**CHAPTER THREE**

**Methodology and Design**

**3.1 Introduction**

A methodology is an approach to rigorous study or investigation, particularly to uncover new facts or information; hence, research methodology should be good enough to make the attainment of the established objectives attainable with certain components, such as methods of data collecting and design. This chapter includes the input/output specifications and system requirements for the development of a telegram bot that provides students with academic-related information, as well as the system modeling (use case, activity, and class diagrams).

**3.2 Methods of Data Collection**

It is crucial to acquire data and facts about the current system before implementing any system since one has to understand what is happening. Three techniques were used to conduct this study.

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

**3.2.1 Observation of the Work Environment**

This approach was used to collect information/data for this study by examining how the manual system was carried out, the method provides varying degrees of control over the context in which they are used, and the careful inspection revealed the obvious flaws in the present system.

**3.2.2 Documentation**

A secondary form of data acquisition is documentation. Journals, manuals, previous projects, publications, and other sources are used in this approach. This type of data collecting is employed because it provides a foundation for comparison with previous research. This includes the internet, a tool for gathering data. The internet was utilized to find information on topics that seemed challenging or unclear.

**3.2.3 Interview**

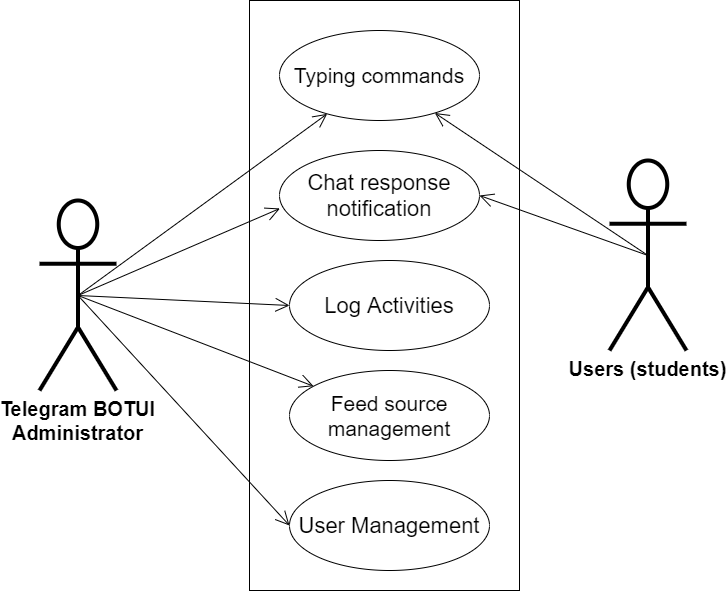
The primary goal of utilizing interviews as a data-gathering strategy is to collect data in a comprehensive and intensive manner. The researcher met with the project coordinators from the department and obtained trustworthy information based on the questions provided by the researcher.

**3.3 System Modeling**

A system model is a conceptual model that describes and represents a system. Any interaction between a group of components that work together to accomplish a single goal is referred to as a system. Visual models of the object-oriented software-intensive systems can be made using a set of graphic notation techniques that are part of the Unified Modeling Language, which is employed in this modern system design. Use case diagrams, class diagrams, and activity diagrams are among the UML diagrams used in this new design.

**3.3.1 Use Case Diagrams**

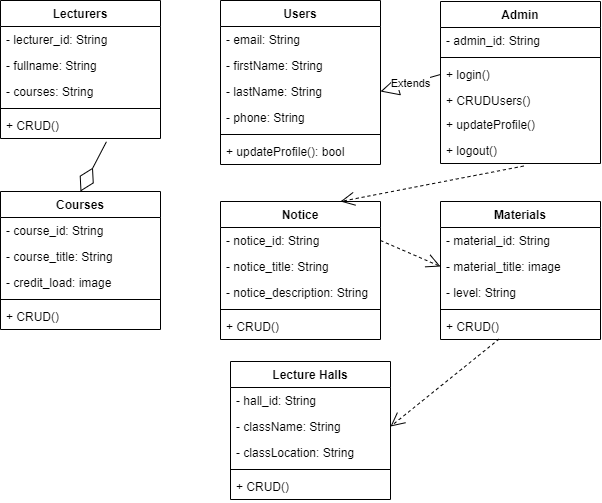
Use cases are collections of interactions between systems and users. Use case diagrams are used to visually summarize a system's functionality in terms of its actors, its goals (represented as use cases), and any dependencies between those use cases.



