**CHAPTER THREE**

**Methodology and Design**

**3.1 Introduction**

A methodology is a process of rigorous study or inquiry, particularly to unearth new facts or information; hence, research methodology should be good enough to enable the achievement of the specified objectives which are achievable using specific components, such as data collection and design procedures and system modeling (use case, activity, and class diagrams). This chapter contains the input/output specifications, and system requirements for the development of a web-based voting application for the computer science department.

**3.2 Methods of Data Collection**

Before developing any system, collecting data and facts about the existing system is critical to understand what is going on. This research was carried out using three methods.

1. Observation of the Work Environment
2. Interview
3. Documentation

**3.2.1 Observation of the Work Environment**

This method was employed to acquire information and data for this study by monitoring how the manual system worked. The most evident flaws in the existing system were discovered via detailed inspection. Using the observational approach, the context in which the observation is made can be modified in a variety of ways.

**3.2.2 Interview**

The main objective of using interviews as a method of data collection is to obtain information thoroughly and rigorously. Based on the questions the researcher provided, the researcher met meet with some election commission members and acquired reliable information.

**3.2.3 Documentation**

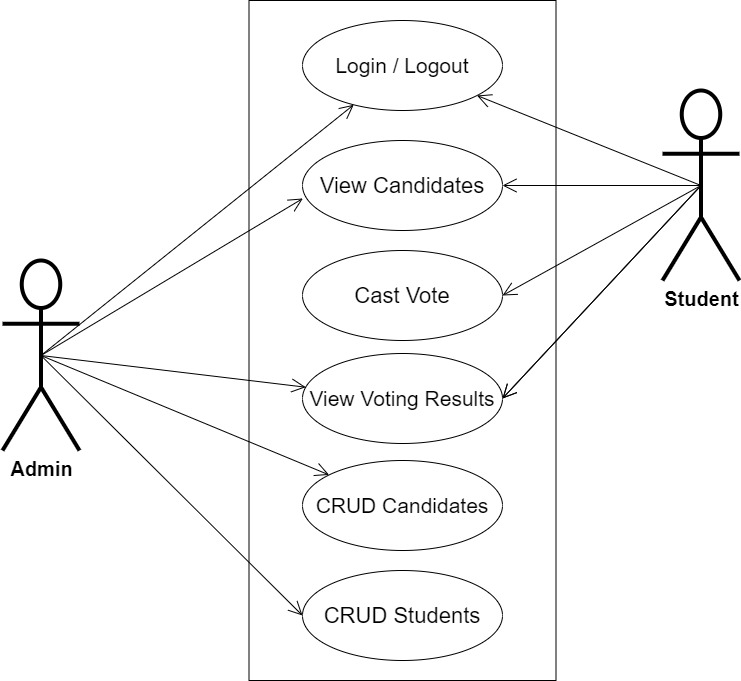
Documentation is a type of secondary data collection. This method makes use of journals, manuals, past work, publications, and other sources. This method of data collection is used because it allows for comparison with past studies. This includes the internet, which is a data collection tool. The internet was used to find information on difficult or ambiguous issues.

**3.3 System Modeling**

A system model is a conceptual model of a system that explains and represents it. A system is any interaction between a set of components that work together to achieve a common purpose. Visual models of object-oriented software-intensive systems may be created utilizing a set of visual notation techniques included in the Unified Modeling Language, which is used in the creation of this contemporary system. UML diagrams utilized in this new design include use case diagrams, class diagrams, and activity diagrams.

