**CHAPTER 1**

**PROJECT OVERVIEW**

# Introduction

One of the services a faculty provided to the students of Asia Pacific College is consultation. A professor has an allotted consultation hour for students to address their academic, curricular, and extra-curricular needs. If a student would like to consult, they go to the faculty department located in 4th floor in APC and there is a log book where they have to fill in the date, time, purpose, and the name of the professor they would like to meet. There is no strict compliance for students to write on the log book. Sometimes, students meet a professor on an off-consultation hour during their convenient time.

The School of Engineering would like to record the student-professor interaction. They would like to know when the meeting happened, what transpired, and who were present. They would like to make use of the record as a basis for curriculum improvement, help students with their academic and curricular needs, and other management decisions.

School of Engineering has low population. The retention rate is low due to students transferring to other program because of the difficulty level of Engineering and Math subjects. Moreover, majority of students who stay are having difficulty passing the subjects. Some of these students who are “at-risk” of failing the subjects need more intervention from the professor who they enrolled in the past. They would like to know if intervention would help students from failing and to determine the root cause why a student is failing.

Student and faculty interaction doesn’t end with academic. A faculty can also advise a student with their curricular needs such as the subject they have to take. It is a process included in the pre-registration to enrolment. They also want these interactions to be recorded.

Currently, APC-SOE has no centralized and automated system to track the student and faculty interaction. Asia Pacific College – School of Engineering Faculty-Student Academic and Curricular Intranet Advising System will help the School to centralize and automate the advising records, allow faculty to review the content of the past advising sessions, and generate report for the management.

The system is like a ticketing system that will gather, record, and retrieve all transaction and will generate a reference number which is a unique identifier. This identifier will be called a "case". It will contain all information such as the case owner, the student, the case for, nature of concern, date and time of the notes.

The system has six (6) users: Admin, Executive Director, Program Director, Adviser Faculty, and Students.  Table 1.2 and Table 1.3 will show the level of access depending on the user level.

The admin will have capability to create an account and manage their module and people access. They are responsible for verifying the user’s identity before registration.

An Executive Director will be able to do all functions on the system. They have the capability to generate reports available in the system which are printable and in pdf format and import database records.

The Program Director does have the same capability. Reports that will be generated will only contain cases related to the student under his program.

The faculty can create a case, add notes on the case, create additional sub case if needed for other faculty intervention, and close the case. He can only access the students' cases that are currently enrolled to his subjects which include previous cases created by another faculty.

The student can create a case, add notes, and close the case. Only cases associated to his login are accessible.

Asia Pacific College-School of Engineering Student-Faculty Academic and Curricular Intranet Advising System will be a communication system that will gather, save and retrieve all interaction to help student with their academic and curricular needs and help SOE management with decision making.

* 1. **Objectives**
     1. **General Objective**

To design and develop a web-based intranet system for Asia Pacific College – School of Engineering that will serve as a medium of communication for student and faculty and to centralize students’ academic and curricular advising records.

* + 1. **Specific Objectives**
* To develop a User Account Registration Module.
* To develop a Login Module.
* To develop a Case module that allows the user to do the following:
* Open, retrieve, and close a case.
* Generate a case ID as a unique identifier of the interaction.
* Categorize the nature of interaction.
* Add and save a note on a case that is stamped with user ID, date, and time.
* Create a sub-case.
* Assign the other users involved.
* To develop a reporting module that will allow users with director level to generate a report.
* To develop an importing module that will allow users with director level to import files as database records.
  1. **Scope and Delimitation**
     1. **Scope**
* The system will be a web-based application.
* The users are the following:
  + SOE Executive Director
  + SOE Program Director
  + Adviser
  + APC Faculties
  + SOE Students
  + Administrator
* The user access to module is defined by their user role:

**Table 1.1 User Roles and Module**

|  |  |
| --- | --- |
| **USER ROLE** | **MODULE** |
| Executive Director | Login, Case, Report, Import |
| Program Director | Login, Case, Report, Import, Registration |
| Adviser | Login, Case |
| Faculty | Login, Case |
| Student | Login, Case |
| Administrator | Login, Registration |

* The user access to cases is defined by their user role:

**Table 1.2 User Roles and Case Access**

|  |  |
| --- | --- |
| **USER ROLE** | **CASE ACCESS LEVEL** |
| Executive Director | All Cases |
| Program Director | Cases by Students under the Program |
| Adviser | Cases under his group |
| Faculty | Cases associated Students Enrolled to his Subject |
| Student | Cases associated with his User ID |
| Administrator | No Accessible |

* The System will have a registration module allowing the user to enter the login and security information.
* The system will have a login module that will allow the user to enter the login information for verification prior to accessing the account. The login module contains account recovery feature in case password has forgotten.
* The system will have a case module that will record, save, and retrieve all interaction.
* The system will have a reporting module that will generate report in PDF format.  The list of the reports are as follows:
  + Case Status
    - All Cases by Student ID
    - All Cases by Faculty ID
    - All Cases per Student ID
    - All Cases per Faculty ID
  + Case Note Details
* The user needs to register before having an account.
* The user must be an enrolled student and currently employed to APC prior to registration.
* A case will have the following information:

**Table 1.3 Case Attributes and Description**

|  |  |
| --- | --- |
| **CASE ATTRIBUTES** | **DESCRIPTION** |
| Age | An indicator that will provide the number of days and hours from the time it was created. |
| Case Closed | The date and time when the case was closed |
| Case Created | The date and time when the case number was generated |
| Case ID | A unique system generated identifier |
| Case Owner | The user who created the case |
| Category | A dropdown to categorize the type of interaction |
| Cluster | (1.) An identifier for a student which cluster he is included.  (2.) An identifier to a faculty whose cluster he advised |
| Faculty | The other user who will be involved in the case |
| Nature | A drop-down where a user can select if the nature is Academic or Curricular |
| Notes | A section where user can add, save, and review the notes of the users involved. Each added note will be stamped with user ID, date, and time |
| Priority | A dropdown that will allow user to categorize the priority level of the case |
| Status | An indicator if a case is Open, Closed, or Automated Closed |
| Sub-Category | A dropdown to narrow down the category of interaction |
| Subject | A dropdown viewable only if Academic to identify to which subject the case is for |
| Title | A user defined title for the case |

* A case can be described by nature, category, and sub-category.

**Table 1.4 Case Nature, Category and Sub-Category**

|  |  |  |
| --- | --- | --- |
| **Nature** | **Category** | **Sub-Category** |
| Academic | Tutorial | Current Subject |
| Academic | Tutorial | Past Subject |
| Academic | Outcome | Passed |
| Academic | Outcome | Failed |
| Academic | Behaviour | Attendance |
| Academic | Project | Consultation |
| Academic | project | Deliverables |
| Academic | Project | Project Output |
| Academic | Other | Other |
| Curricular | Enrolment | Advising |
| Curricular | Enrolment | Load revision |
| Curricular | Pre-registration | Pre-registration |
| Curricular | Flow Chart | Advising |
| Curricular | Flow Chart | Change |
| Curricular | Other | Other |

* + 1. **Delimitation**
* The system can only be accessed within APC network.
* The system will only be available to APC – SOE students and selected faculties.
* The system will not be integrated to any existing APC system and it will be a standalone application.
* The system administrator account will be setup by the programmer. Login information can be changed by the admin.
* The student ID will be the user ID for students and the Employee ID for the faculty.
* A case cannot be deleted once a case number is generated.
* A case notes cannot be deleted nor modified once saved
* A case will automatically change its status from “Open” to “Automated Close” if no activity after 30 days from the last case notes.
  1. **Assumptions and Constraints**
     1. **Assumptions**
* The system will be hosted by APC-ITRO.
* The user checks the system in a daily basis.
* The user sends new case notification via APC Email in case other party is outside APC.
* The user undergone a system training familiarization and case management training.
* The administrator verifies the identity of the user in the APC System prior to the registration.
* The faculty creates a case per student regarding the outcome of the enrolled students to his subjects before the submission of Final Grades.
* All cases, academic in nature, should be closed before the deadline of submission of Final Grades.
* Cluster List, Offered Subject, Enrolled Student per Subject, and faculty list will be imported by the Executive or Program Director every term to the system’s database.
  + 1. **Constraints**
* The APC-ITRO might not be able to accommodate hosting the system due to limited resources.
  1. **Project Deliverables**

**Table 1.5 Major Milestone its expected start and end date.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Major Milestones | Frequency | Reporting From | Reporting to | Medium | Communication | Start Date | End Date |
| Requirement Analysis | Daily | PM | CLT | Email, Meeting | Review Plan | 1-15-18 | 1-26-18 |
| Business Process Plan | Daily | PM | CLT | Email, Meeting | Review Plan | 2-5-18 | 3-28-18 |
| Business Process Framework | Daily | PM | PD | Email, Meeting | Review Plan | 2-5-18 | 3-28-18 |
| System Development | Weekly | SD | DM | Email, Meeting | Review Status | 4-2-18 | 7-1-18 |
| Content Development Plan | Weekly | CD | PM | Email, Meeting | Review Status | 4-2-18 | 7-12-18 |
| Testing Plan | Daily | Te | DM | Email, Meeting | Test Result | 6-14-18 | 7-2-18 |
| Deployment Plan | Weekly | PM | CLT | Email, Meeting | Progress | 7-12-18 | 7-13-18 |
| Audit | Weekly | DM | PM | Email, Meeting | Audit Result | 2-5-18 | 7-13-18 |

Refer to Table 4.9 for the reporting to/from data definition. CLT stands for Client.

