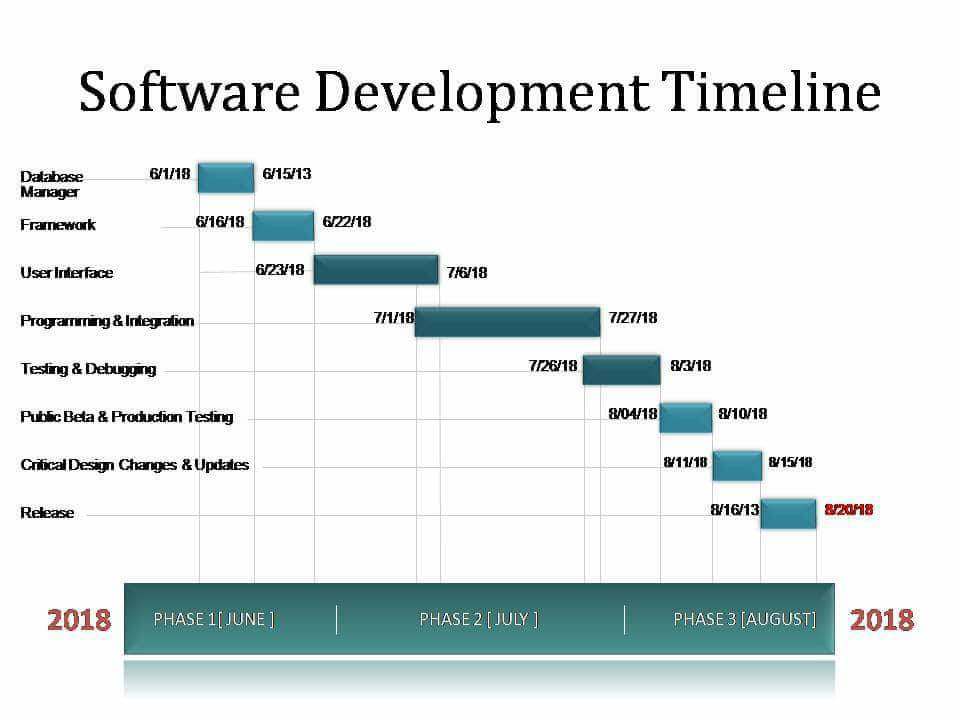
**CHAPTER V**

**Results and Discussion**

**5.1 Development Phase**

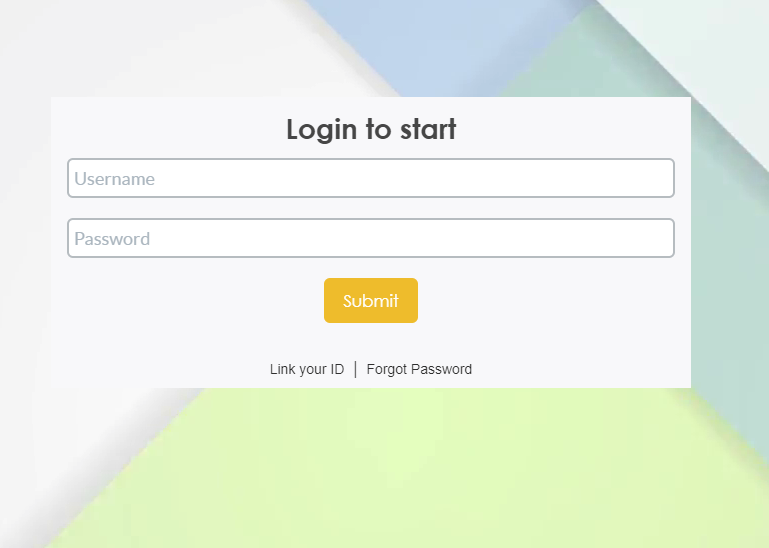
The development phase of the system, shown in Figure 5.1, was scheduled in a timeline given to us by the programmers. Started off with the manifestation and refinement of the proposed database. Following the creation of the database, the programmers selected a pre-built or existing framework of a website that best suited our system’s needs. Following the framework selection and reconfiguration, the programmers started creating the graphical user interfaces without its functions and links enabled. After the altering of the objects in all the UIs of the users, they began to code the functions and processes that is required to majority of the objects to function. While the interfaces are undergoing its development, the programmers slowly integrate them accordingly by the system’s hierarchal levels. They did various testing, most of which they followed from the functional test of our proposed system in our documentation. Once some the functions of each page of various users are working, the programmers uploaded the website in their company’s host server. As the development phase went along the timeline, with some minor setbacks or delays, the programmers managed to hasten the development with borrowed time due to the critical changes and requirements that are required by the proponents for the system to be mostly presentable with majority of the functions working regardless of the delay of the aesthetics upgrade to the website. Alpha testing was later conducted by the programmers and proponents for the checking of the requirements of the system.