**3.3.1 Use Case Diagrams**

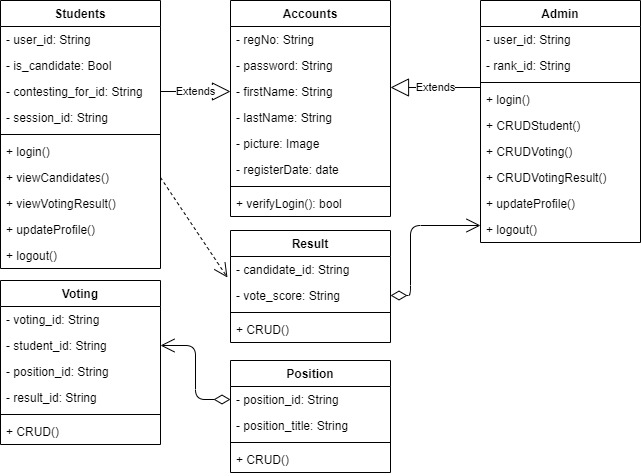
Use cases are collections of system-to-user interactions. Use case diagrams are used to graphically characterize the functionality of a system in terms of its actors, goals (represented as use cases), and dependencies between those use cases.



**Fig 3.1 System Use Case Diagram**

**3.3.2 Class Diagrams**

The Unified Modeling Language (UML) class diagram is an implementation of an independent perspective of how the system interface would look, with each class having its own set of attributes and demonstrating how they interact with one another. Class diagrams employ the Unified Modeling Language standards to visually portray a given system's static structure and composition (UML).



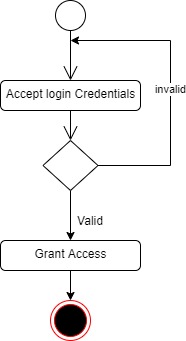
**Fig 3.2 System Class Diagram**

**3.3.3 Activity Diagrams**

An activity diagram, like a flowchart or a data flow diagram, visually illustrates a series of events or the flow of control in a system, but it acts more like an enhanced version of both.

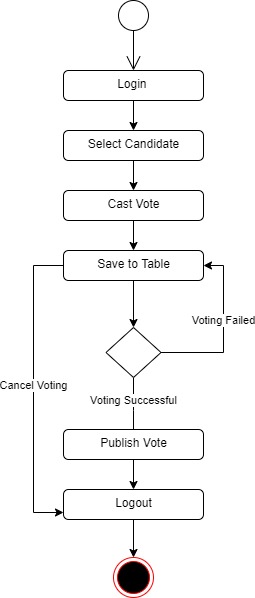
**Login**

The process for gaining access to the system is depicted in the diagram below; the username and password must be accurate to gain access.



**Fig 3.3.1 Login Activity Diagram**

**Voting**

The process for voting for a candidate is depicted below, to vote one has to select from each category of available positions.

**Fig 3.3.2 Voting Activity Diagram**

**3.4 Database Design**

The logical explanation of how data is kept in the computer's memory is called input specification. The freedom experienced in using the system, as well as the convenience of retrieving and reading the data and assuring applicability across the internet, make SQL standards essential for ensuring that structured data is uniform and independent of applications. Some of the input specifications employed in this project work are presented below.

1. Accounts Table: contains basic information about all system users.
2. Voting Table: contains the voting record for the system.

**Table 3.1 Account Table input specification table**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Reg\_no | String | 150 | Reg\_no for login (case insensitive) |
| Password | String | 150 | Access Code (case sensitive) |
| Firstname | String | 150 | User first name |
| Lastname | String | 150 | User last name |
| Picture | Image | - | User profile picture |
| RegisterDate | String | 100 | The date the user was registered |
| acct\_id | String | 64 | A unique string for identifying users |

**Primary key:** acct\_id

**Table 3.2 Voting Table input specification table**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Student\_id | String | 150 | Reference to the student model |
| Position\_id | String | 150 | Reference to the position model |
| Result\_id | String | 150 | Reference to the result of the voting |
| Voting\_id | String | 64 | A unique string for identifying votes |

**Primary key:** voting\_id

**3.5 Output Design**

This declares and displays the outcome of the given input. The automated system's output is dependent on its input. The output specification is listed below.

