**CHAPTER THREE**

**Methodology and Design**

**3.1 Introduction**

This chapter expands on the examination of the suggested system and its architecture. That is to say, the components, features, and design elements (outputs, inputs, database) of the proposed system, Digital Diary. The same goes for the tools and technology that are employed.

**3.2 Method of Data Collection**

The research methodology used in this research work includes documentation and primary observation of the existing system.

**3.2.1 Observation Method**

This approach was used to collect information and data for this study by observing how the manual system functioned. Careful observation using the observational technique was used to identify the glaring flaws in the current system.

**3.2.2 Documentation**

A supplementary form of data acquisition is documentation. Journals, manuals, previous projects, play store reviews, and other sources are used in this approach. This type of data collecting was employed because it provides a foundation for comparison with earlier studies.

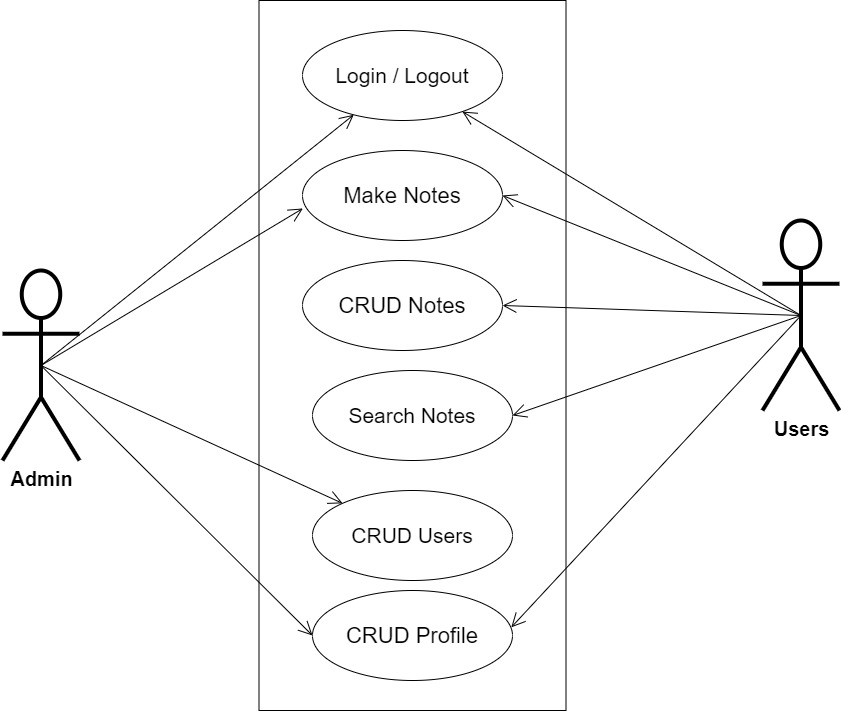
This project was carried out primarily using the agile development technique. The project will be able to adapt and grow as additional knowledge and techniques are acquired thanks to the agile approach. Breaking down each portion of the project into smaller phases will provide me with more time to focus on the parts that call for more reading and study. As this is an individual project, several techniques from the agile methodology won't work for a single person, but the fundamental development process will. These will consist of the more manageable tasks that are routinely monitored to make sure they are completed as effectively as possible. To make sure the system is on track to meet the needs that were previously outlined, regular evaluations of the system as a whole will be conducted, either by a developer or with users, to identify any changes that may be necessary to the system itself. Any additional features won't be added until the core functionality is performing to expectations.

**3.3 System Modelling**

The Unified Modeling Language was used in this project design which includes a set of graphic notation techniques to create visual models for the Object-oriented application. The UML applied in this new design includes Use Case Diagram, Class Diagram, and Activity Diagram.

**3.3.1 Use Case Diagrams**:

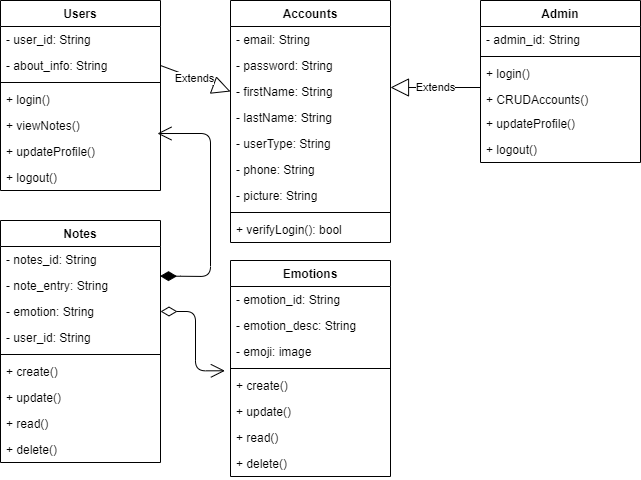
The use cases and primary tasks that a user will be able to perform within the program are depicted in the accompanying diagrams.