**Figure 5.1 Given Software Development Timeline**



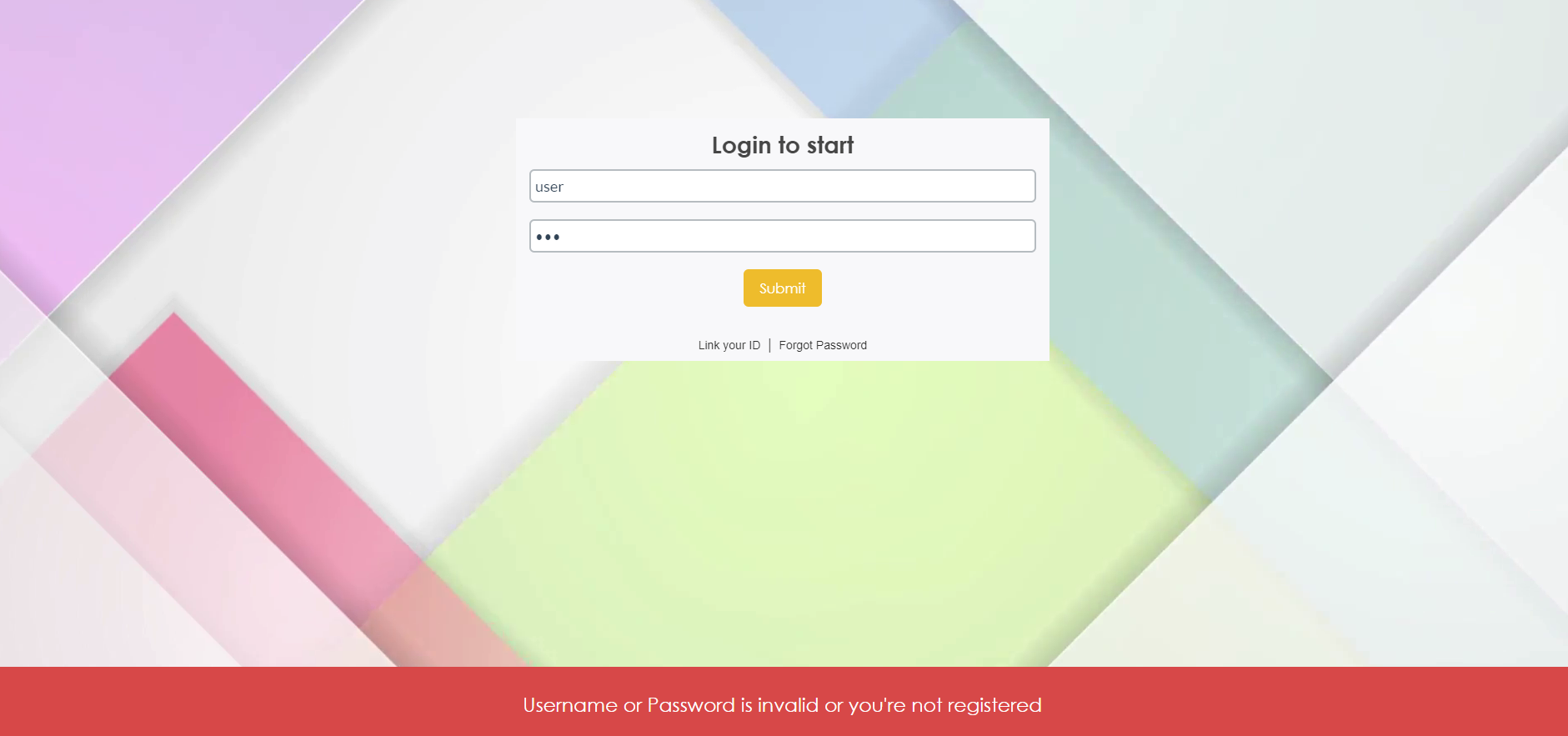
**Figure 5.2 Proposed Login Page**

  
**Figure 5.3 Revised Login Page**

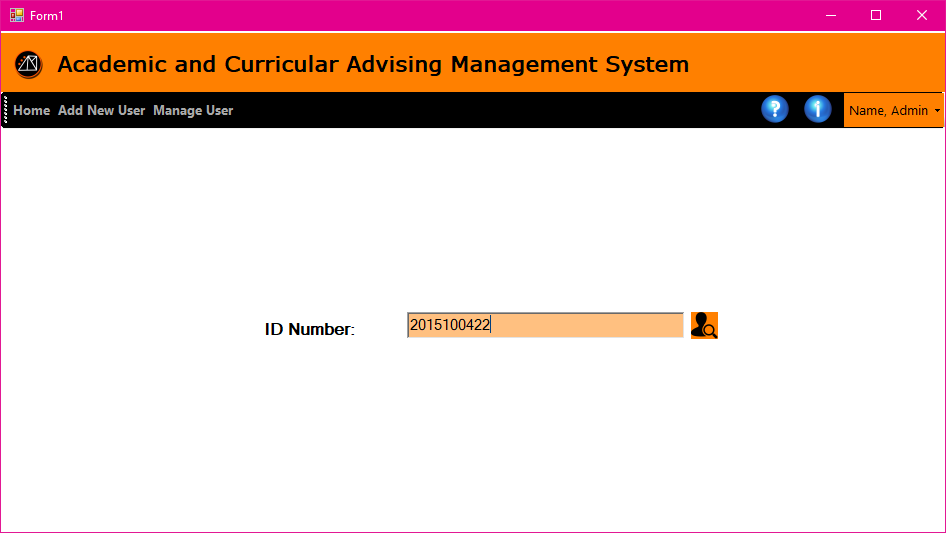
In Figure 5.2 and 5.3, we can see the drastic changes in the aesthetics. We can also observe the modernization of the design of the UI. We changed the Forgot Password Button to a quick link to follow the trend of the modern website standard designs.



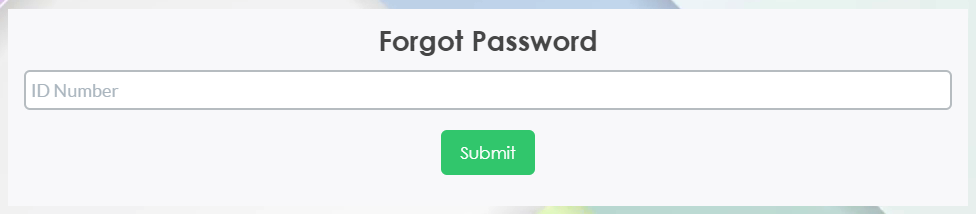
**Figure 5.4 Proposed Popup for Invalid Log in Information**

  
**Figure 5.5 Revised Popup for Invalid Log in Information**

With Figures 5.4 and 5.5 we can see that the popup is no longer a sort of new window popup but a label popup on the bottom of the website. Also it is more efficient to do a label type popup than a floating webpage popup.

