## Criterion B: Design

---

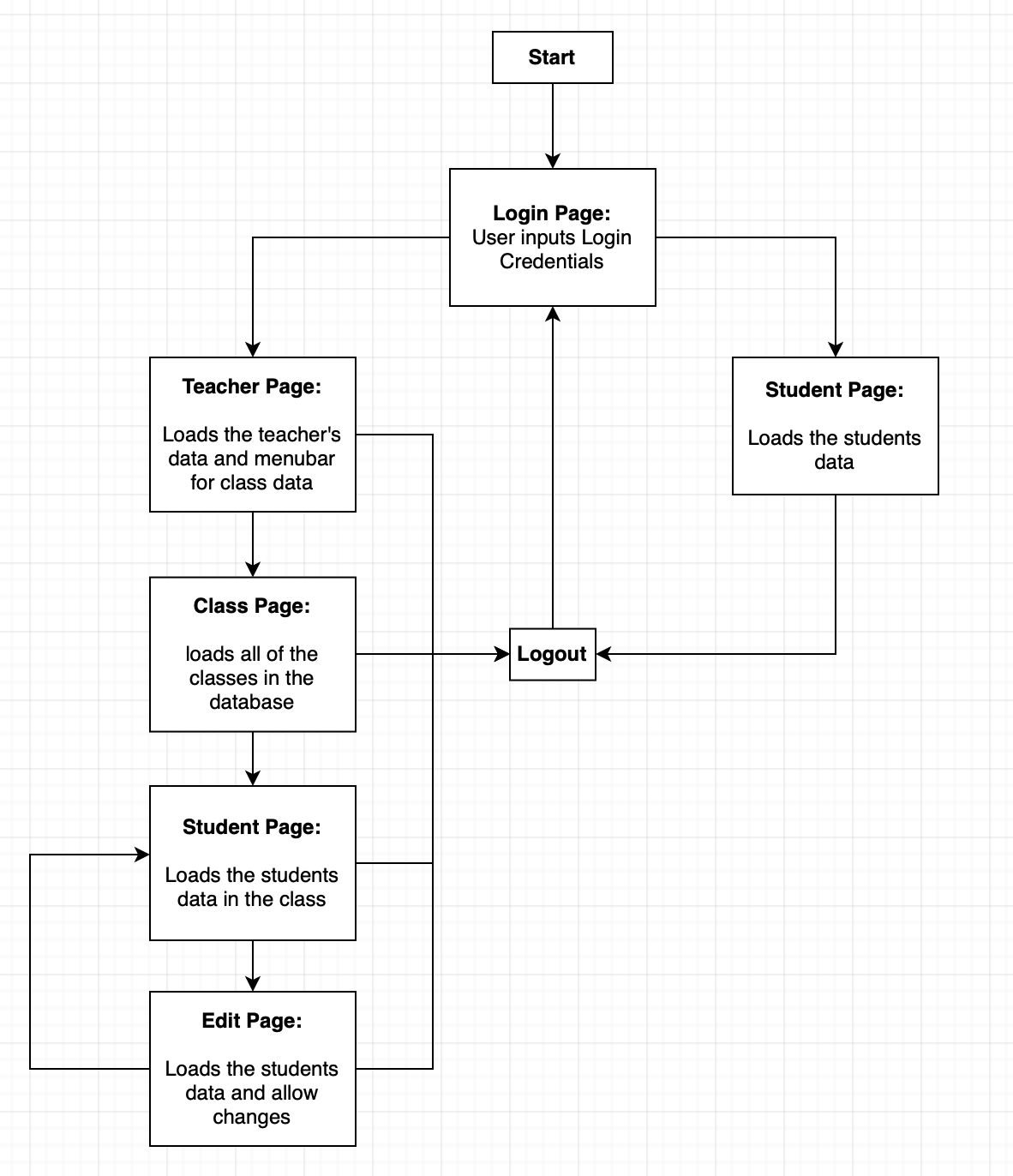
1. Class Variable / Data Dictionary

**Table1. accounts (Database)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable Name | Data Type | Default Value | Extra Information | Developer notes |
| IDUsers | Int(50) | *None* | Auto\_Increment | ID of the user, to count the number of users in the database |
| UIDUsers | String | *None* | - | Username of the account for login. This will be unchangeable. |
| EMAILUsers | String | *None* | - | The Email of the user. This can also be used to log in with the connected account. |
| PSWDUsers | String | *None* | - | The password of the account. This will serve as a security measurement. This password will be hashed so that the person who has access to the database will not be able to see the password of the account. |
| CLASSUsers | String | *none* | - | this will determine which class is the student is in. |
| ATUsers | String | *None* | - | This will specify whether the account is a Student or a Teacher. This will be used to direct the user to their specified direction |
| PSUsers | Int(50) | *None* | - | This will serve as the points of the student. (disiplinary points ) |
| Other content in the user’s account | | | | |
| DOB | Int | *None* | - | Date of Birth of the User |