Table 1.5 shows the expected start and end for each major milestone. It is also specified the required personnel to deliver the reports and its medium and to whom it will be submitted.

**Table 1.6 Major phases and the actual activities and its expected start and end date**

|  |  |  |
| --- | --- | --- |
| **Conceptualizing** | **Start Date** | **End Date** |
| Define Problem | 15-Jan-18 | 01-Feb-18 |
| Define Project Objectives | 15-Jan-18 | 02-Feb-18 |
| Determine requirements | 15-Jan-18 | 02-Feb-18 |
| Define Business Process | 15-Jan-18 | 02-Feb-18 |
| Define Project Framework | 15-Jan-18 | 02-Feb-18 |
| Define Project Methodology | 19-Jan-18 | 02-Feb-18 |
| Define Reports requirement definition | 22-Jan-18 | 02-Feb-18 |
| Verification | 07-Feb-18 | 08-Feb-18 |
| Validation | 07-Feb-18 | 08-Feb-18 |
| get sign off | 07-Feb-18 | 08-Feb-18 |
| **Workflow Analysis** | **Start Date** | **End Date** |
| Define CM workflow | 05-Feb-18 | 02-Mar-18 |
| Define User types | 21-Feb-18 | 02-Mar-18 |
| Define Access privileges | 22-Feb-18 | 02-Mar-18 |
| Define Content types | 23-Feb-18 | 02-Mar-18 |
| Analysed project workflow | 27-Feb-18 | 23-Feb-18 |
| Present Project workflow for approval | 26-Mar-18 | 28-Mar-18 |
| Finalize Workflow | 26-Mar-18 | 28-Mar-18 |
| Verification | 26-Mar-18 | 28-Mar-18 |
| Validation | 26-Mar-18 | 28-Mar-18 |
| get sign off | 26-Mar-18 | 28-Mar-18 |
| **Tools Selection** | **Start Date** | **End Date** |
| Identify Tools | 26-Feb-18 | 16-Mar-18 |
| short list appropriate tools | 05-Mar-18 | 23-Mar-18 |
| Evaluate tools | 07-Mar-18 | 27-Mar-18 |
| Test Tools | 12-Mar-18 | 29-Mar-18 |
| Select tool | 29-Mar-18 | 29-Mar-18 |
| Verify tools | 29-Mar-18 | 29-Mar-18 |
| Validation | 29-Mar-18 | 29-Mar-18 |
| get sign off | 30-Mar-18 | 30-Mar-18 |
| **System Development** | **Start Date** | **End Date** |
| Install tools | 02-Apr-18 | 03-Apr-18 |
| Configure tools | 02-Apr-18 | 03-Apr-18 |
| Content Development | 02-Apr-18 | 29-Jun-18 |
| Determine Data Flow Diagram | 02-Apr-18 | 27-Apr-18 |
| Define UML or Use cases | 30-Apr-18 | 08-Nov-08 |
| Program coding | 02-Jun-18 | 29-Jun-18 |
| Initial content build-up | 11-Jun-18 | 12-Jun-18 |
| Conduct Test | 12-Jun-18 | 29-Jun-18 |
| Verification | 29-Jun-18 | 29-Jun-18 |
| Validation | 29-Jun-18 | 29-Jun-18 |
| **Functional Test** | **Start Date** | **End Date** |
| Setup test environment | 14-Jun-18 | 20-Jun-18 |
| Define Test Scenarios | 15-Jun-18 | 20-Jun-18 |
| Create Test Cases | 18-Jun-18 | 20-Jun-18 |
| Conduct Test | 18-Jun-18 | 02-Jul-18 |
| Verification | 02-Jul-18 | 02-Jul-18 |
| Validate | 02-Jul-18 | 02-Jul-18 |
| Release component | 02-Jul-18 | 02-Jul-18 |
| Get Sign off | 02-Jul-18 | 02-Jul-18 |
| **Validation Test** | **Start Date** | **End Date** |
| Setup test environment | 22-Jun | 22-Jun |
| Upload / install released component | 26-Jun | 26-Jun |
| Conduct Validation Test | 26-Jun | 04-Jul |
| Verification | 04-Jul | 04-Jul |
| Validation | 04-Jul | 04-Jul |
| Create development package | 04-Jul | 04-Jul |
| Deploy | 04-Jul | 04-Jul |
| Get Sign off | 04-Jul | 04-Jul |
| **Content Building** | **Start Date** | **End Date** |
| Actual content build-up | 03-Jul-18 | 11-Jul-18 |
| Verifying encoded content | 10-Jul-18 | 11-Jul-18 |
| Verification | 11-Jul-18 | 12-Jul-18 |
| Validate | 11-Jul-18 | 12-Jul-18 |
| Get sign off | 13-Jul-18 | 13-Jul-18 |
| **Training** | **Start Date** | **End Date** |
| Prepare Training materials and Equipment | 12-Jul-18 | 13-Jul-18 |
| Conduct User training/s | 12-Jul-18 | 13-Jul-18 |
| Get Feedback | 12-Jul-18 | 13-Jul-18 |
| Verification | 12-Jul-18 | 13-Jul-18 |
| Validation | 12-Jul-18 | 13-Jul-18 |
| Get sign off | 12-Jul-18 | 13-Jul-18 |
| **Go Live** | **Start Date** | **End Date** |
| Check Website accessibility w/in APC | 12-Jul-18 | 13-Jul-18 |
| Install Compatible Browser, Plugins, Add-ons | 12-Jul-18 | 13-Jul-18 |
| Configure Tools | 12-Jul-18 | 13-Jul-18 |
| Install System Application | 12-Jul-18 | 13-Jul-18 |
| Verify Installation | 12-Jul-18 | 13-Jul-18 |
| Validate | 12-Jul-18 | 13-Jul-18 |
| Go Live | 13-Jul-18 | 13-Jul-18 |
| Get Sign off | 13-Jul-18 | 13-Jul-18 |

Table 1.6 shows the expected start and end date for each actual activity.

* 1. **Definition and Acronyms**

The following table consists of terms used in the Academic and Curricular Advising Management System. The terms consist of technical terms for and non-technical terms for the specific details of our so called “cases”.