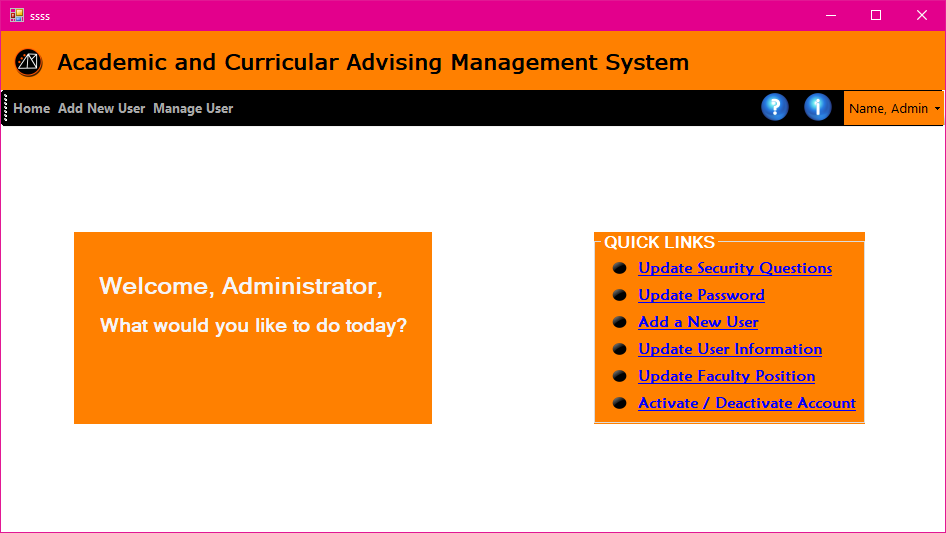


**Figure 5.6 Proposed Account Recovery**

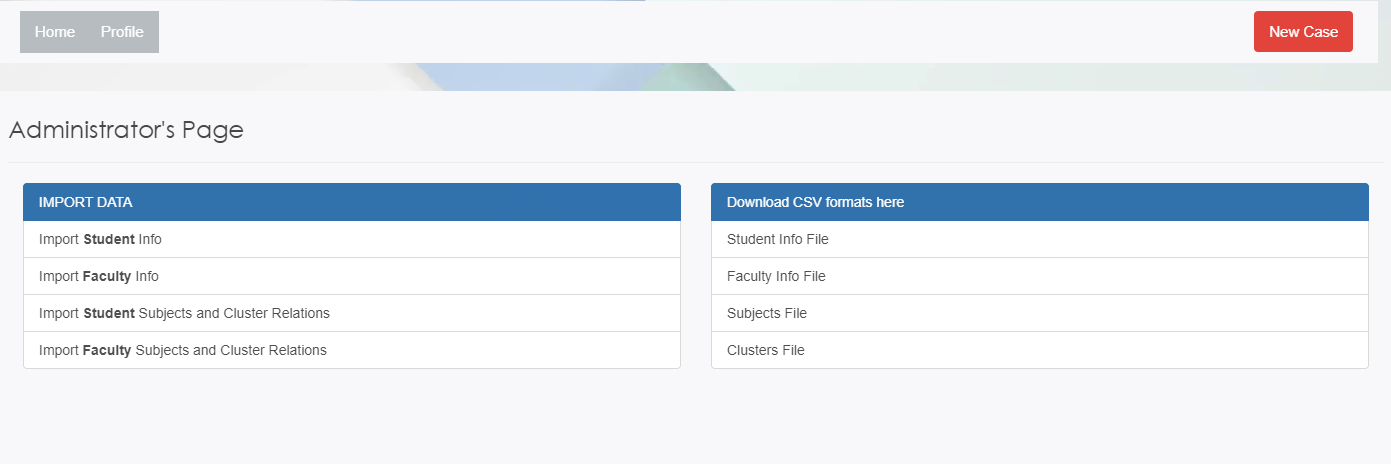


**Figure 5.7 Revised Account Recovery**

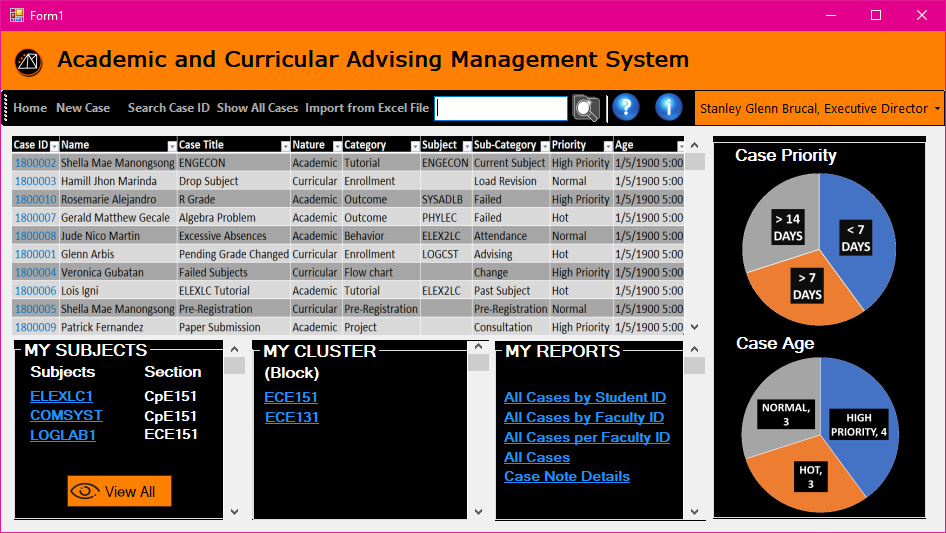
In Figure 5.6 and 5.7 we can see that there were less changes in terms of the graphics of the user interface. The menu bar on top of the proposed account recovery was removed in the forgot password page or account recovery page because it was unnecessary for this page. A submit button is added in the revised account recovery page because it was not present in the proposed, or rather it was not clear what the function of the button on the side of the textbox.



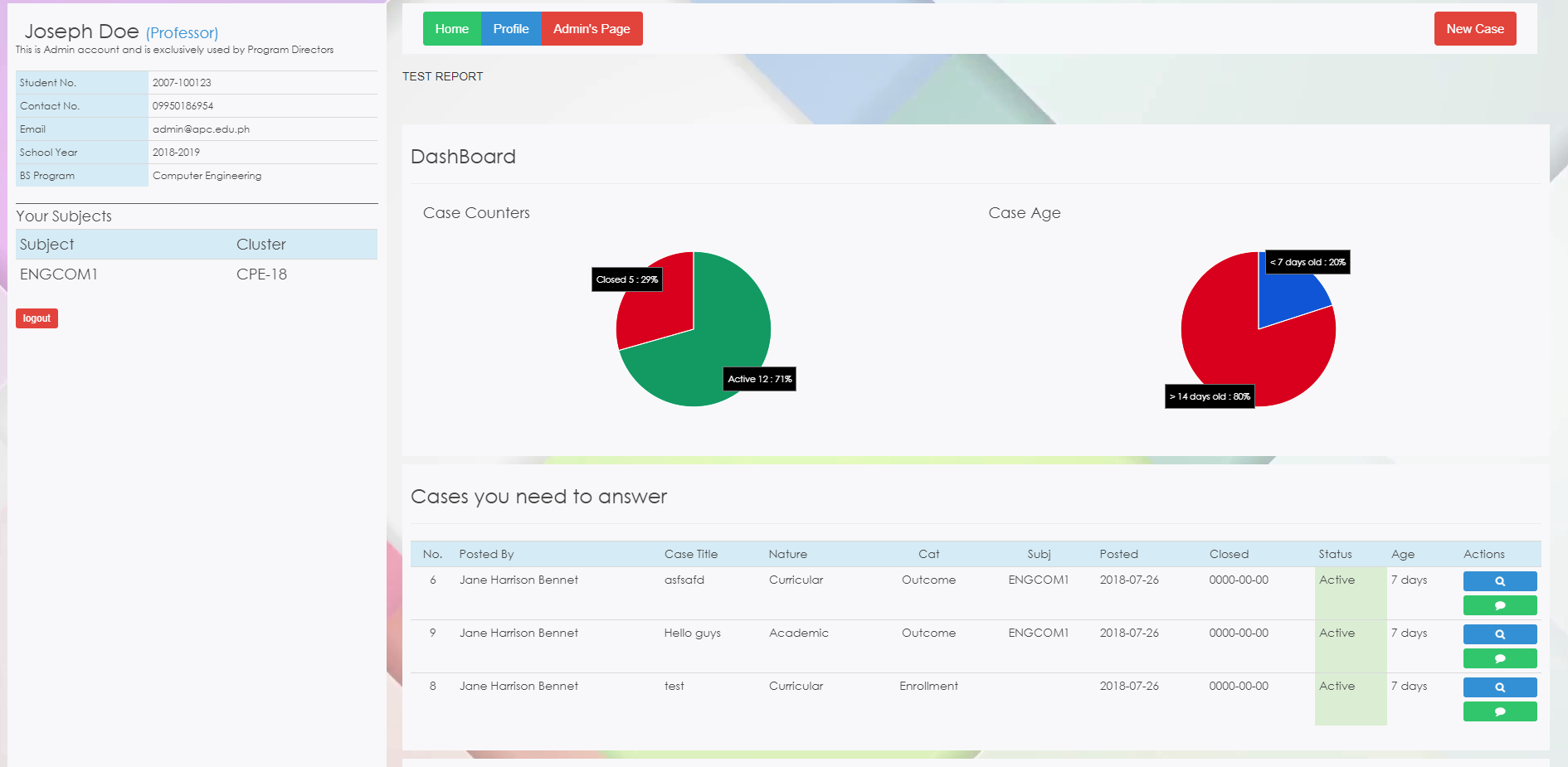
**Figure 5.8 Proposed Admin Homepage**

  
**Figure 5.9 Revised Admin Page**

For the Admin Page in Figures 5.8 and 5.9, again we could observe the modernization of the style of the layout of the page. Quick links that have no underline and link type color was changed to a regular text that is clickable in the revised admin page. The menu bar was also separated in the admin page to the main homepage. The dashboard contains all the other functions, while the admin privileged functions was placed in a new page in the revised version