**Fig 3.1 System Use Case Diagram**

**3.3.2 Class Diagram**

The fundamental organization of each class in the system is depicted in the diagram below. Each class's essential variables and methods are displayed, and certain variables and methods have not been added to the diagram for the sake of clarity. There is also the opportunity to access and subsequently instantiate Main Activity, Home Page, and Analysis Home within each of the main classes. The navigation bar makes this possible, but these instantiations haven't been included in the diagram because it would complicate it.



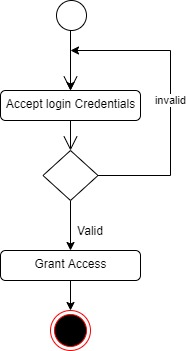
**Fig 3.2 System Class Diagram**

**3.3.3 Activity Diagrams**

Activity diagrams have been developed to further demonstrate how each class can access all others and to display the possible activities within each of the system's divisions.

**Login**

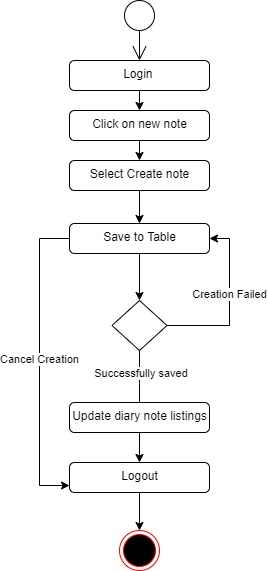
The process for gaining access to the system is depicted in the diagram below; to gain access, the email address and password must be accurate.



**Fig 3.3 Login Activity Diagram**

**Taking Diary Notes**

The process for making a diary note is depicted below



**Fig 3.4 Taking Diary Notes 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 convenience of retrieving and reading the data and assuring applicability throughout the internet. The following are some of the input specifications used in this project effort.

1. Accounts Table: contains basic information about all system users.
2. Note Table: contains every user note information.

**Table 3.1 Account Table input specification table**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Email | Varchar | 150 | Email for login (case sensitive) |
| Password | Varchar | 150 | Access Code (case sensitive) |
| Firstname | Varchar | 150 | User first name |
| Lastname | Varchar | 150 | User last name |
| Phone | Varchar | 11 | User phone number |
| Picture | Varchar | 100 | User profile picture |
| acct\_id | Varchar | 64 | A unique string for identifying users |

**Primary key:** acct\_id

**Table 3.2 Note Table input specification table**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Note\_id | Varchar | 150 | A unique string for identifying tricycles |
| Note\_entry | Varchar | - | Diary note |
| User\_id | Varchar | 150 | A unique string for identifying user |

**Primary key:** note\_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 Output Design Table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Email** | **Password** | **Firstname** | **Lastname** | **Phone** | **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 Note Output Design Table**

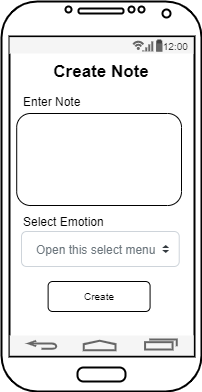
|  |  |  |
| --- | --- | --- |
| **Tricycle\_id** | **User\_id** | **Note\_entry** |
| XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX |

**3.6 Input & User Interface Design**

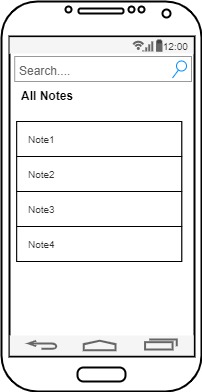
This shows a visual representation of the system interface; it will be made to be intuitive to use, quick to respond to, and visually appealing. Additionally, it will be properly protected, so signing in will be necessary to view some levels of the contents. A mid-fidelity wireframing application named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



**Fig 3.5 User Login Screen**



**Fig 3.6 Create Note Screen**



**Fig 3.7 Search Note Screen**

**3.7 System Requirement**

The goal of this project is to offer users comprehensive solutions for keeping track of their everyday activities, thoughts, and chores using only their mobile devices. Users will be able to sign in or register with ease. The following list of initial functional requirements.

* + 1. **Hardware Requirements:**

1. CPU: i5 core processor
2. Clock speed: 2.4 GHz
3. Main memory: 8GB RAM
4. Secondary memory: 500GB Hard Disk
   * 1. **Software Requirement:**

To build this project, there is the utilization of the following Software resources:

1. At least Windows 10 OS (Operating System).
2. Flutter Installation.
3. Vs. Code / Android studio installation.
4. Emulator installation.

**3.8 Choice of Programming Language**

The proposed design will be implemented using Flutter for its user interface (frontend) while Python will be used for the backend programming, Sqlite3 will be used for its database due to its portability, and Django REST Framework will be employed for its REST-full APIs, the combination of the above modern technology forms the technology for this research work