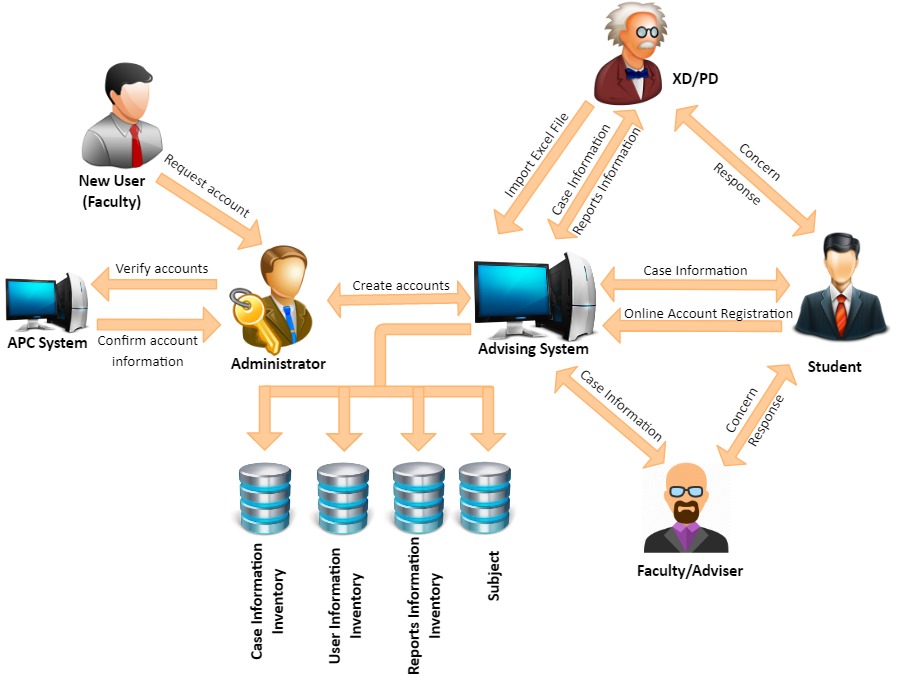
**Table 1.7 Definition and Acronyms**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Academic | The case is Academic in nature if the topic is related to the enrolled subjects |
| At-risk | It is discretion of the faculty based on the running numbers/performance of the student |
| Automate | to decrease the human effort needed to perform a certain task and to make something operate automatically by using machines or computers |
| Automated Close | A category under case status. The Status of the case will be changed from “Open” to “Automated Close” if a case has no activity for 30 days after the last notes. |
| Case | An instance of a particular situation that is related to academic or curricular matters. |
| Centralize | Concentrate (control of an activity or organization) under a single authority. |
| Consultation | The action or process of formally discussing about grades, subjects, or anything about academic or curricular. |
| Curricular | The case is Curricular in nature is the topic is related to activities or school process such as pre-registration, enrolment, load revision, etc. |
| DFD | Data Flow Diagram. This represents the flow of data process and the data entities involved within the system. |
| ERD | Entity Relationship Diagram shows how each entity is connected to another and the tables to be used for the databases. |
| HTTP | Hypertext Transfer Protocol. It is the underlying protocol used by the World Wide Web, and this protocol defines how messages are formatted and transmitted. |
| Interaction | When the student and faculty communicate with each other. |
| Intranet | A local or restricted communications network, especially a private network created using Word Wide Web software. |
| Nature | The choice of the user if the case created is Academic or Curricular matter. |
| Security Confirmation | A confirmation that informs the user if the security questions are already answered or if the user changed his/her password. |
| Security Information | The practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information. |
| Web-Based Application | Any program that is accessed over a network connection using HTTP, rather than existing within a device's memory. |

**CHAPTER II**

**PROJECT ORGANIZATION**

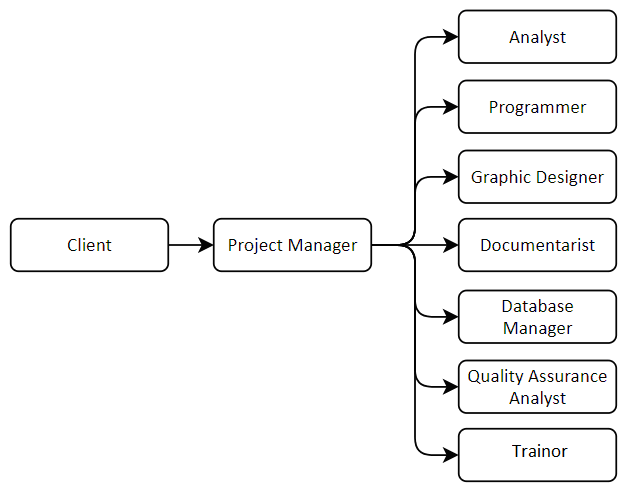
**2.1 External Interfaces**



**Figure 2.1 External Interfaces**

Figure 2.1 shows the relationship of the entities with the system. When a new user needs to register, ne needs to go to the administrator. The administrator will look into the APC system to validate and verify the new user identify. If verified, the new user will proceed with the registration to create an online account. Once done, he is ready to login. ACAMS allows student - faculty interaction to be organized, focused, efficient, and effective. When a user has a concern, regardless of nature, he just needs to find a computer within the APC premises and go to ACAMS website. He just needs to create a case and once a case number is generated and saved, it will be available to the other user involved. Ultimately, the system is a means to support the faculty or student to deal with any interaction in the school, managing the interaction from the moment they’re captured through to their resolution.

**2.2 Internal Structure**



**Figure 2.2 Internal Structure**

Figure 2.2 shows the people involved in planning, assigning, analyzing, and developing the system proposed by the client in a hierarchical model. Each entity corresponds to the type of manpower as well as which phase they are included in the whole phase of the system development.

**2.1 Roles and Responsibilities**

**Table 2.1. Roles and Responsibilities**

|  |  |
| --- | --- |
| **Name of Position** | **Description of Task** |
| Client | The person who is knowledgeable about the system to be created. He serves as the source of information and verification with regard to the project itself. |
| Project Manager | He will manage, plan, monitor, assign task and ensure that the project is properly designed and created to fulfill all the requirements specified by the client. He is responsible for the its successful completion. |
| Analyst | He is responsible for ensuring that the requirements of the client is captured and documented correctly before a solution is developed and implemented for making the project more efficient and effective. |
| Programmer | The one who will develop the analyses and the design approved by the client in a system. |
| Graphic Designer | His role is styling and lay outing of pages with content, including text and images. He will focus on the design process relating to the front-end (client side) design of a website. |
| Database Manager | A specialist that models, designs and creates the databases and tables used by a software solution. This role combines Data Administrator (logical) and DBA (physical). |
| Quality Assurance Analyst | The one who will develop test plans, test cases and test scripts for projects, among other assigned duties. He is in charge of ensuring that the system contains no errors by analyzing development data. |
| Documentarist | The person creates and keeps a copy of all the requirements needed as a proposal for Software Engineering project. He oversees making the minutes of the meeting for every consultation and discussion with the client. |
| Trainor | The person who will create the training module and conduct training sessions that will help the user in understanding how the system works. |

Table 2.1 identifies the personnel involved and their responsibilities in the project making. Their responsibilities describe the nature of work activity and supporting process.

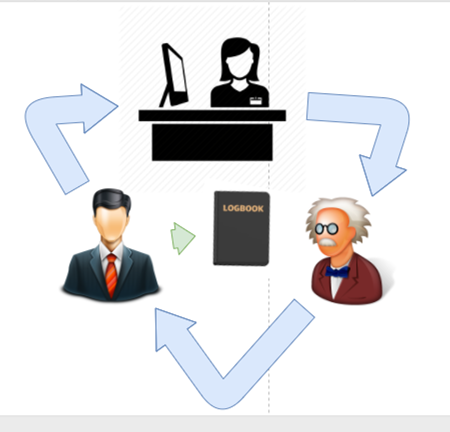
**CHAPTER III**

**TECHNICAL PROCESS PLANS**

**3.1. Functional Description**

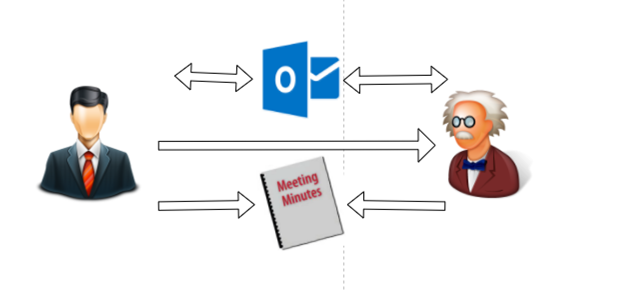
**3.1.1. Current Functional Description**

**CASE 1:** Students go to the faculty department if needed a consultation. A receptionist is available at the lobby. She checks the faculty schedule and their availability for consultation. If the faculty is available, receptionist will call the faculty department to inform the faculty that a student is looking for them. They will meet at the lobby or in a conference room. There is a log book available at the counter where student can write the date, time, purpose, and the faculty name but it is not strictly implemented.

