**CHAPTER THREE**

**Methodology and Design**

**3.1 Introduction**

A methodology is a rigorous study or inquiry, particularly to unearth new facts or information; thus, 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 provides the input/output specifications as well as the system requirements for developing an announcement system.

**3.2 Methods of Data Collection**

Before constructing any system, it is necessary to collect data and facts about the existing system to comprehend what is going on. Two approaches were used in this study.

1. Observation of the Work Environment
2. Documentation

**3.2.1 Observation of the Work Environment**

By monitoring how the manual system worked, this method was employed to acquire information and data for this study. A careful review highlighted the most evident flaws in the current system. When utilizing the observational approach, the environment in which the observation is taken can be changed in a variety of ways.

**3.2.2 Documentation**

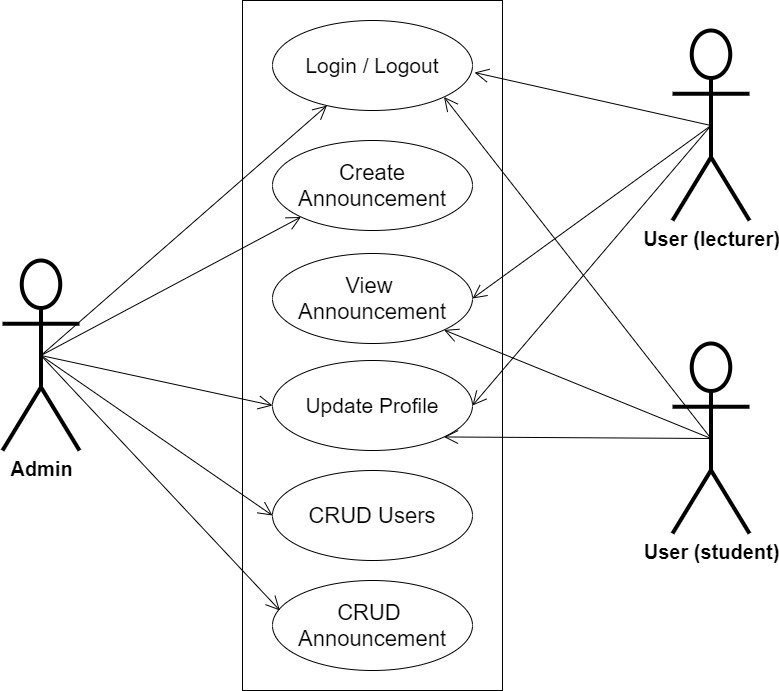
Documentation is part of secondary data collection. In this case, journals, manuals, past work, publications, and other sources are utilized. This method of data collection was chosen because it allows for comparison with past studies. This includes the internet, which is a data collection tool. The internet was used to study difficult or ambiguous situations.

**3.3 System Modeling**

A system model is a conceptual model of a system that explains and depicts it. A system is any interaction between a group of components that work together to achieve a common purpose. A collection of visual notation techniques inherent in the Unified Modeling Language, which was used to design this current system, may be used to construct visual models of object-oriented software-intensive systems. UML diagrams utilized in this new design include use case diagrams, class diagrams, and activity diagrams.

**3.3.1 Use Case Diagrams**

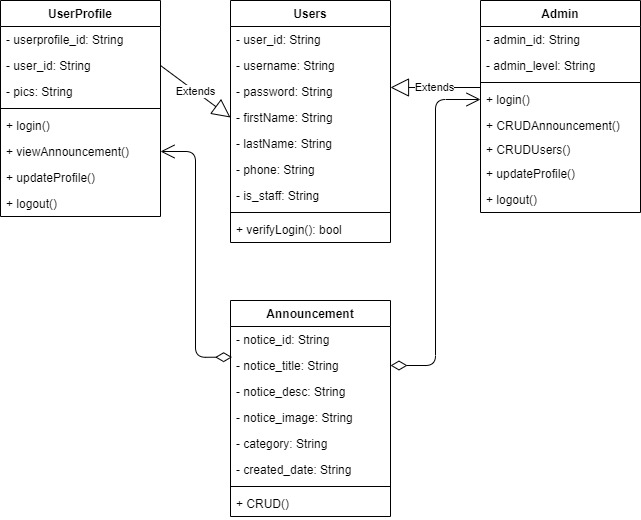
Use cases are groups of interactions between a system and a user. Use case diagrams are used to graphically portray the functioning of a system in terms of its actors, goals (expressed as use cases), and dependencies between those use cases.



**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 view of how the system interface might appear, with each class having its own set of properties and displaying how they interact with one another. Class diagrams use the Unified Modeling Language standards to visually depict the static structure and composition of a given system (UML).



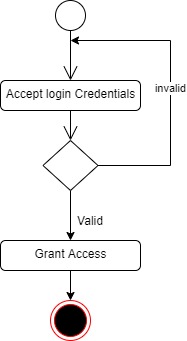
**Fig 3.2 System Class Diagram**

**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.

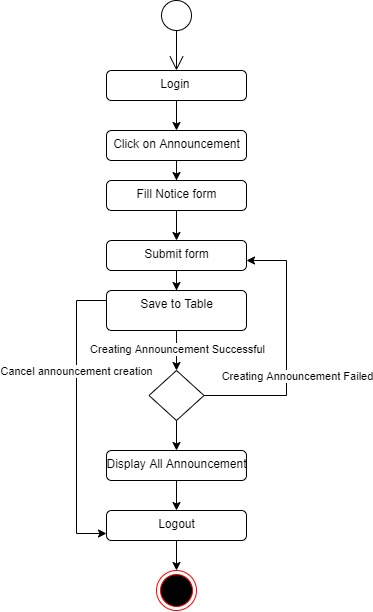
**Login**

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 Login Activity Diagram**

**Creating Announcement**

The process for creating a notice for other users to view is depicted below, to create a notice one has to be authenticated and must have proper authorization.

