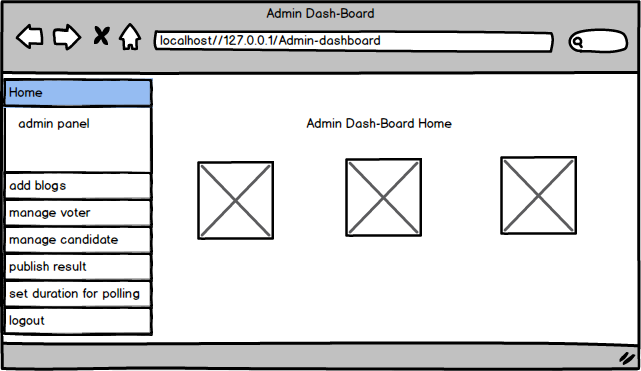


Figure 22 admin dashboard

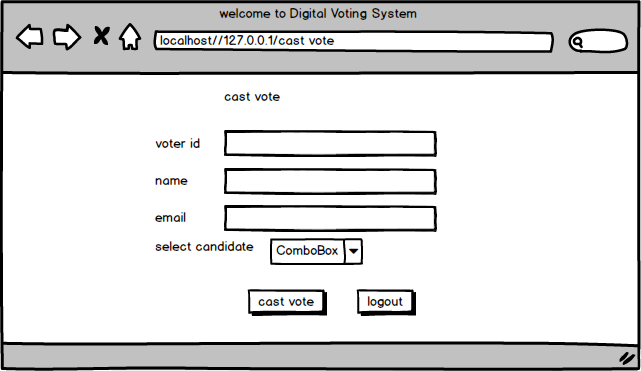


Figure 23 cast vote by voters