****

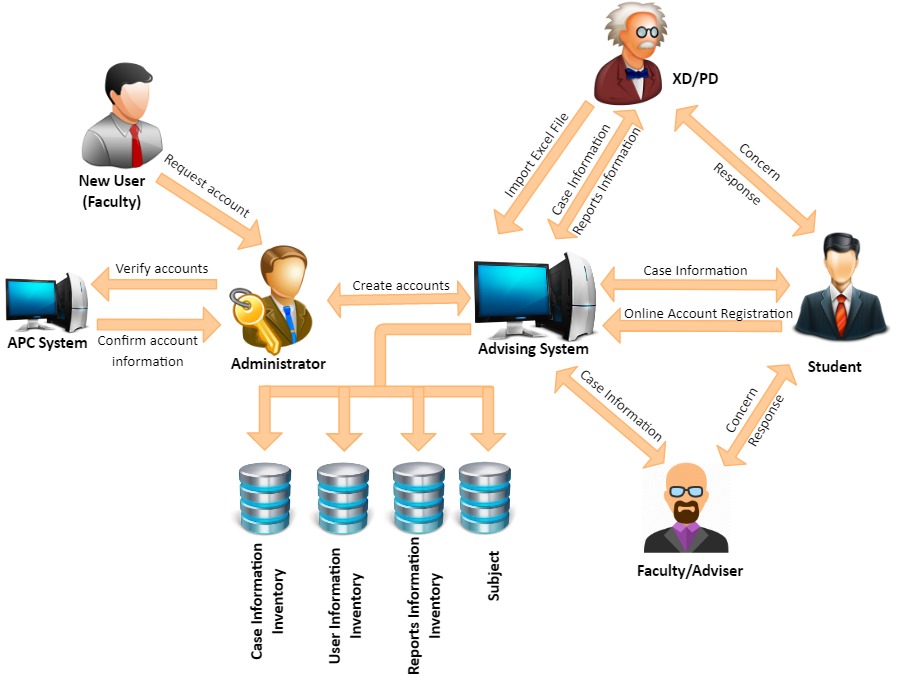
**Figure 3.1 Student consultation Diagram through faculty receptionist**

**CASE2:** The student and faculty meet on their convenient time. The meetings are not documented unless there's a need for a “Minutes of the Meeting” or a meeting invite was sent through outlook.

****

**Figure 3.2 Student Consultation Diagram through Scheduled Meeting**

**3.1.2 Proposed Functional Description**



**Figure 3.3 Proposed Functional Description**

The new design will use automation. It will be a system that will that will create, save, update, and retrieve advising details effectively and efficiently. All interaction will be save in a database, so all the authorized users can access the records.

To make the system accessible, each user will have an online account. The student information will be imported in bulk by directors. Once done, they can proceed with online self-registration where they can verify setup their login and security information. Faculty online account will be setup by admin.

A consultation or advising regardless of concern will be called a case. A case will represent an interaction between a student and a faculty. It will be academic or curricular in nature and the user can further describe the category of concern by the available dropdowns. A case does contain indicators for reporting purposes. Users can exchange notes on a case and the notes are stamped with date and time, user ID. All pending cases will show up on the users’ homepage and those cases that has no activity 30 days after the last added notes will automatically change its status from "Open" to "Automated Close".

The following are the modules available in the system: 1. Registration Module; 2. Login Module; 3. Case Module; 4. Importing Module; and 5. Reporting Module;

The Login Module is the first page that the user can see when they visit the system. This module allows the user to enter the user ID and password. This information is required before accessing the other modules for security verification. The login module also has an account recovery feature. This feature allows the user to enter the answers to the security questions in case the password has forgotten. The security questions were the question selected during the registration.

Registration Module would be the module used to setup the login and security account information. This module would be accessible to student after they received an auto generated email that contains a link. This link directs to the registration module. The users have to fill up the forms to complete the registration process.

The Case Module allows the users to create, save, retrieve and update a case. An interaction is equivalent to a case. It has nature, category, and sub-category that would help users to properly describe the nature of the case. This category selection can be changed by executive director by importing new list. This feature is provided to executive director for system flexibility. A case does contain attributes. A case will have a case ID. It is a unique case indicator that is system generated. It is only shared by a faculty and a student. A case if left unattended or no activity will automatically close by the system. Its status will change from open to automated closed if no activity 30 days after the last added notes.

The Importing Module allows the users with directors’ level to manage database records in bulk.

Importing data:

The executive/ program director has the capability to import files on the system. This data will be used as records on the database. The following are the list of tables that they can import:

-List of Offered Subjects and the faculty that will facilitate (current term)

-List of Student currently enrolled (per subject)

-List of Student Information

-List of Cluster and its respective adviser

-List of Student (per cluster)

Registration Process:

For student - They have to do a self-registration. the student information should be imported prior to the registration. There will be an email that will be sent to the student after the import is done.

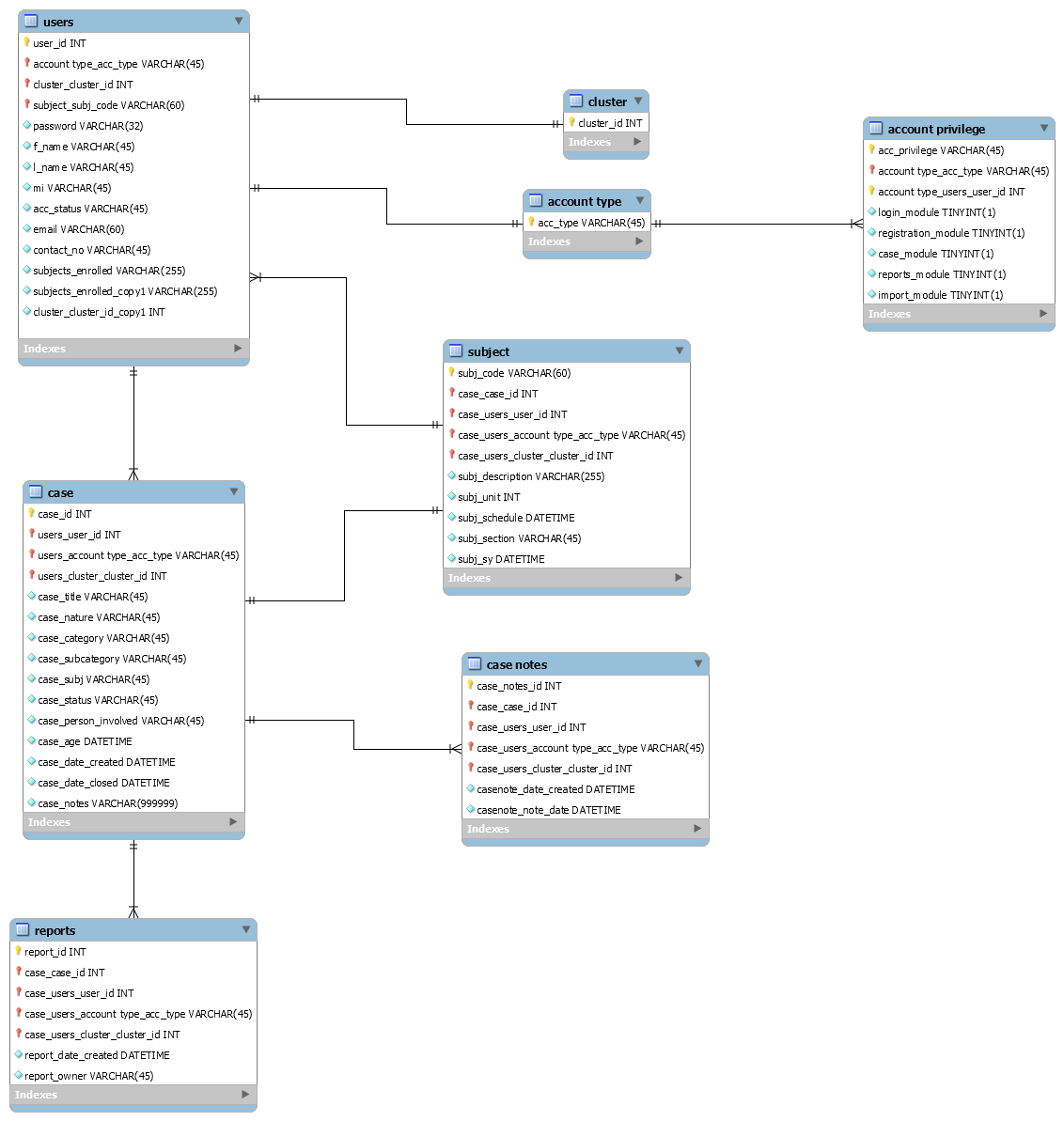
For faculty - They have to coordinate to the administrator to setup their account.