1. System Flowchart:

**Figure 2.1 - User Navigation ( flowchart )**

This shows the user navigation of both student and teacher from the main screen to the following.

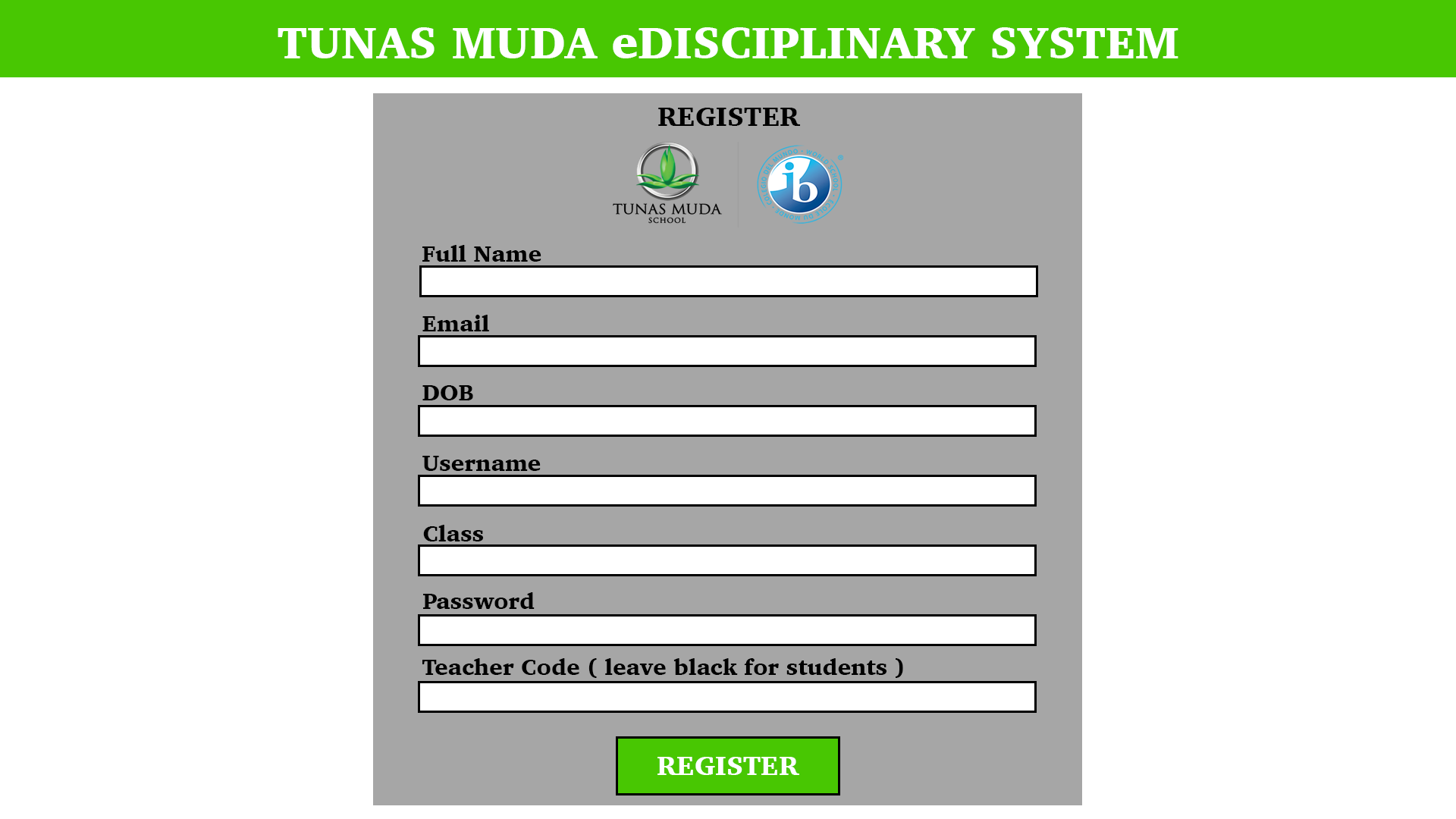
1. GUI Designs

( see Appendix B for initial design meeting with client )