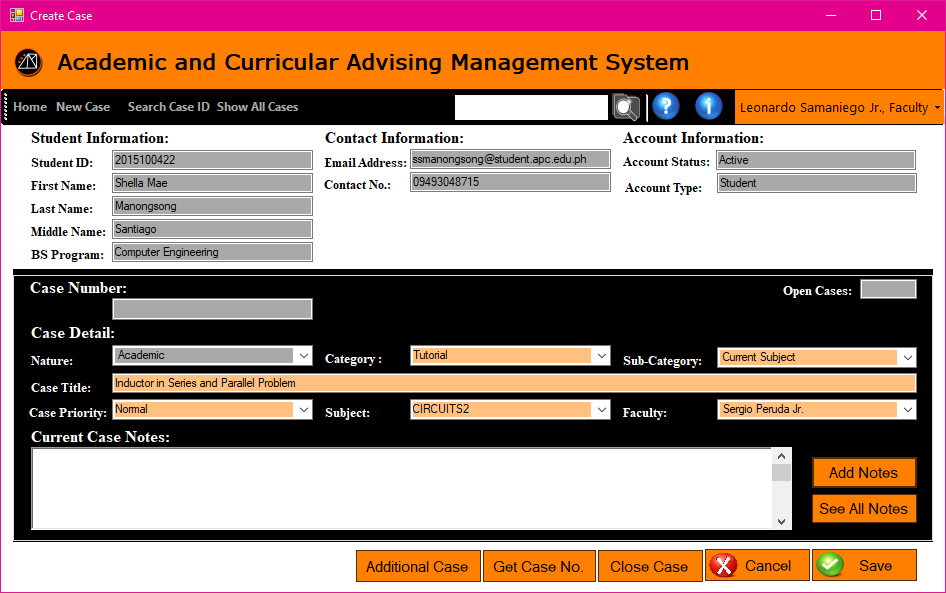


**Figure 5.10 Proposed User Homepage**

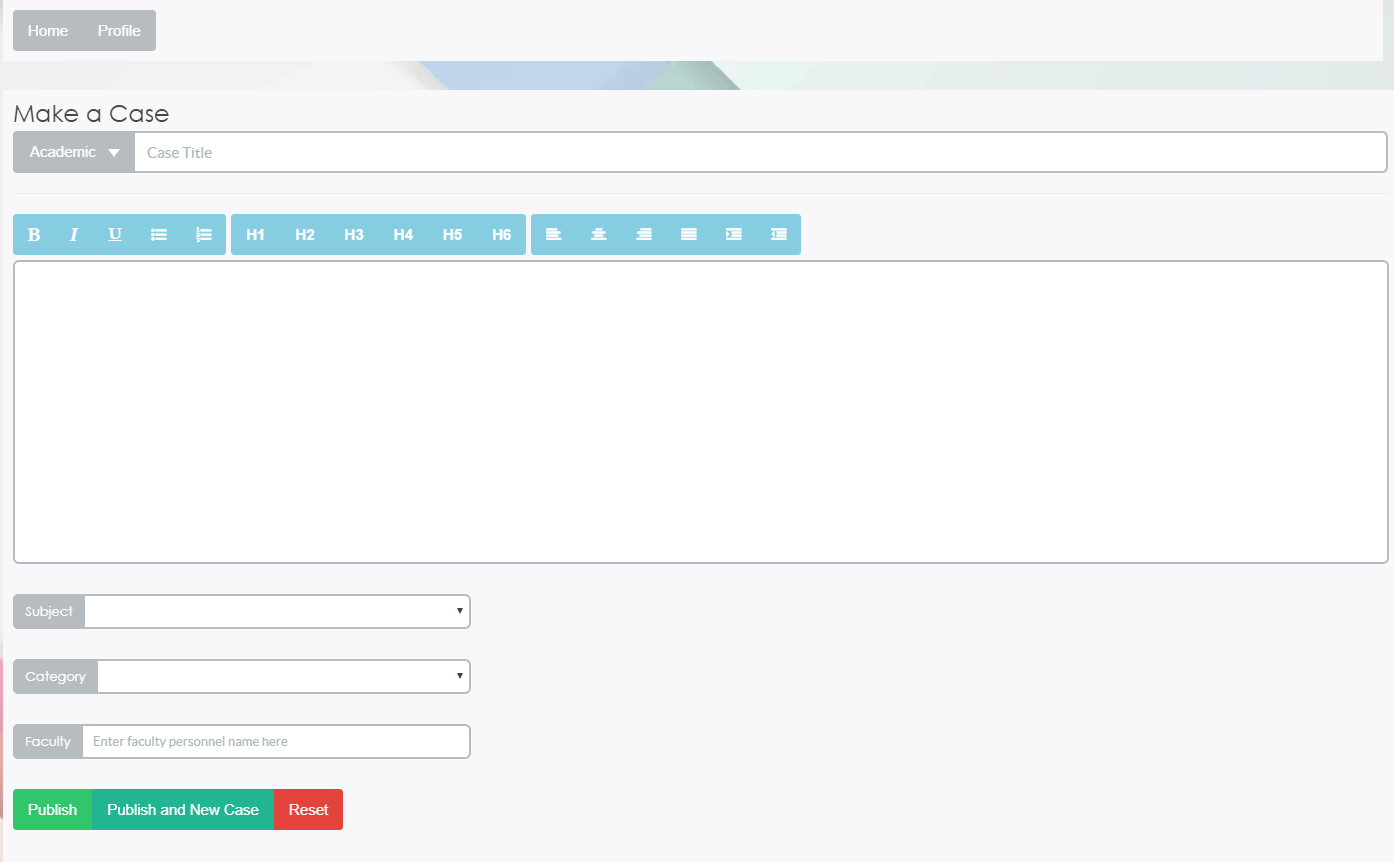


**Figure 5.11 Revised User Homepage**

The newer homepages in Figures 5.10 and 5.11 are quite similar to the but with the newer revised homepage having a modern look. Some other buttons and objects are rearranged. And other functions were added like the buttons on the side of the cases which are helpful to all users because it is self-explanatory and understandable with the symbols of the button the functions of each side buttons in the case overview in the dashboard.