Create/ Managing a case:

The list of all open case will be available on the user home screen right after they login. This is for the user to effectively manage their cases. The data can be sorted depending on the users preferred field. To view the case detail, the user can simply click on the case number and it will be directed to the case detail page. This page contains the student information and the case detail such as the case title, notes, nature, etc. The student and the faculty can exchange notes within the case. It notes will be stamped with their user ID, date, and time. Once the notes is saved, it cannot be deleted nor modified. Cases that has no activity within 30 30 days after the last added noted will be automatically closed by the system by changing the case status from "Open" to "Automated Close".

**3.2. Data Model**

**3.2.1. Entity-Relationship Diagram (ERD)**

****

**Figure 3.4 Entity Relationship Diagram**

In figure 3.4, the Entity Relationship Diagram (ERD) in our system used the crowfoot notation. It is a bit simpler than the chen and a bit cleaner to look at. Our tables composed of primary keys that are interconnected to other tables as foreign keys. There are 14 tables in the diagram. The tables consists of the user, case, case notes, subject, cluster, account privilege, account type, and the six user entities which are the student, faculty, adviser, program director, executive director, and admin.

**3.2.2. Entity Attribute List**

**Table 3.1 Entity Attribute List**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Data Type** | **Required** | **Key** |
| Users | user\_id | INT | YES | PK |
| Users | password | VARCHAR | YES |  |
| Users | f\_name | VARCHAR | YES |  |
| Users | l\_name | VARCHAR | YES |  |
| Users | mi | VARCHAR | YES |  |
| Users | acc\_status | VARCHAR | YES |  |
| Users | acc\_type | VARCHAR | YES |  |
| Users | email | VARCHAR | YES |  |
| Users | contact\_no | VARCHAR | YES |  |
| Users | subject\_enrolled | VARCHAR | YES |  |
| Users | cluster\_id | INT | YES | FK |
| Case | case\_id | INT | YES | PK |
| Case | user\_id | INT | YES | FK |
| Case | case\_title | VARCHAR | YES |  |
| Case | case\_nature | VARCHAR | YES |  |
| Case | case\_category | VARCHAR | YES |  |
| Case | case\_subcategory | VARCHAR | YES |  |
| Case | case\_subj | VARCHAR | YES |  |
| Case | case\_status | VARCHAR | YES |  |
| Case | case\_person\_involved | VARCHAR | YES |  |
| Case | case\_age | DATETIME | YES |  |
| Case | case\_date\_created | DATETIME | YES |  |
| Case | case\_date\_closed | DATETIME | YES |  |
| Case | case\_notes\_id | INT | NO | FK |
| Case Notes | case\_notes\_id | INT | YES | PK |
| Case Notes | user\_id | INT | YES | FK |
| Case Notes | casenote\_date\_created | DATETIME | YES |  |
| Case Notes | casenote\_note\_date | DATETIME | YES |  |
| Subject | subj\_code | VARCHAR | YES | PK |
| Subject | user\_id | INT | YES | FK |
| Subject | subj\_description | VARCHAR | YES |  |
| Subject | subj\_unit | INT | YES |  |
| Subject | subj\_schedule | DATETIME | YES |  |
| Subject | subj\_section | VARCHAR | YES |  |
| Subject | subj\_sy | DATETIME | YES |  |
| Cluster | cluster\_id | INT | YES | PK |
| Cluster | user\_id | INT | YES | FK |
| Reports | report\_id | INT | YES | PK |
| Reports | report\_date\_created | DATETIME | YES |  |
| Reports | report\_owner | VARCHAR | YES |  |
| Reports | user\_id | INT | YES | FK |

Table 3.1 is the Data Dictionary table. This table consist of the tables in the database and its attributes. This table defines the attribute information such as the data type, format, range, if required to have a value and the key type.

**3.3 Process Model**

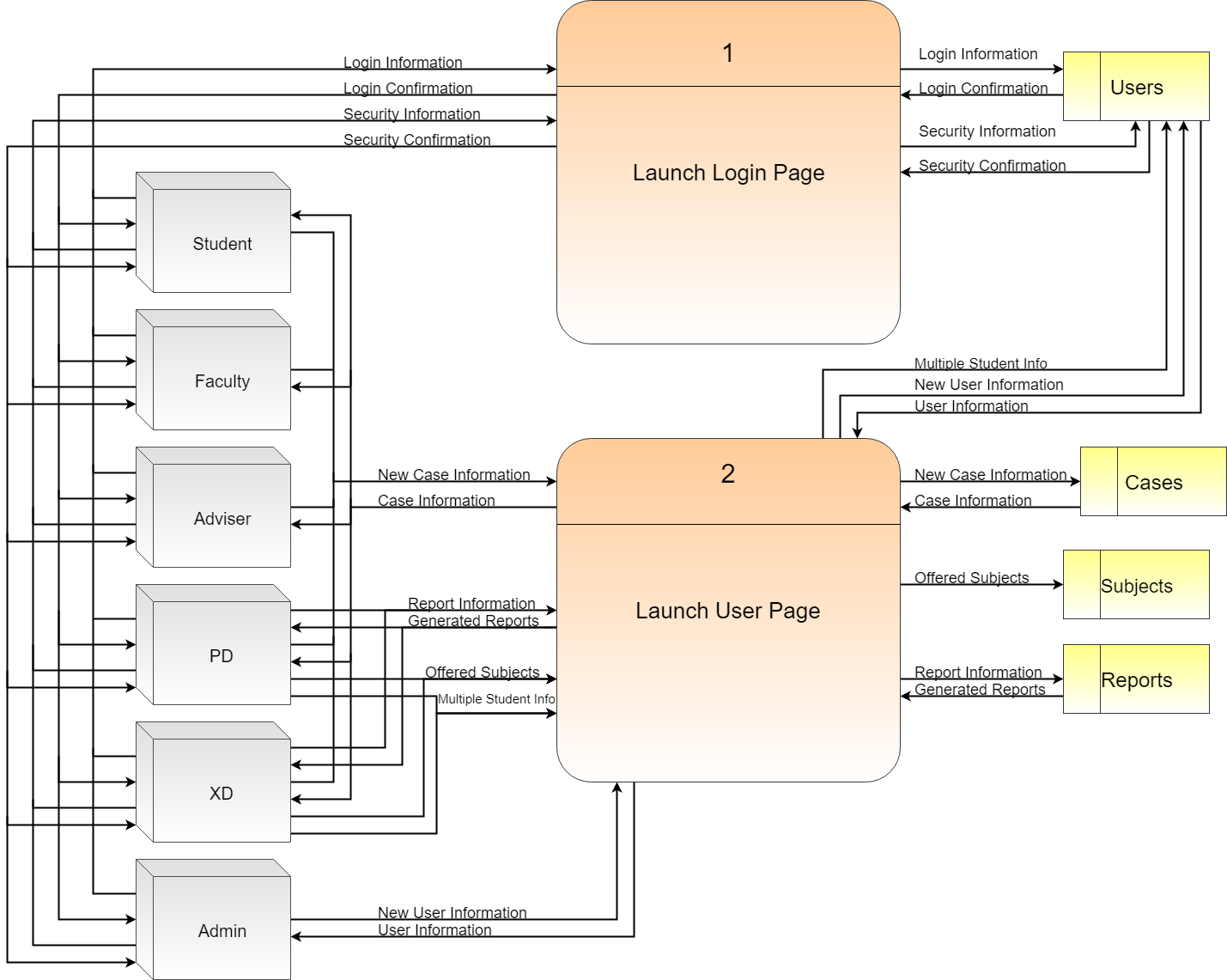
**3.3.1. Project Framework**

In figure 3.5, the context diagram for the Academic and Curricular Advising System contains six entities and four data stores. It includes the inputs from the login, case, reports, and user information. Note that in the users table, there are four inputs and three outputs. One does not go out because it is from the import module which only imports multiple users. The generalization of the arrows is utilized to reduce the redundancy of the input and output data flow.