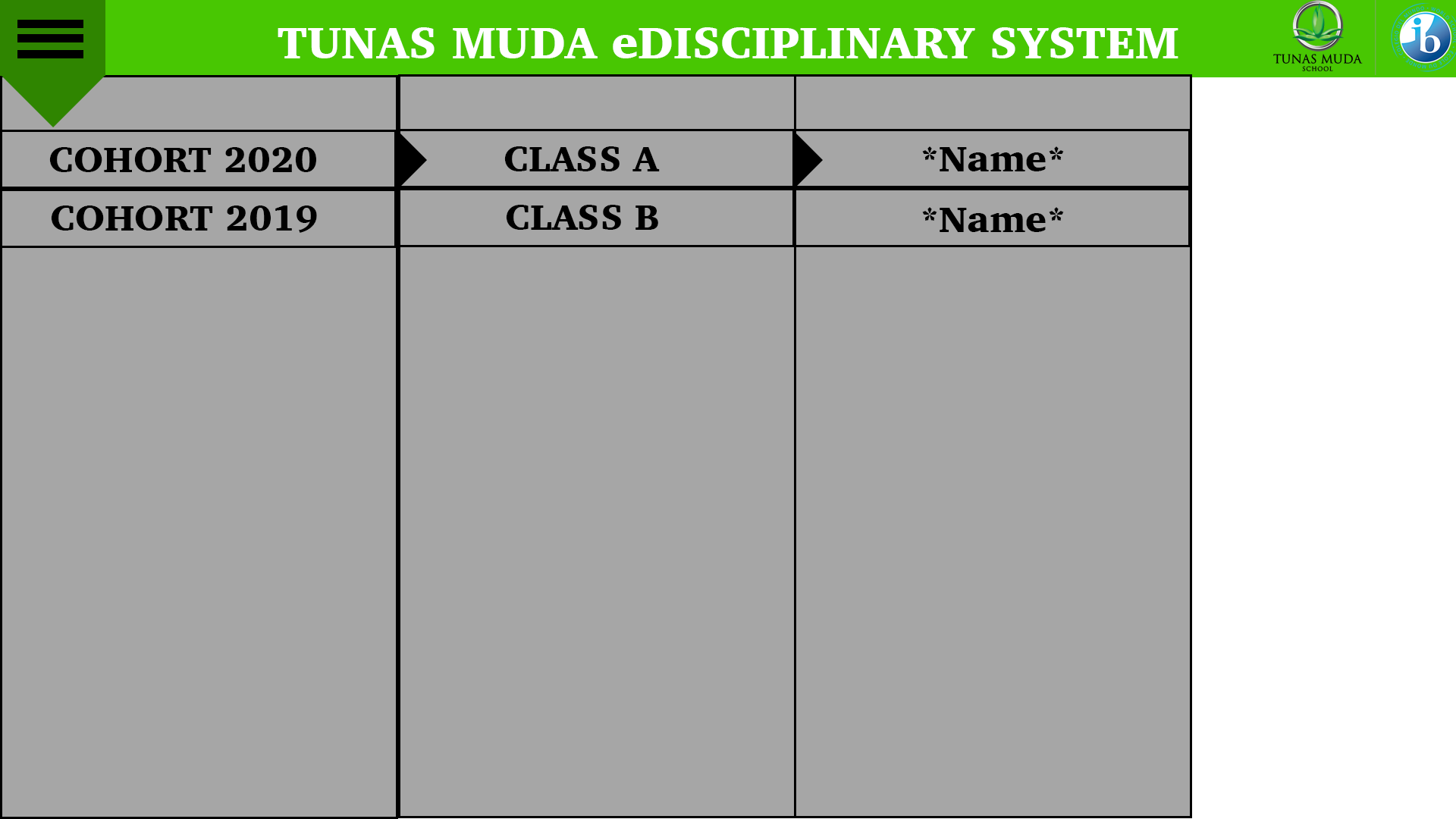
* **Sketch of Login page**

****

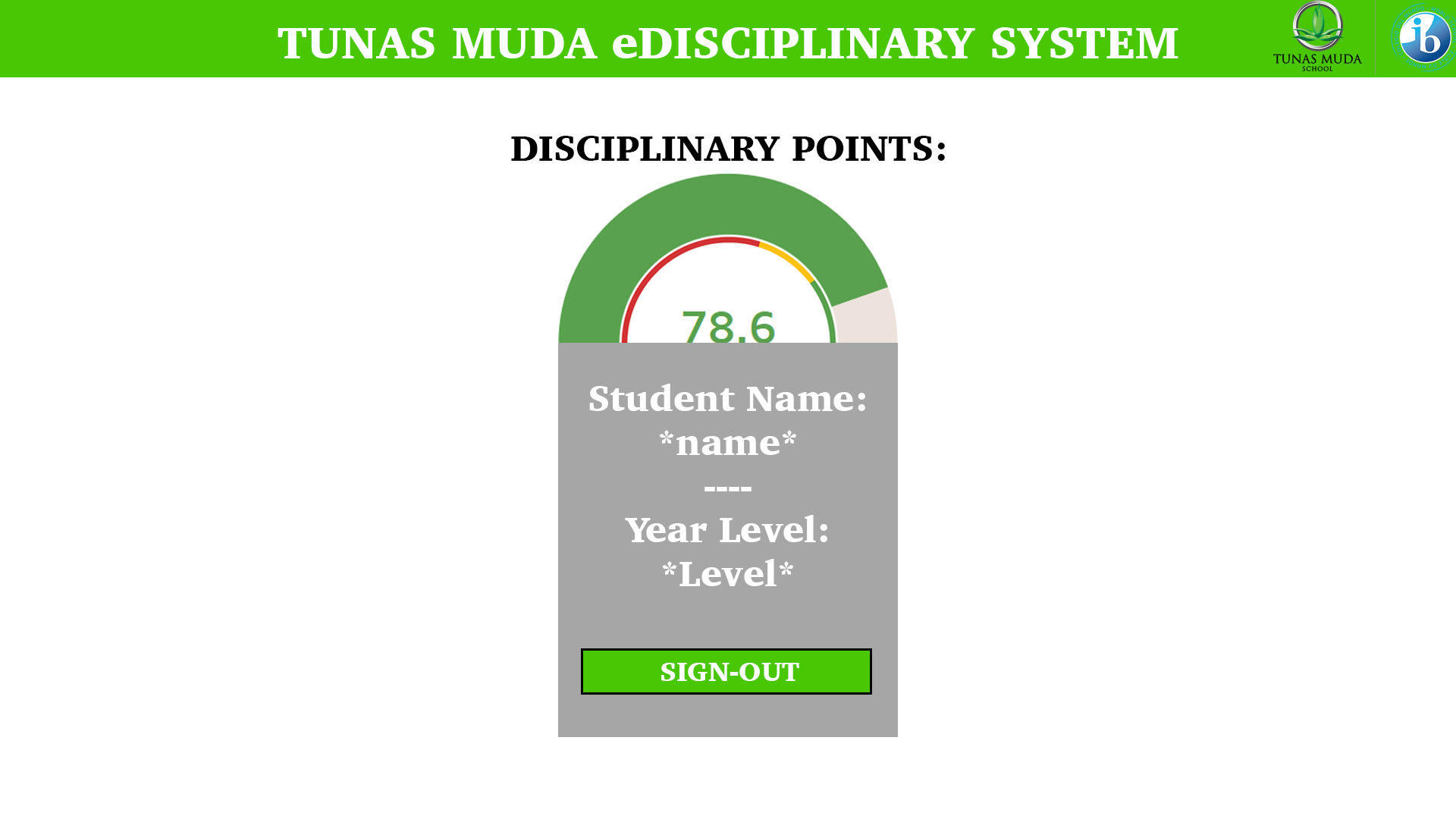
* **Sketch of Register page**

****

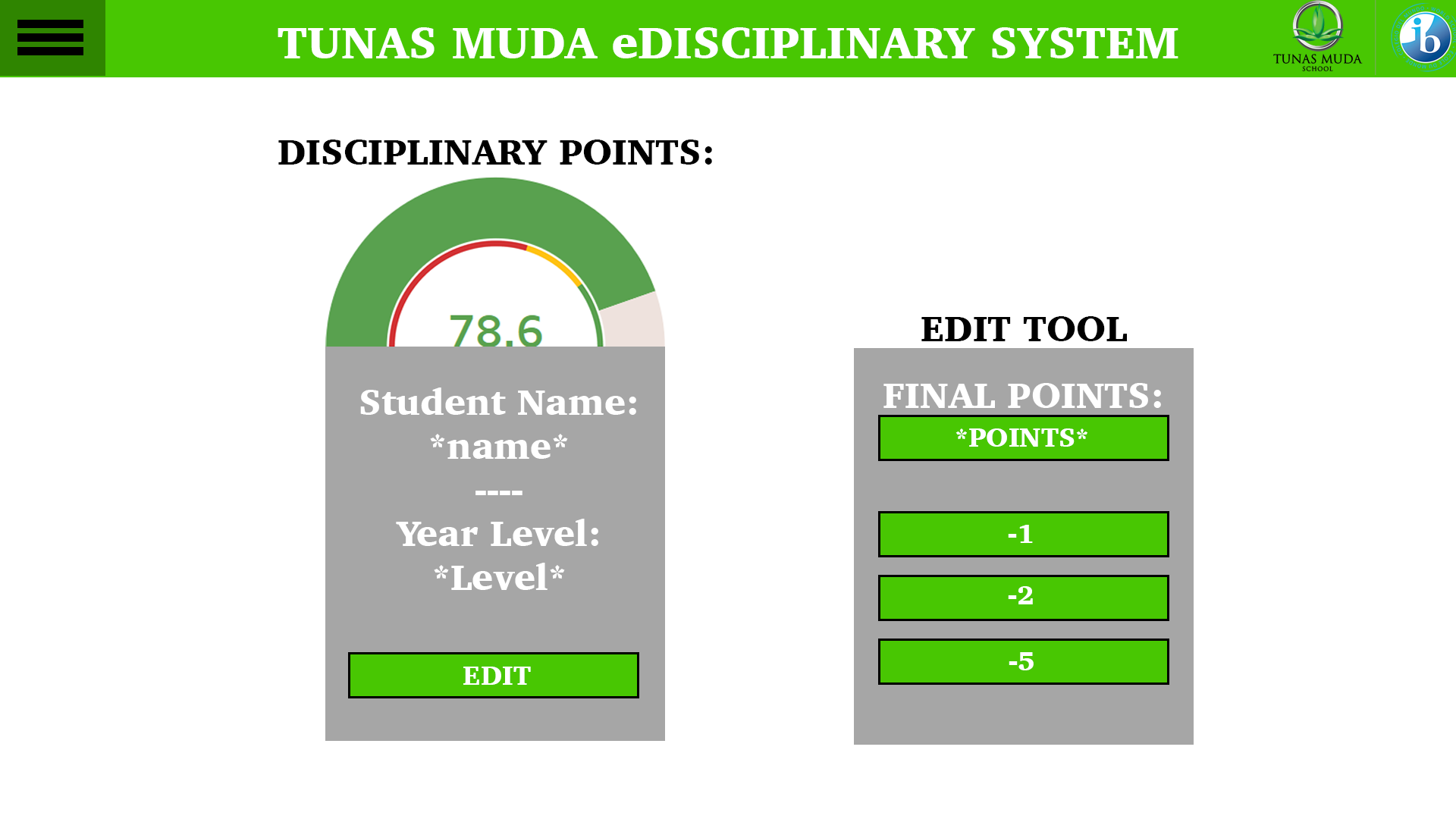
* **Sketch of Find Class page**

****

* **Sketch of Student point page**

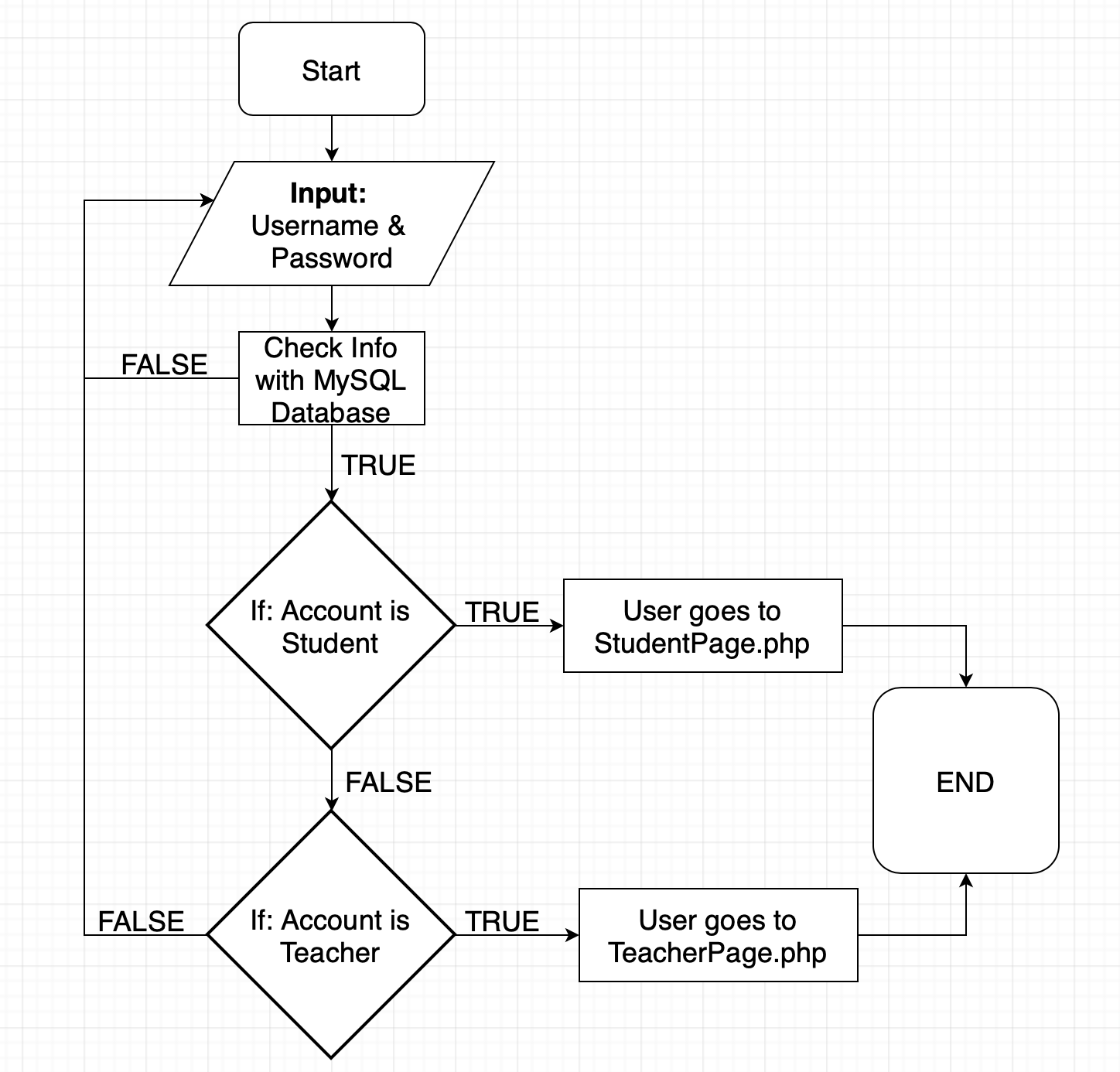


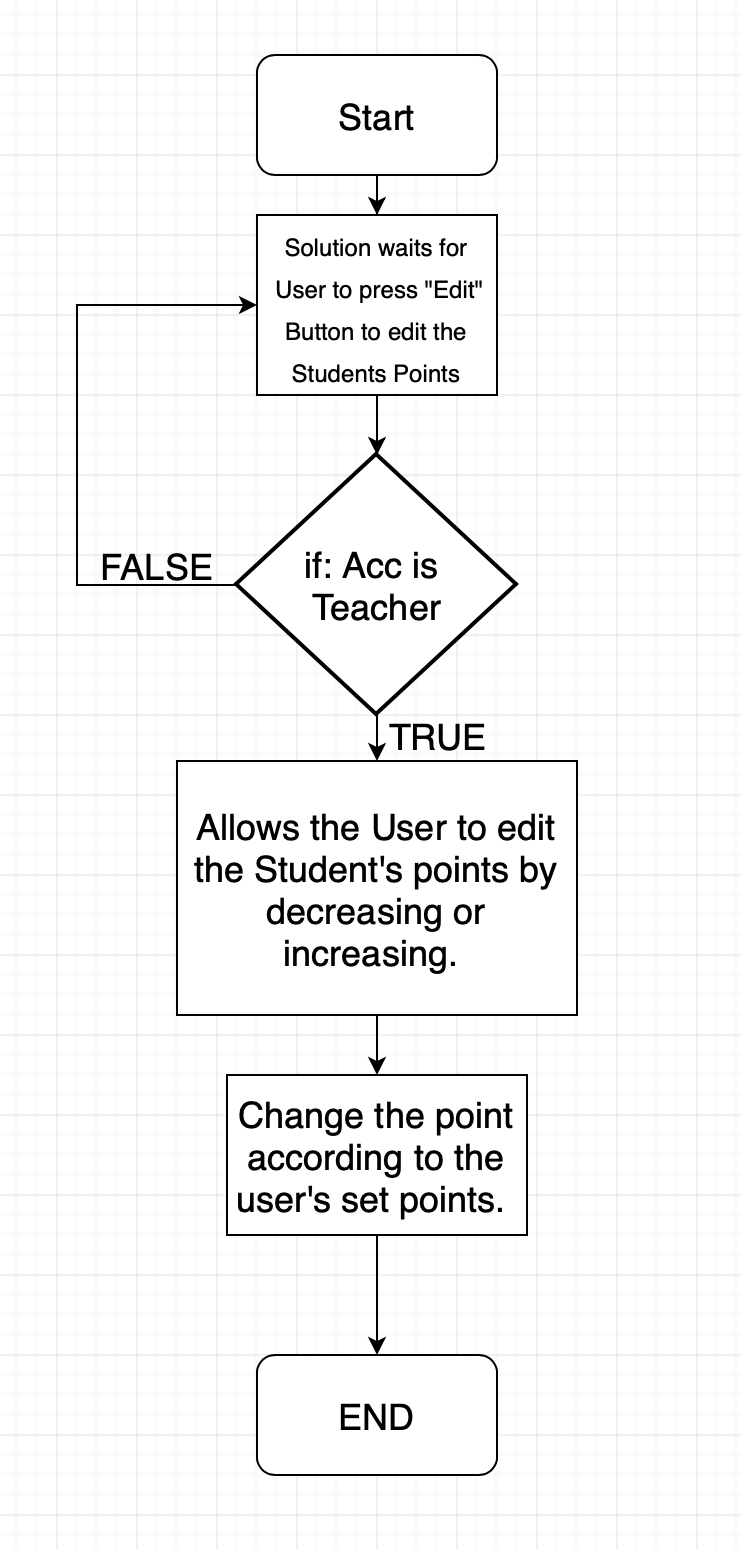
* **Sketch of Edit page**



1. Flowchart:

* **Figure 4.1 - Login Process ( Student / Parents & Teachers )**