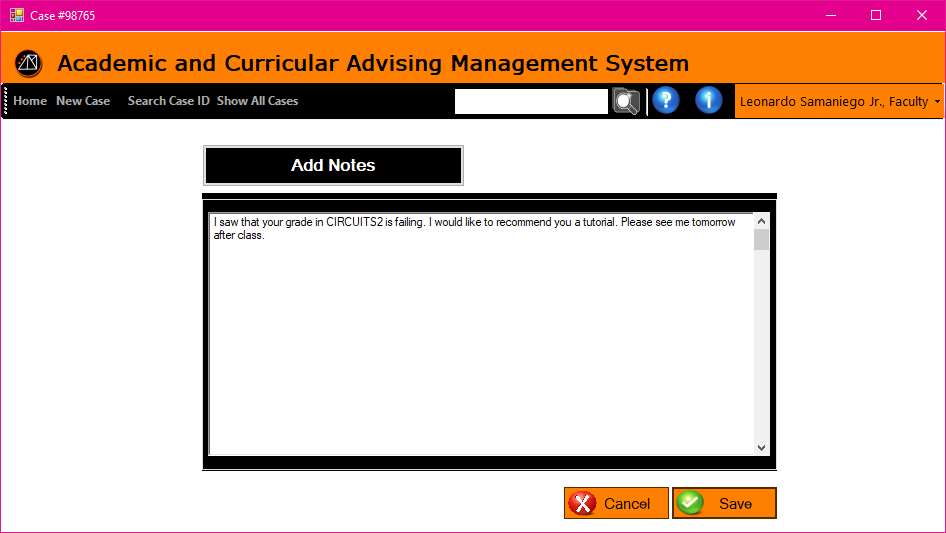


**Figure 5.12 Proposed Case Page**

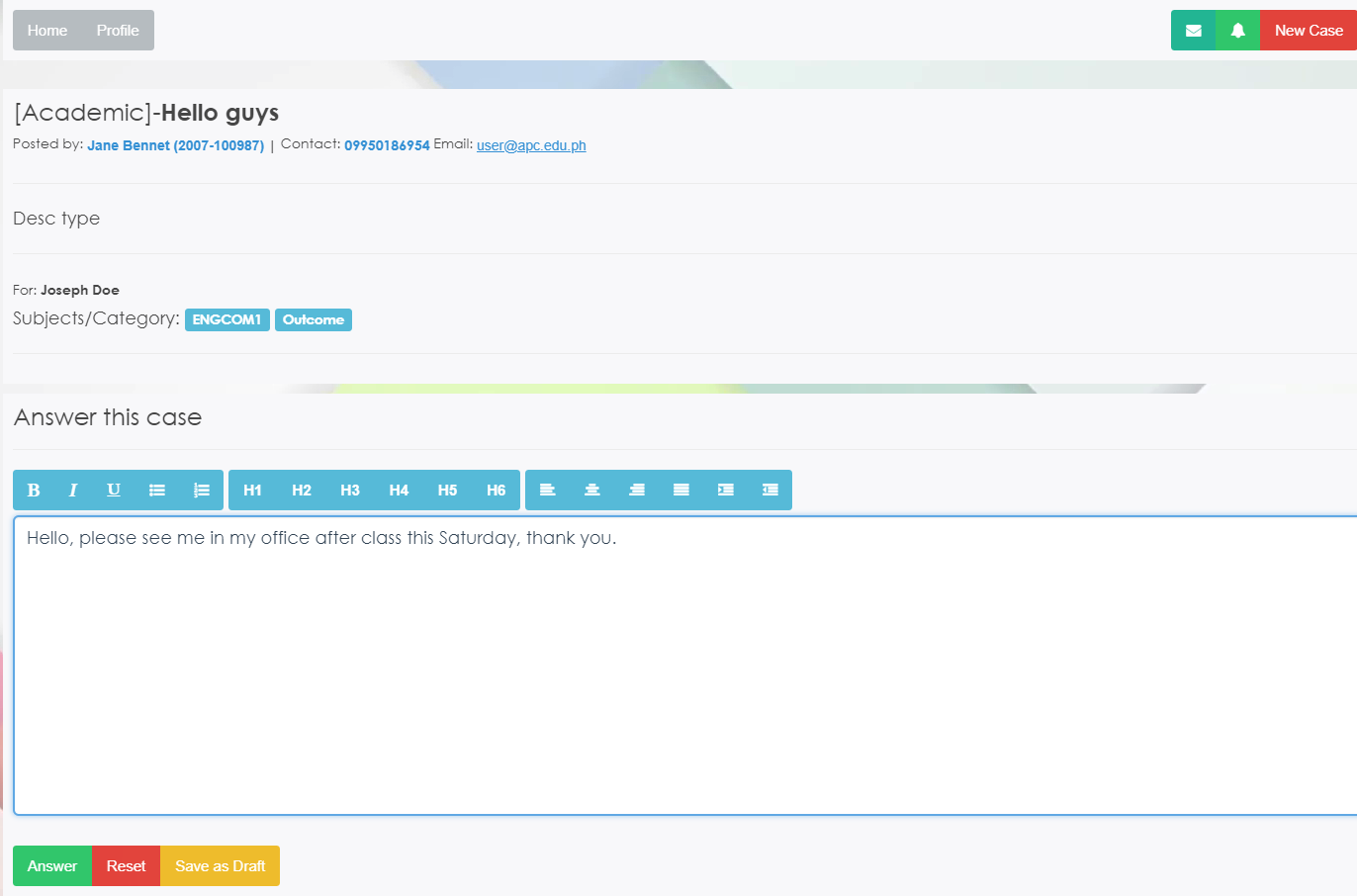


**Figure 5.13 Revised Case Page**

The case creation pages, found in both Figure 5.12 and 5.13, of the proposed version compared to the revised version is very far off from each other in terms of the style and design of the GUI. The proposed version is a more vintage type of program-type array of objects that are usually used in older ticketing systems. While the newer version is simplified, clean, and has extra text functions which can alter the properties of the input text. The revised case page is as it is now because it was derived from a framework used by the programmers.

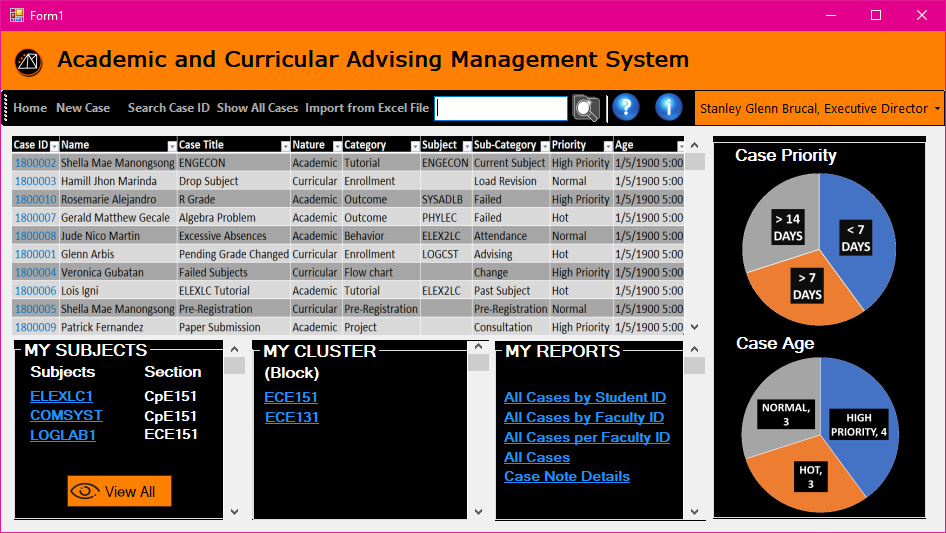


**Figure 5.14 Proposed Case Notes Page (Add Notes)**

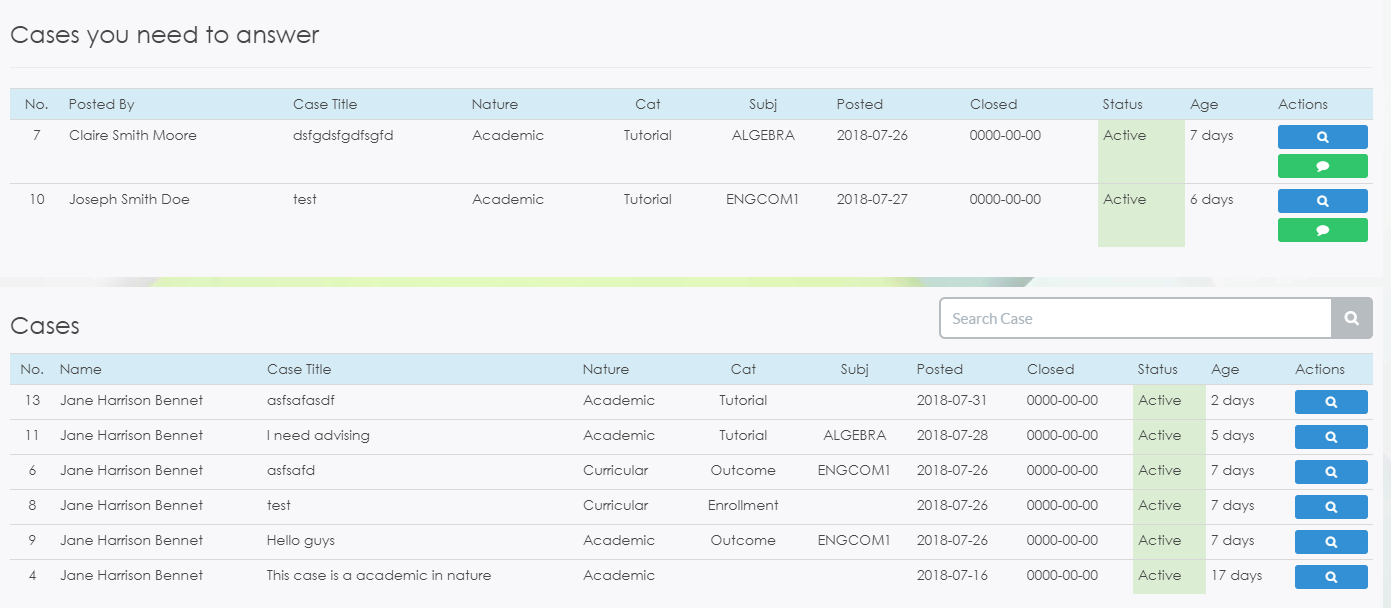


**Figure 5.15 Revised Case Notes Page (Add Notes)**

The case notes page in 5.14 proposed case note page was sort of an old design compared to its newer version case not page in the revised version in 5.15. Again bringing up the modernization of text input like what we see in email, the text properties can be altered with the buttons above the text box input. We added the button, save as draft, so that the user can save to their drafts the cases they would like to save and not submit in the meantime.