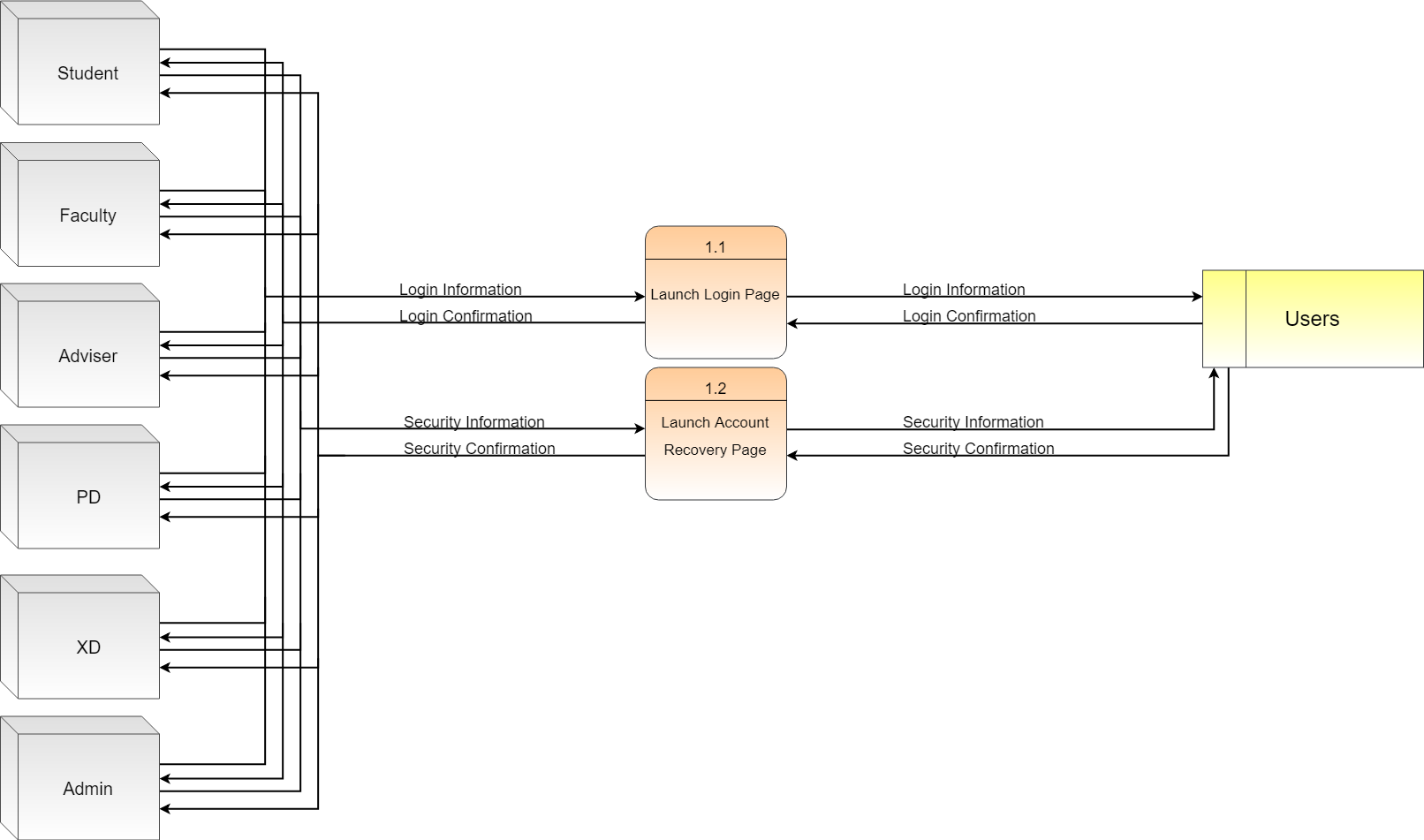
**Figure 3.5 Context Diagram**

**3.3.2. Process Details**



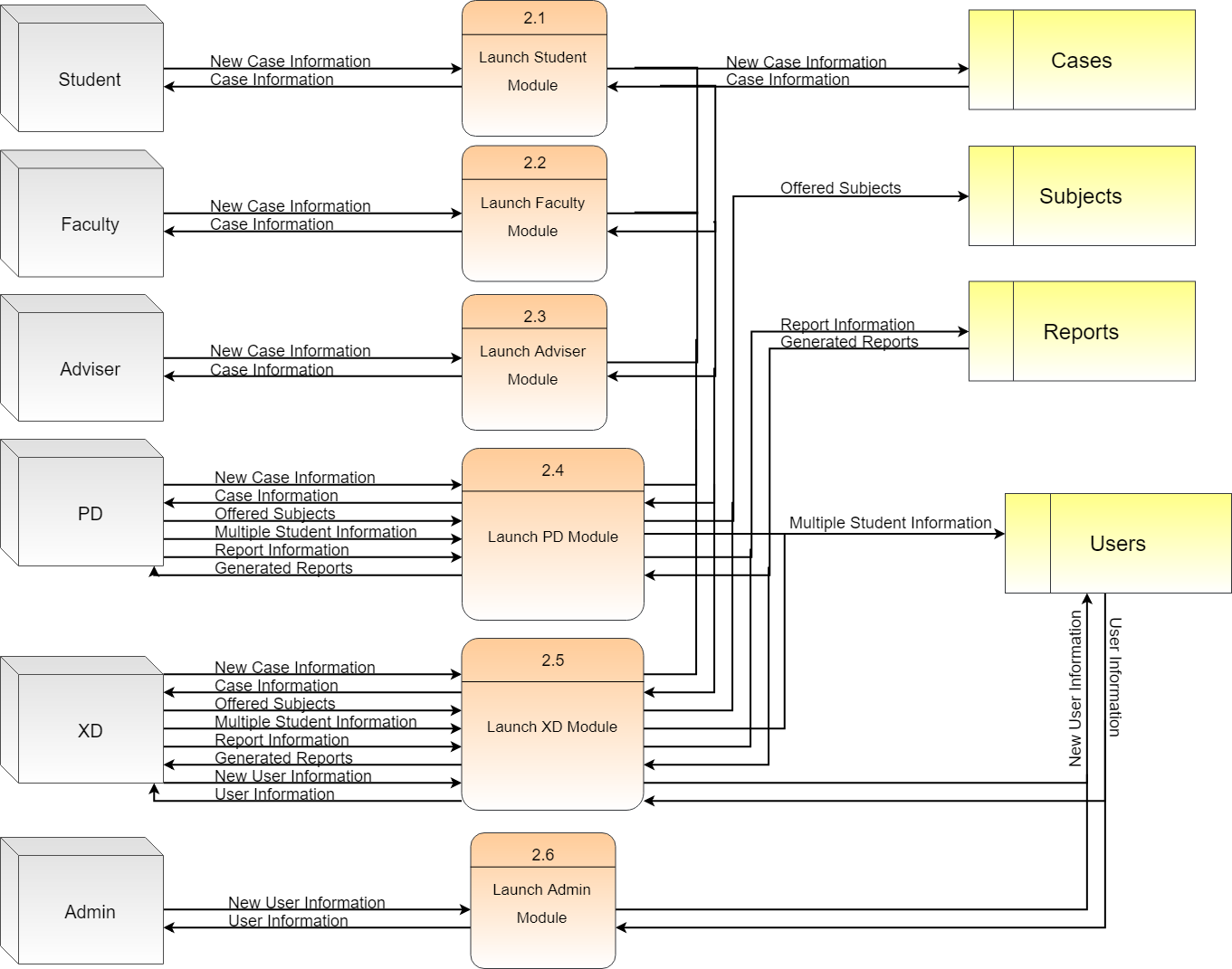
**Figure 3.6 Lvl0 Data Flow Diagram**

The Lvl0 is composed of 2 processes which is the login page and the user pages. The login page is where the users verify and confirm their login information and security details. The user pages are for the user’s corresponding pages with their respective modules.