* **Figure 4.2 - Disciplinary system / edit points**

1. Testing methods

|  |  |  |
| --- | --- | --- |
| Task | Test method | Notes |
| 1. The students and teachers are able to access the website with their own login data. | Input login credentials of both students and teachers | Check whether the system is able to recognize a student or a teacher is logging in |
| 1. The user is able to register new students, parents and teachers. | Register a new account to the database then checking the database to see if it is inputted | Check whether all of the inputs are correct and that it is inserted into the database. |
| 1. The teacher is able to edit the points of the chosen student. | Change the points of the selected student and | Input from the main page for teachers, then to the search, then to find the student. |
| 1. The student’s data is shown clearly by the class the student is in. | Change the points of the student through edit page. | Change the student’s points and the display at the student’s side is also correct |
| 1. The program should recognize the user ( student or teacher ) and direct to the correct page | Login with 2 accounts, one student and one teacher. the program should recognize the account and direct it to the correct pages. | Check the code if any problem occur and this will help the program be used by both student and teacher. |
| 1. Error handling should be present throughout the program. For example, login error, wrong inputs, or unable to find. | Input all the incorrect inputs and check whether the page is redirected correctly. | Incorrect Login = Fill again and notification  Incorrect class input = Fill again and notification  Missing registration data = notify  Registration password not the same = input password again |