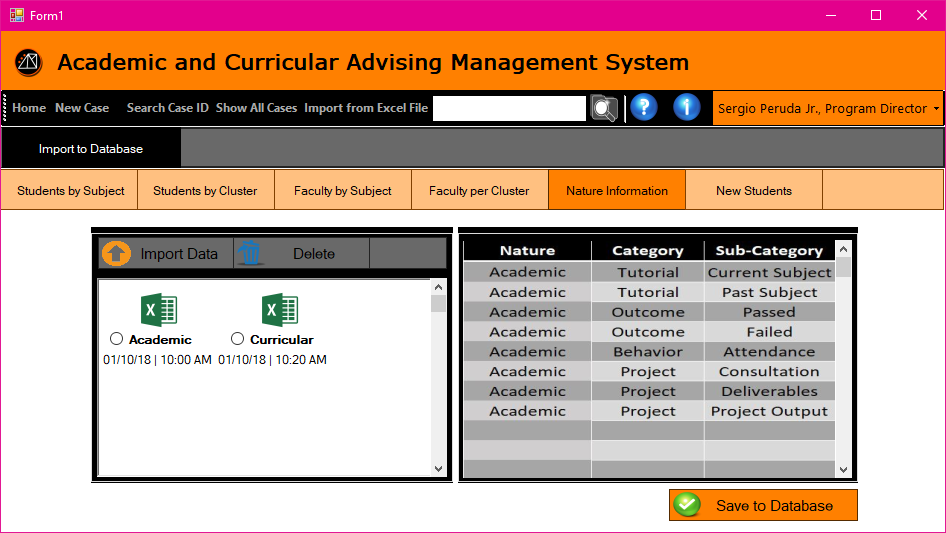


**Figure 5.16 Proposed Global Search Bar**

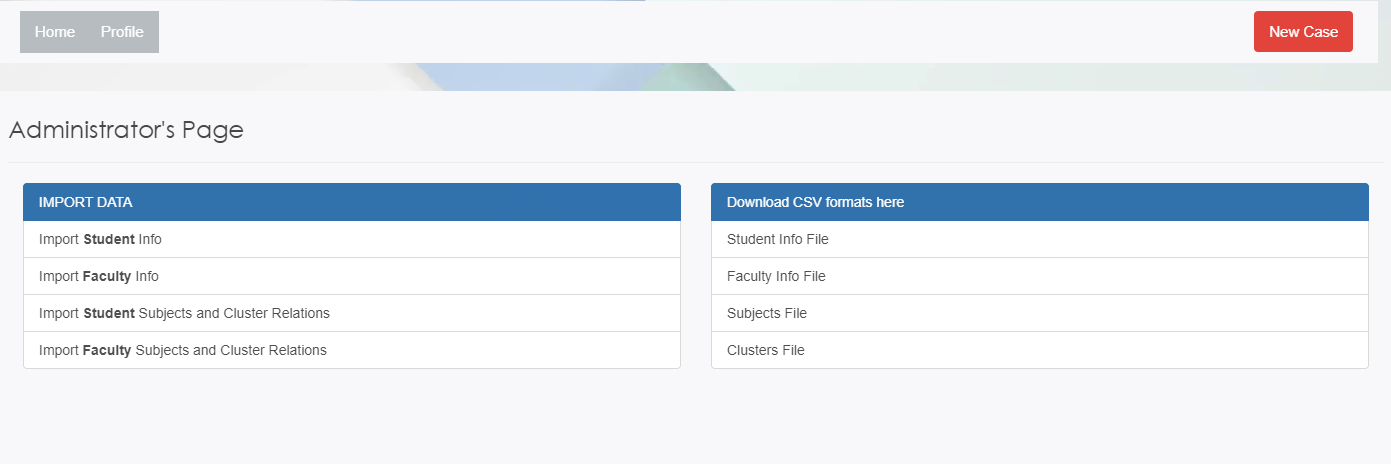


**Figure 5.17 Revised Global Search Bar**

The Global Search Bar in the proposed version is the same with the new version but with the new version having a more simplistic aesthetic. The tables in Figure 5.17 are more pleasing to the eye than the proposed version’s table of cases which is found in Figure 5.16.