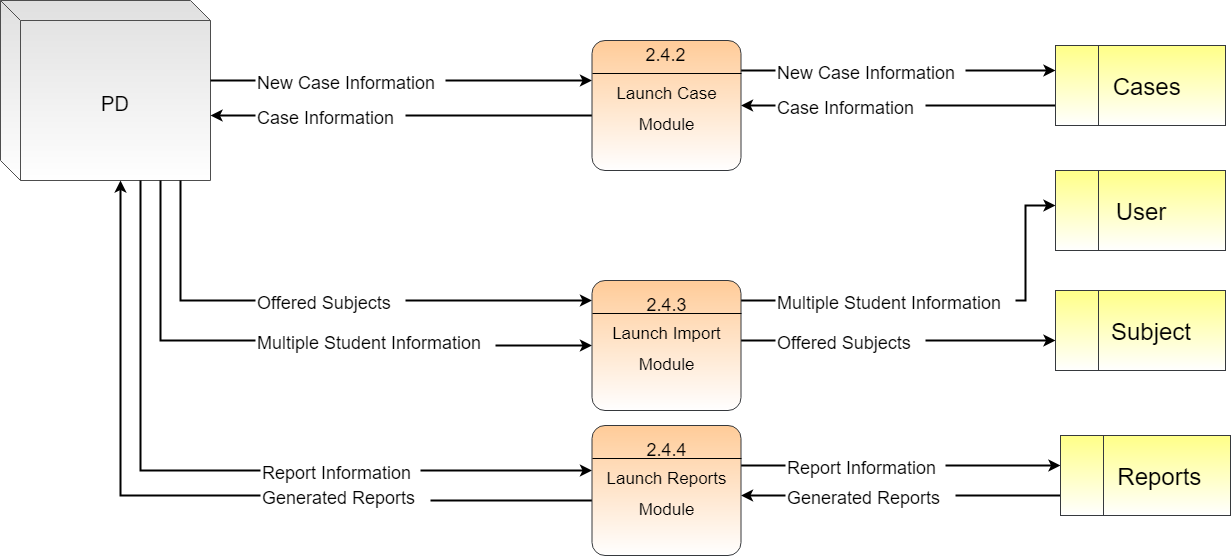
**Figure 3.7 Lvl1: Launch Login Page**

The exploded view of the Lvl1’s login page is shown in Figure 3.7. The login module’s inputs and outputs are generalized for the sake of spacing and redundancy, but in the inputs and outputs.

****

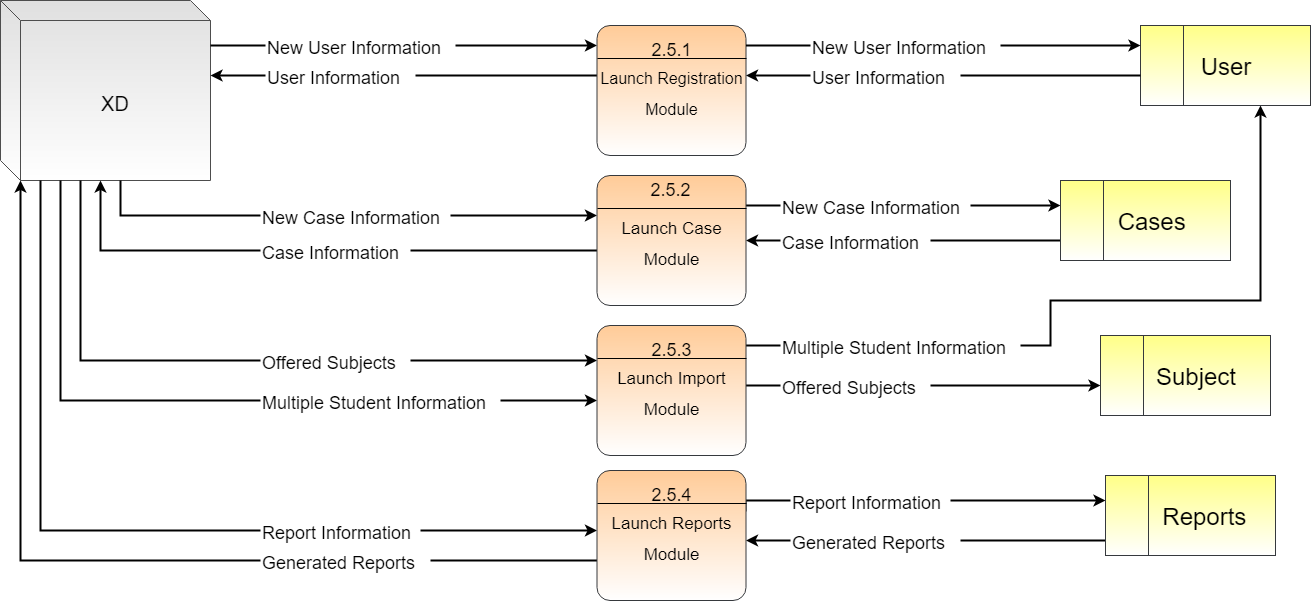
**Figure 3.8 Lvl1: Launch User Page**

The exploded view of the Lvl1’s user page is shown in figure 3.8. Each entity’s individual inputs are specified for the visualization of inputs a user or entity has.



**Figure 3.9 Lvl2 DFD: Program Director (PD)**

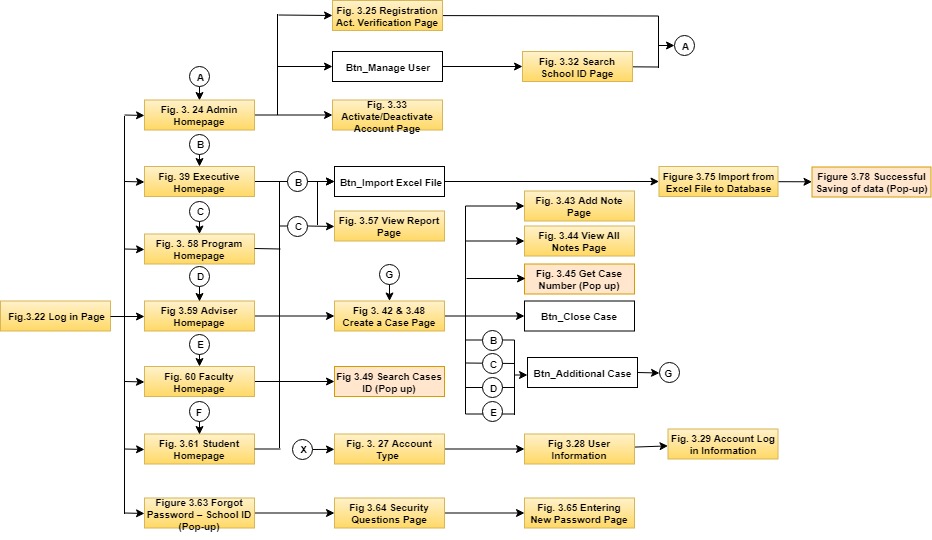
The program director (PD) has three processes: case module, import module and reports module. The case module is similar to the student, faculty, and adviser where they input new case information and could view other case information that are accessible corresponding to their power. Import module is where the PD can import the subjects offered during the term and multiple student information as new users in the users table. A reports module is where only both the PD and XD could utilize. The reports information inputted are the parameters of the reports to be generated and the generated reports could be in a form of pdf or excel file.



**Figure 3.10 Lvl2 DFD: Executive Director (XD)**

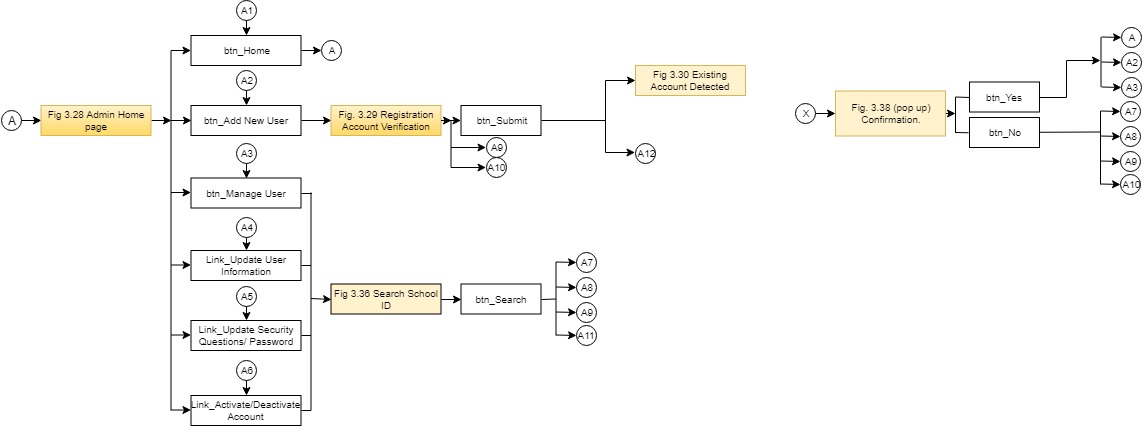
The executive director (XD) has one more process than the PD, which is the registration module where the XD could register new single users. This process is specifically for special case students like transfer students, and for faculty members. Another single process entity is the admin, although it has similar inputs for a process with the XD, the admin’s module has a different level of input like how the admin can edit the details of any user while the XD could only create. As seen in Figure 3.8, the student, faculty, and adviser entities uses the same module which is the case module. The case module is where they input new case information and could view other case information that are accessible corresponding to their power.