**Fig 3.1 System Use Case Diagram**

**3.3.2 Class Diagrams**

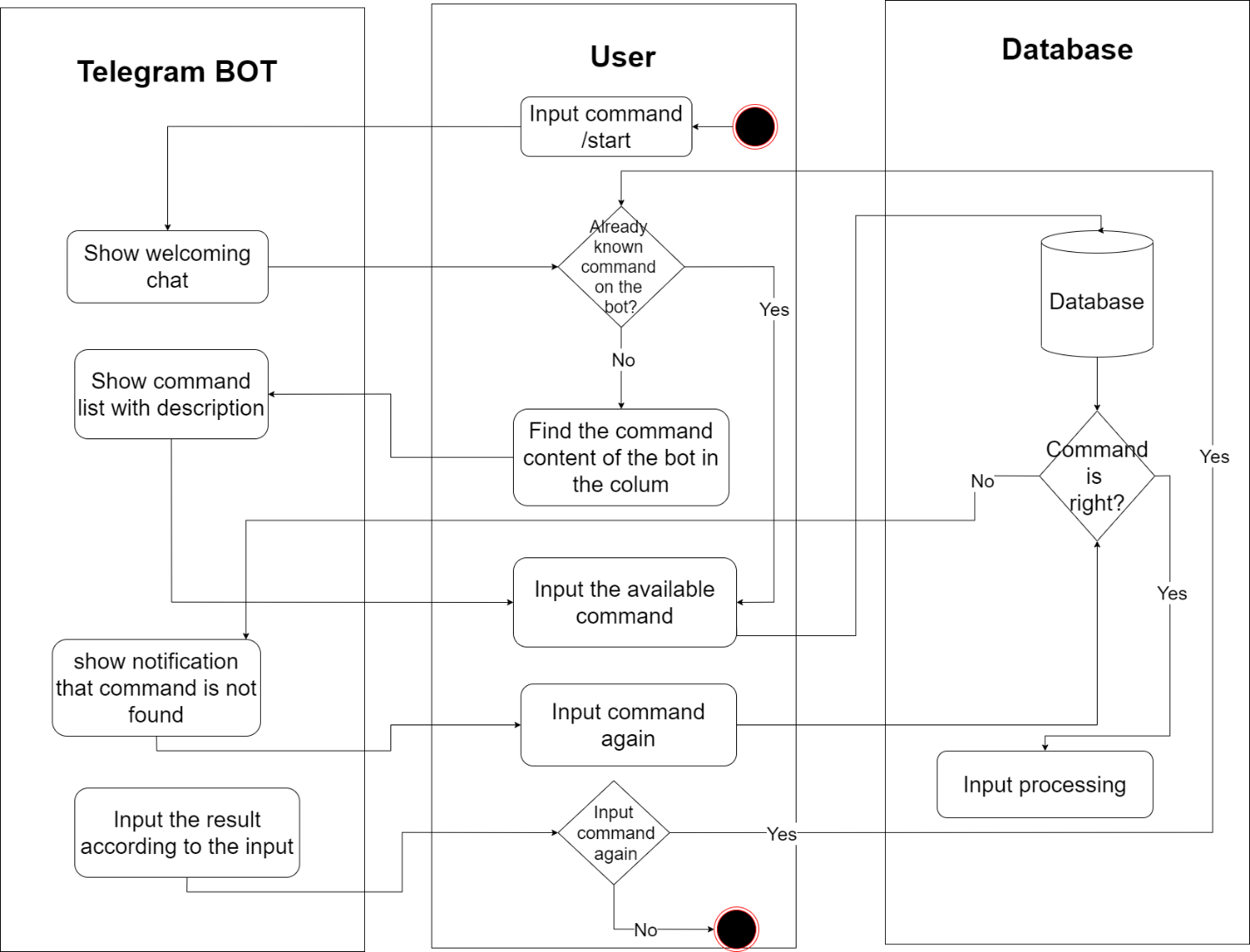
The Unified Modeling Language (UML) class diagram is an implementation of an independent view of how the system interface will be, with each class having its own properties and illustrating how they interact with one another. Class diagrams use the rules established by the Unified Modeling Language to visually depict the static structure and composition of a specific system (UML).



**Fig 3.2 System Class Diagram**

**3.3.3 Activity Diagrams**

Similar to a flowchart or a data flow diagram, an activity diagram visually depicts a sequence of events or the flow of control in a system, but it functions more like an advanced version of both.



**Fig 3.4 Application 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 and 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.

i. Users Table: contains basic information about all system users.

ii. Courses Table: contains department course information.

**Table 3.1 Users** **Table input specification table**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Email | String | 150 | User email address |
| Firstname | String | 150 | User first name |
| Lastname | String | 150 | User last name |
| Phone | String | 150 | User phone number |
| user\_id | String | 64 | A unique string for identifying users |

**Primary key:** user\_id

**Table 3.2 Courses** **Table input specification table**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Course\_title | String | 150 | Course title |
| Course\_load | String | 150 | Course credit load |
| Course\_desc | String | 150 | Course description |
| Course\_level | String | 150 | Level offering the course |
| course\_id | String | 64 | A unique string for identifying courses |

**Primary key:** course\_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 Users** **Table input specification table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Email** | **Firstname** | **Lastname** | **Phone** | **User\_id** |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |

**Primary key:** user\_id

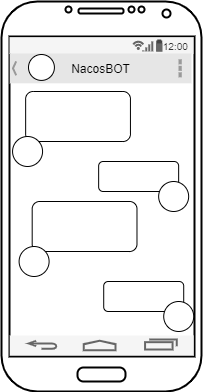
**Table 3.3 Courses** **Table input specification table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course\_title** | **Course\_load** | **Course\_level** | **Course\_desc** | **Course\_id** |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |

**Primary key:** course\_id

**3.6 Input & User Interface Design**

This shows a visual representation of the system interface; it will be made to be intuitive to use, quick to respond to, and visually appealing. Additionally, it will be properly protected, so signing in will be necessary to view some levels of the contents. A mid-fidelity wireframing application named Draw.io is used to assist with the designs.



**Fig 3.6.1 Chat Screen**

**3.7 System Requirement**

Every piece of software that is created has preset system requirements that it must meet in order to run at its best. However, the system requirements are the bare minimum hardware and software needed for the system's intended operation.

**3.7.1 Hardware Requirement**

System Hardware Requirement Include:

a. Minimum of 8 GB of RAM (Random Access Memory) installed.

b. Minimum of intel core i3 processor.

c. Minimum of 250GB HDD (Hard Disk Drive).

**3.7.1 Software Requirement**

The software requirements include:

a. At least windows 10 OS (Operating System).

b. Python Installation.

c. Vs. Code installation.

d. Browsers include Chrome and Firefox.

**3.8 Choice of Programming Language**

This research work will be a mobile-based application and will be implemented on a relational database system (SQLite). The student telegram Bot will be developed using the python request library, BotFather API, while Python will serve as the backend these are the modern language used in developing the system.