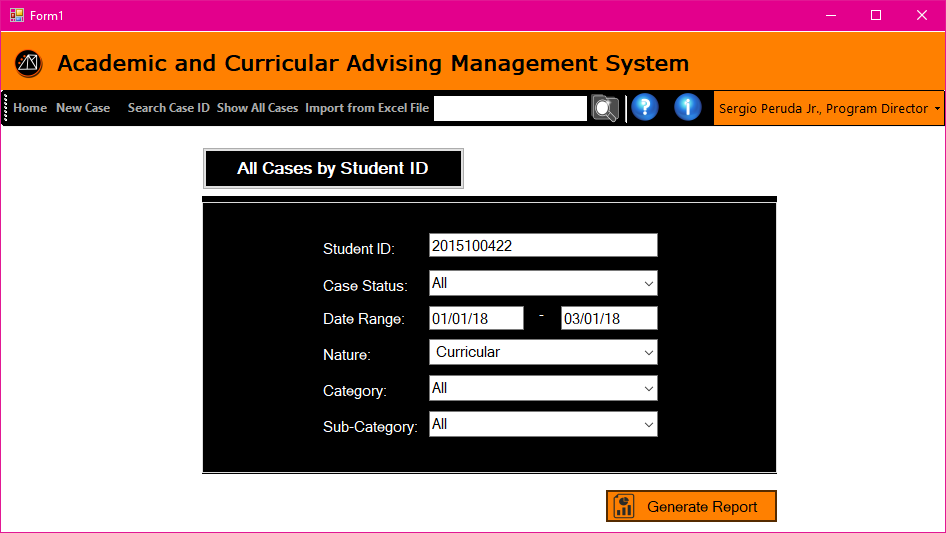


**Figure 5.18 Proposed Import Page**

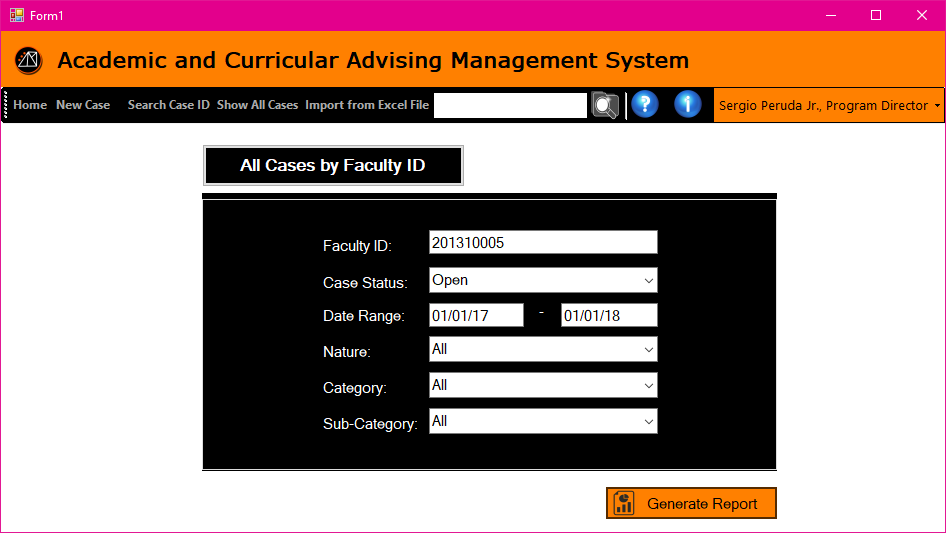


**Figure 5.19 Revised Import Page**

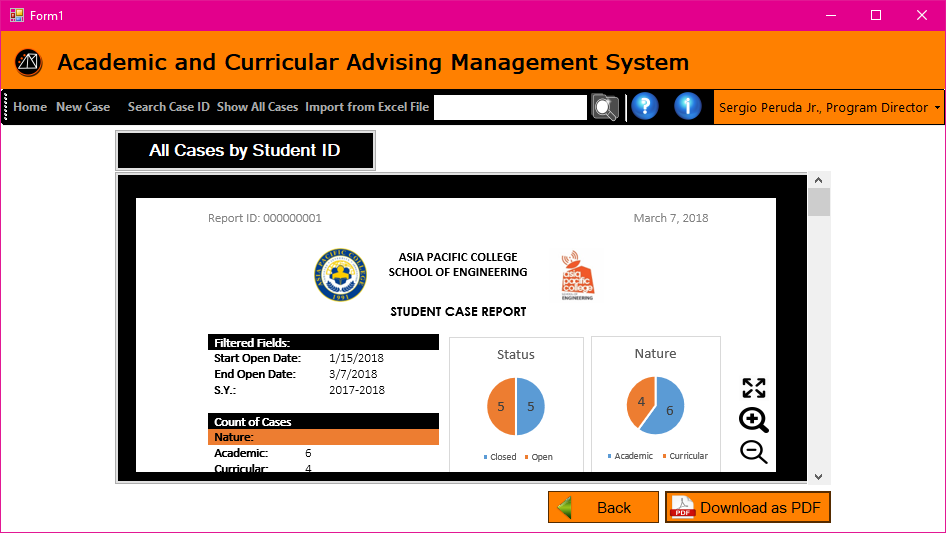
The import pages viewed in Figures 5.18 and 5.19 are relatively far off from each other. The old proposed import page shows a viewable file explorer inside the page and a preview on the side of the file selected. Also the array of mode of import was horizontal and not properly categorized. In the newer revised version, we can observe that the separation of the category whether you want to import or download of the files. Also the menu was removed as shown in the proposed import page. The removal of the file explorer in the internal webpage helped with the simplicity of the coding and the programmer and it will help make the webpage load faster.



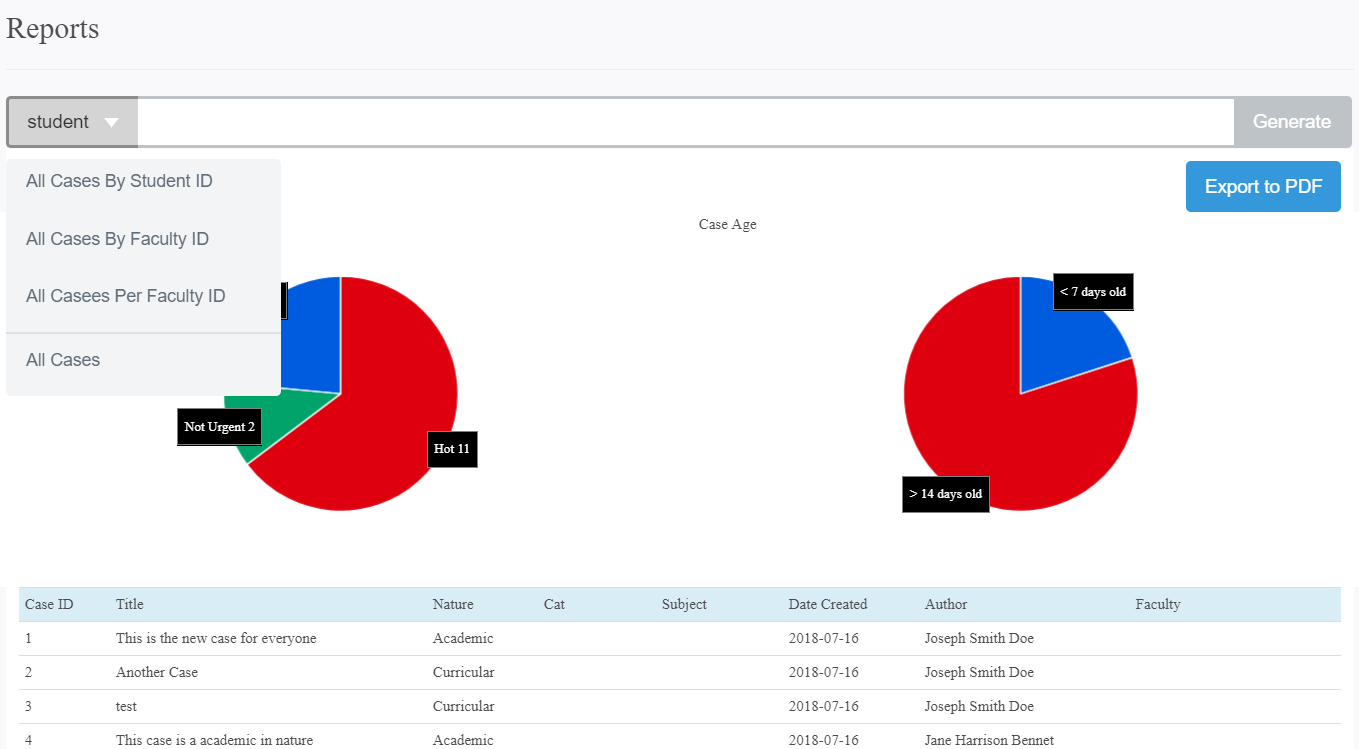
**Figure 5.20 Proposed Reports Page (Student)**



**Figure 5.21 Proposed Reports Page (Faculty)**



**Figure 5.22 Proposed Generated Report Preview**

****

**Figure 5.23 Revised Reports Page**

There are many pages in the reports page that are linked. All are which seen in Figures 5.20, 5.21, and 5.22. We simplified all of those pages into one page viewed in Figure 5.23. We also placed the reports module below the admin or rather the executive and program director privileges. It is observed that the other parameters for input from the proposed reports page are many, and that the revised reports page lacks these input parameters and that you could only input the user ID of the desired choice, during the development phase, we are a bit delayed in the completion of the reports parameter inputs. But rest assured it will be fully implemented within the timeline of the proponent’s development phase and testing phase.

**5.2 System Test**

**Log in Module**

Test Case: Login

**Table 5.1 Test # 1 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Script** | **Output** |
| The user logs in his/her account using their User ID (School ID) and password. | -The user is redirected to their designated homepage.  - If the user’s input is wrong, a message will appear at the bottom of the page stating, “Username and Password is invalid, or you’re not registered.”  -if the user did not enter any login details, a pop up will appear stating, “Please put login details!”  - If one of the two fields are missing, a long message will appear at the bottom of the page from the database. |

Test Case: Forgot Password

**Table 5.2 Test # 2 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| The user clicks the “forgot password” button and answers the security questions | - The user is shown with a popup that will make the enter their security question answers are correct and will be redirected to the enter new password page.  - If the user didn’t enter anything in the textbox, then the page will extend proceeding to the password recover without anything.  - If the user entered the user ID that doesn’t exist in the database, there will be an error message stating, “The student ID is not existing” |

**Registration Module**

Test Case: Self Registration

**Table 5.7 Test # 1 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| Enter user ID | - If the user entered the correct school ID, it will append his or her user account details and is ready for registration.  - When the user inputs a non-existing account, the web page will output an error message stating, “the user ID is not existing” |

Test Case: Link Account

**Table 5.8 Test # 2 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| The user inputs his or her information and click the link button | - If the user information is completed, there will be a message stating, “Successfully link account”  - If one of the fields is missing, there will a pop-up label stating, “Please fill out this field” |

**Case Module**

Test Case: Creating a case

**Table 5.7 Test # 1 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| Any faculty member will click the “create a case” button and will enter the case information along with the student involved. | - A popup notified the faculty member that their case is successfully created.  - If one of the fields is missing, there will be a message stating, “Failed to file a case” |

Test Case: Check if the case created and its information is the same with the corresponding information inputted in it