**Fig 3.4 Creating Announcement Activity Diagram**

**3.4 Database Design**

Input specification is the logical explanation of how data is stored in the computer's memory. SQL standards are vital for guaranteeing that structured data is uniform and independent of applications due to the flexibility experienced when using the system, as well as the simplicity of accessing and reading the data and ensuring applicability throughout the internet. The following are some of the input specifications used in this project effort.

1. Users Table: contains basic information about all system users.
2. Announcement Table: contains every system-saved announcement information.

**Table 3.1 Users Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| user\_id | Varchar | No | PK | 32 | Unique string for identifying users |
| username | Varchar | No |  | 100 | Username |
| password | Varchar | No |  | 128 | User Password |
| firstName | Varchar | No |  | 60 | User first name |
| lastName | Varchar | No |  | 60 | User last name |
| phone | Varchar | No |  | 20 | User type (student/others) |
| is\_staff | Bool | No |  | 100 | User profile picture |

**Table 3.2 Announcement Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| notice\_id | Varchar | No | PK | 32 | Unique string identifying notice |
| notice\_title | Varchar | No |  | 60 | Title of the notice |
| notice\_desc | Varchar | No |  | 100 | Notice Description |
| notice\_image | Varchar | No |  | 100 | Notice image |
| category | Double | No |  | 128 | Category of the notice |
| created\_date | Date | No |  | 20 | Creation date |

**3.5 Output Design**

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

**Table 3.3 Users** **output design table**

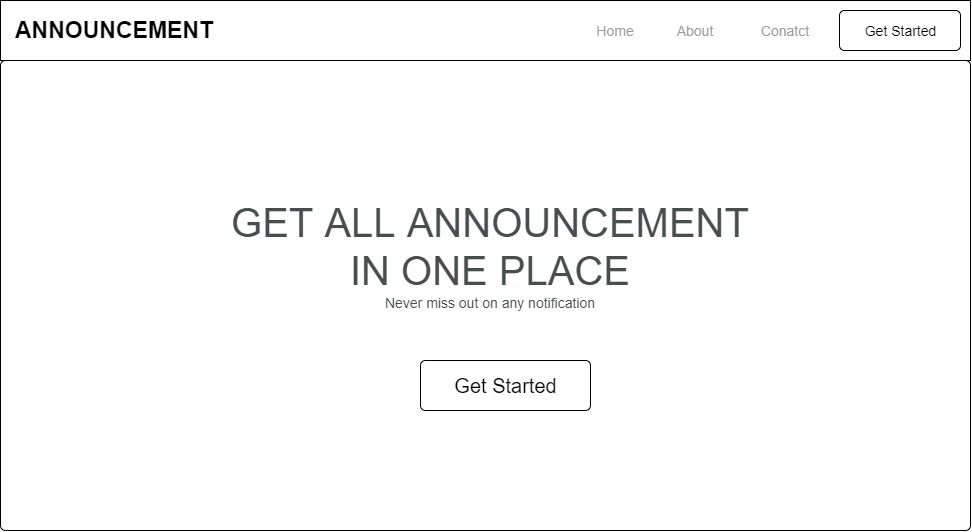
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User\_id** | **Username** | **Password** | **FirstName** | **LastName** | **Phone** | **Is\_staff** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**Table 3.4 Announcement** **output design table**

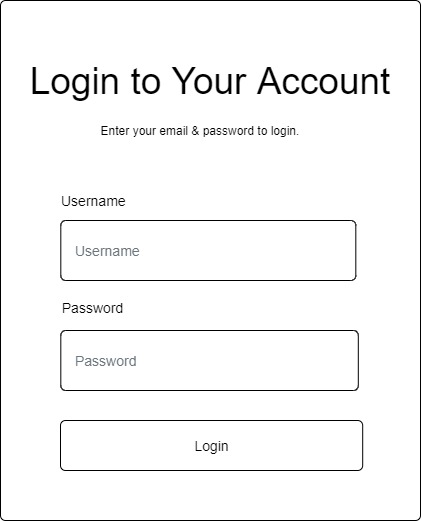
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Notice\_id** | **Notice\_title** | **Notice\_desc** | **Notice\_image** | **Category** | **Created\_date** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX |  | XXXX |

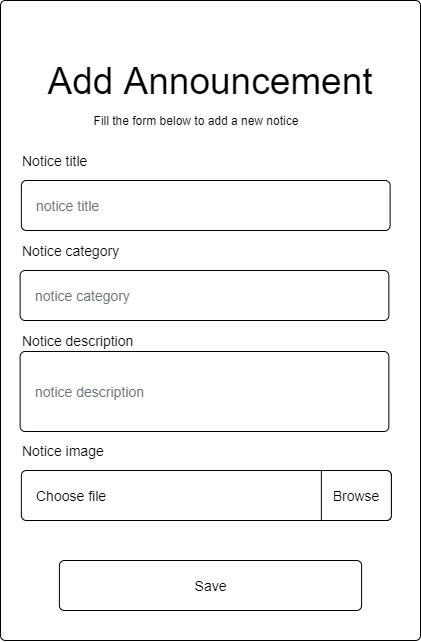
**3.6 Input & User Interface Design**

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 3.5 Announcement Home Page**



**Fig 3.6 User Login Screen**

**Fig 3.7 Add Announcement Screen**

**3.7 System Requirement**

Every piece of software has predefined system requirements that it must fulfill 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 for the frontend development while Django (Python) will be employed for the backend programming.