**Table 3.3 Account Table Output Design**

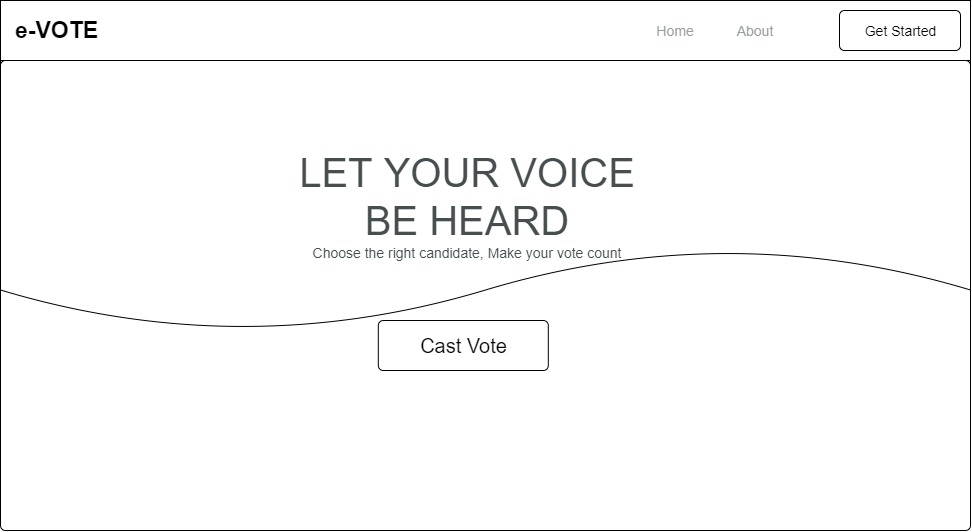
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Reg\_no** | **Password** | **Firstname** | **Lastname** | **RegisterDate** | **Picture** | **Acct\_id** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**Table 3.4 Voting Table Output Design**

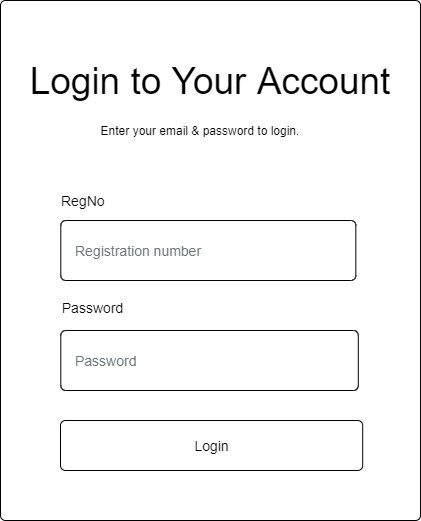
|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_id** | **Postion\_id** | **Result\_id** | **Voting\_id** |
| XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX |

**3.6 Input & User Interface Design**

This is a graphic depiction of the system interface; it will be designed to be user-friendly, responsive, and visually beautiful. Furthermore, it will be fully secured, thus authentication will be required to see various levels of the information. To help with the designs, a mid-fidelity wireframing program called Draw.io is employed.



**Fig 3.6.1 Home Page**



**Fig 3.6.1 User Login Screen**



**Fig 3.6.2 Contestant Form**

**3.7 System Requirement**

Every piece of software-generated has predefined system requirements that it must fulfill in order to function properly. The system requirements, on the other hand, are the bare minimum of hardware and software required for the system's intended operation.

**3.7.1 Hardware Requirement**

System Hardware Requirement Include:

1. Minimum of 2 GB of RAM (Random Access Memory).
2. Minimum of Intel Dual core processor.
3. Minimum of 250GB HDD (Hard Disk Drive).

**3.7.2 Software Requirement**

The software requirements include:

1. At least windows 7 OS (Operating System).
2. Vs. Code IDE installation.
3. Browsers include Chrome and Firefox.

**3.8 Choice of Programming Language**

This research work will be a web-based application and will be implemented on a relational database system (SQLite). HTML (hypertext markup language), CSS (cascading style sheet), and JavaScript will be employed in the front end while Django(python) will be employed for the backend programming. The above are the modern languages used in implementing this system.