**Table 5.8 Test # 2 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| Any user, preferably the one who created the case, will view the newly created case and check whether the details one had enter from the test case of creating a case will be correct. He/she can check it by pressing the “view cases” and search for their corresponding cases. | The newly created case showed the correct information inputted by its creator. |

Test Case: Check if the case was created and if the people involved are notified

**Table 5.9 Test # 3 Test Scripts and Expected Output**

|  |  |  |
| --- | --- | --- |
| **Test Instruction** | | **Output** |
| For the people involved in the newly created case, they must check if there is a notification that a case that was newly created with them involved will show. | The notification in their menu page showed a red indicator number 1 because of the newly created case that they are involved with. | |

Test Case: Viewing of the cases selected

**Table 5.10 Test # 4 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| When searching for a case, the user must enter the case details he/she would like to view from the “view case” button and the following criteria/attributes of the case will show, and the user will enter and select their specifications. | The cases with the corresponding attributes selected by the user will be gathered/appended and are shown to the user who called upon the cases. |

Test Case: Adding Case Notes

**Table 5.11 Test # 5 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| In adding case notes, a user must first select the case they want to add case notes on by viewing the case first, then pressing the “add case notes” button for them to create a thread for the case notes. Then the user will enter the notes he/she would like to enter. | - A popup label shows that the case note was successfully added to the case notes thread of that case.  - If the user doesn’t input anything in the textbox, there will be an error message stating “Failed to answer this case” |

Test Case: Check if the case notes are properly created and people who are involved in the case are updated by the latest case notes created

**Table 5.12 Test # 6 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| After creating the newly added case notes, the users involved in the case will be notified that the case notes thread for the case is updated. | Notifications in the user involved have appeared in their corresponding menus. |

**Report Module**

Test Case: Generating a report

**Table 5.13 Test # 1 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| This test is only for the X.D. and P.D. users. They must press the “generate a report” button in their menus and select the attributes that they want to generate in the report. | - A popup notified the X.D./P.D. that their report is successfully generated.  - There will be no cases in the report if the user ID inputted is non-existent. |

Test Case: Check if the correct attributes called by the XD/PD are printed correctly

**Table 5.14 Test # 2 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| The X.D./P.D. can view their newly created reports in an exported file (document/pdf/etc.) | The exported file after the generation of the report has the correct information entered in it. |

**Import Module**

Test Case: Downloading the formats

**Table 5.13 Test # 1 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| Click any of the CSV format. | - The format selected should correspond to the file that has been downloaded. |

Test Case: Import Data

**Table 5.14 Test # 2 Test Scripts and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Output** |
| Click any of the type of data you want to input | - When the user successfully imported a correct CSV file, the system will accept and will output a message stating, “Upload successful”  - When the data uploaded does not meet the format of the fields, it will not accept the data in the database and will show an error message stating, “Invalid Data or Format” |

**CHAPTER VI**

**Summary of Finding, Conclusions, and Recommendations**

**6.1 Summary of Findings**

Earlier on in our Systems Analysis and Design class, we were given the project that should web-based, intranet, and for Asia Pacific College – School of Engineering department. The project was meant to replace the current manual system of the interaction between a student and a faculty member, along with the executive, and program director. A medium in which they could communicate as well as have their communication and or conversation documented and archived automatically given the set period of time. Our system development started out very well planned. At first, we were in doubt because we thought of how big the system would be, but instead we accepted the Advising System since one of the member has an experience on data management from his previous work that we could relate to this project. We also had difficulties on arranging a meeting with our client because of his tight schedule, so we must arrange meeting days before and be on time, that benefits us to have a good time management. First, we planned on naming the project and we arrived with Asia Pacific College-School of Engineering Academic and Curricular Intranet Advising system, or Academic and Curricular Advising Management System (ACAMS). After the approval of the project title we immediately started doing the first chapter of the document. We all did the task that are assigned to us, but we are advised to define what the “case” means in our system and include it in the Definition and Acronyms and we had a several revisions in our scope and delimitations, also with the constraints. After a several revisions in first chapter, we proceed to the second and third chapter simultaneously. We divided the task to each member of the group, and even though we have different tasks, we still help each other by checking each other’s work if there is something we still have to add to that particular part of the documentation. Our group had many revisions in data flow diagram and entity-relationship diagram, to the point that even it is already our summer break we are still revising it to be precise and simplified. On our reference on the past software engineering, we saw that their user interface’s hierarchy is with the screenshot of their proposed graphical user interface, we decided to make it simpler and made a diagram with the flow of the process of the system. The fourth chapter is about the start-up plan that states the estimated costing of the project, staffing, and resource allocation. In fourth chapter, it is also stated there the work plan and test plan. We can say that the test plan part is very tedious in thoroughly checking the scripts and you have to make sure that the proposed user interface is reflected in the test phases. After completing the contents of the document, we also had a difficulty on printing it because the printer that we used is paper jamming and printing the documents slowly.

The defense we had in System Analysis and Design is our first defense. We prepared a handout for the diagrams, graphical user interface and other figures so that the panel can view it easily. We had a smooth presentation and explained the system thoroughly. We and the panel observed that whenever a panel states our mistake we will group ourselves to that specific panel and defend instead of staying in the middle of the room.

After completing all the requirements in System Analysis and Design, we already looked for a programmer that suits our time and someone who can provide our needs on software engineering. On the first programmer we met, he said that the system is too big he will be needing a longer time to finish it because he also has a job and only a part time programmer. On the second programmer, the transaction was smooth and very professional. We felt secure because he provided a contract regarding the system. This programmer we got is very responsive and easy to talk to, so whenever he has a question about the system he will immediately call or send us a message to clarify about the things he doesn’t understand about the system.

Now, in our Software Engineering course we have devised a timeline along with the programmers that will set the deadline of the project or system that will be developed. There were also many various modifications in our systems proposed database. Still they were only trivial changes, but it could prove fatal when not addressed immediately. The type of development executed by the programmers was a student-based project, and not a commercial-type project. It was agreed upon as a student-based project so that we could view the codes for the documentation. The other type of project which the programmers proposed has codes that are confidential and that they could not share it even if they were to build our system. Majority of us also agreed upon the development being based off a template framework website, modified to meet our functions and needs. The system looked very simple at first, but it proved a bit tedious when integrating due to its functions being interconnected with one another, causing a few minor setbacks in our weekly progress report in our current subject which we are required to develop the said software, Software Engineering. We are also lucky that our website is viewable on line through the programmer’s webhost server and not having to setup the website with local host every time there is a major or even a minor modification.

**6.2 Conclusion**

Truly, not every task is easier than the other. Each has its own difficulty regardless of their complexity, or quantity, etc. We did not expect our system to be that tedious in developing. Regardless, we had a good connection with our programmers, and we are very lucky for their frequent compliance. Frequent consultation with your client, and your programmers is key in developing a software or a webpage system.

Time management will be developed within the development of a system. We learned its value during the final week before our defense. In a nick of time, we manage to sort out what needs to be done and executed them with care and consistency. It is better when things are done earlier rather than cramming before the date of submission. The quality of things that are done over a sufficient amount of time along with the refinement through the process will sow good results.

**6.3 Recommendations**

* When designing a web-based system, whether in the graphical, and or menu-driven aspects, select modern designs and trends to follow the generation’s standard designs in terms of the aesthetics, location of the objects like buttons, text boxes, etc. Simplicity can help reduce the transition redundancies. Also, ergonomically placed features can be appealing to most users.
* Normalization of the systems database turned out good, but too much normalization and overthinking of how to normalize a database can be a crucial mistake or redundancy in reduction which may cause the development phase to slow down due to the resolving process of connections as well as functionality of affected and related functions.
* Constant consultation with the client, conversation with the programmer, and meeting with the proponents are keys to develop a successful system.
* Reduction of page redirection could help the efficiency of the loading of the webpages. Like reduction of pages with small content that can be integrated in an existing webpage, or making it a popup, etc.