**3.4 Hierarchy**

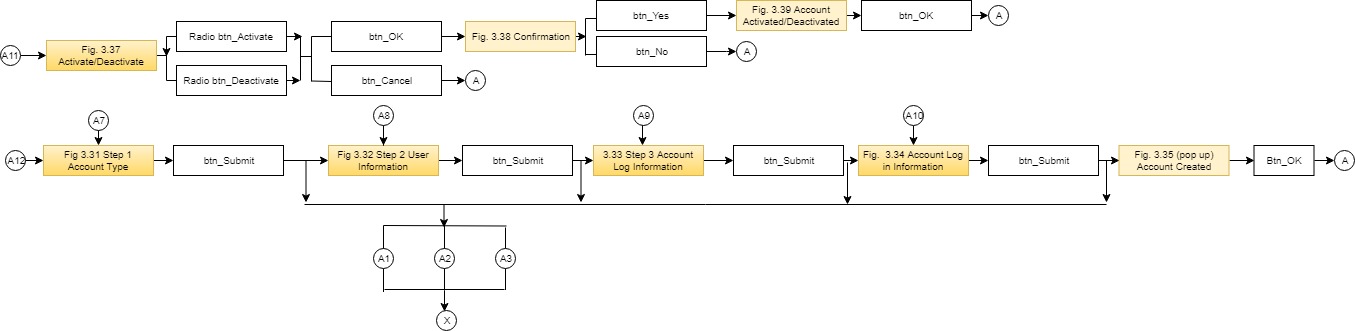


**Figure 3.11 Main Hierarchy**

Figure 3.11 shows the overall flow of the system depending on the position of the user and when the user forgot his password.

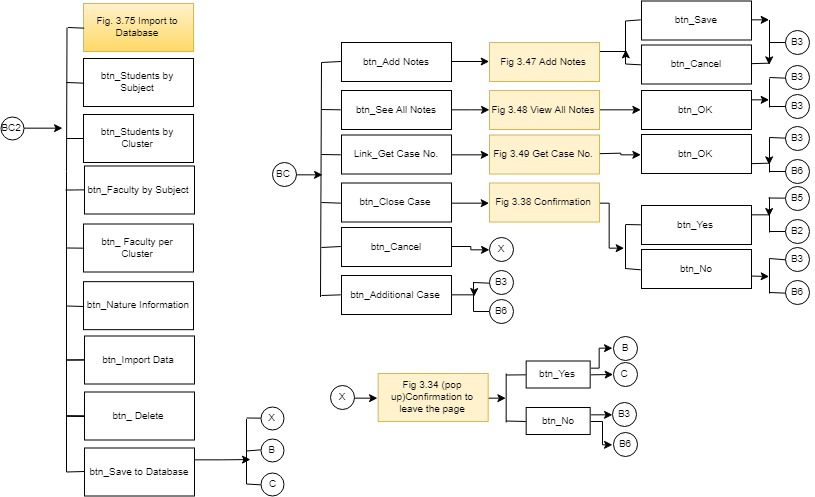


**Figure 3.12 Admin Hierarchy (A)**

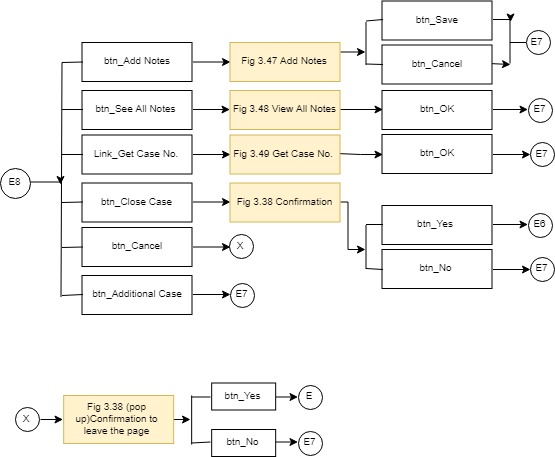


**Figure 3.13 Admin Hierarchy (B)**

Figures 3.12 and 3.13 show the hierarchal processes of the admin user. It contains the elements utilized from the admin menu page as well as the registration page.

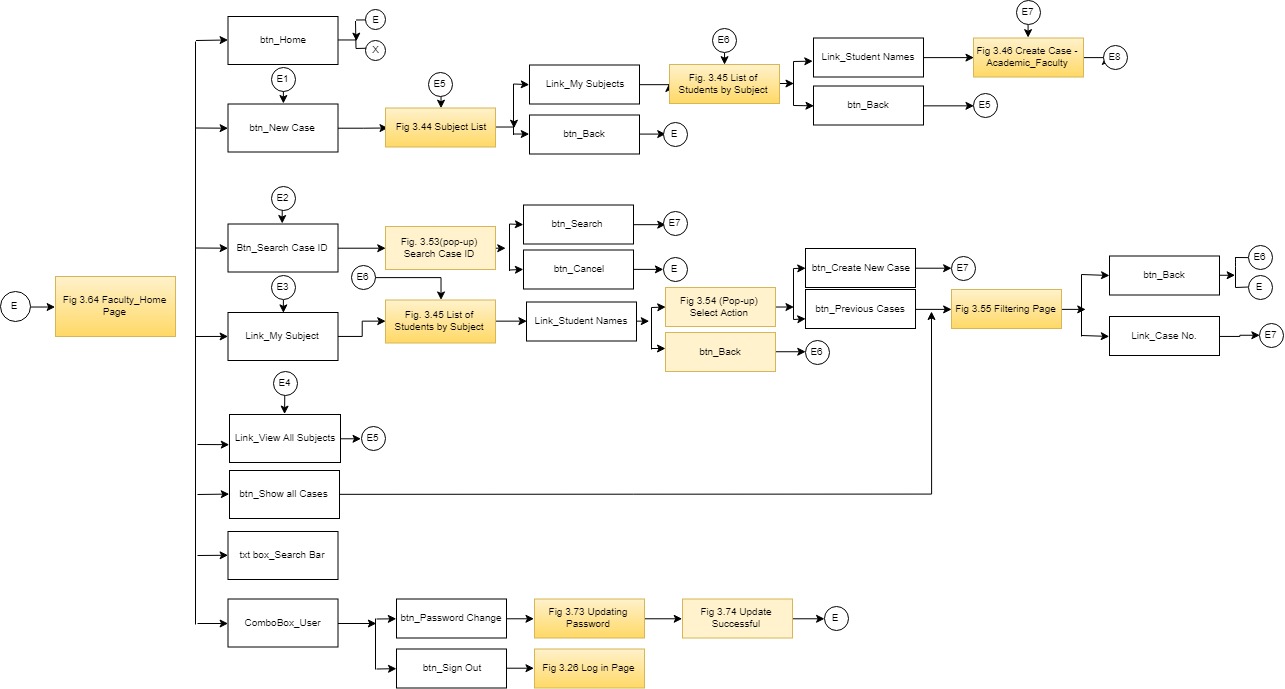


**Figure 3.14 Adviser Hierarchy (A)**

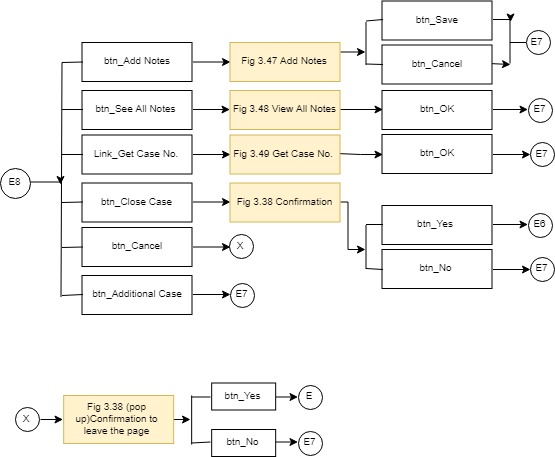


**Figure 3.15 Adviser Hierarchy (B)**

Figures 3.14 & 3.15 show the hierarchal processes of the adviser user. It contains the elements utilized from the adviser menu page as well as the case page.

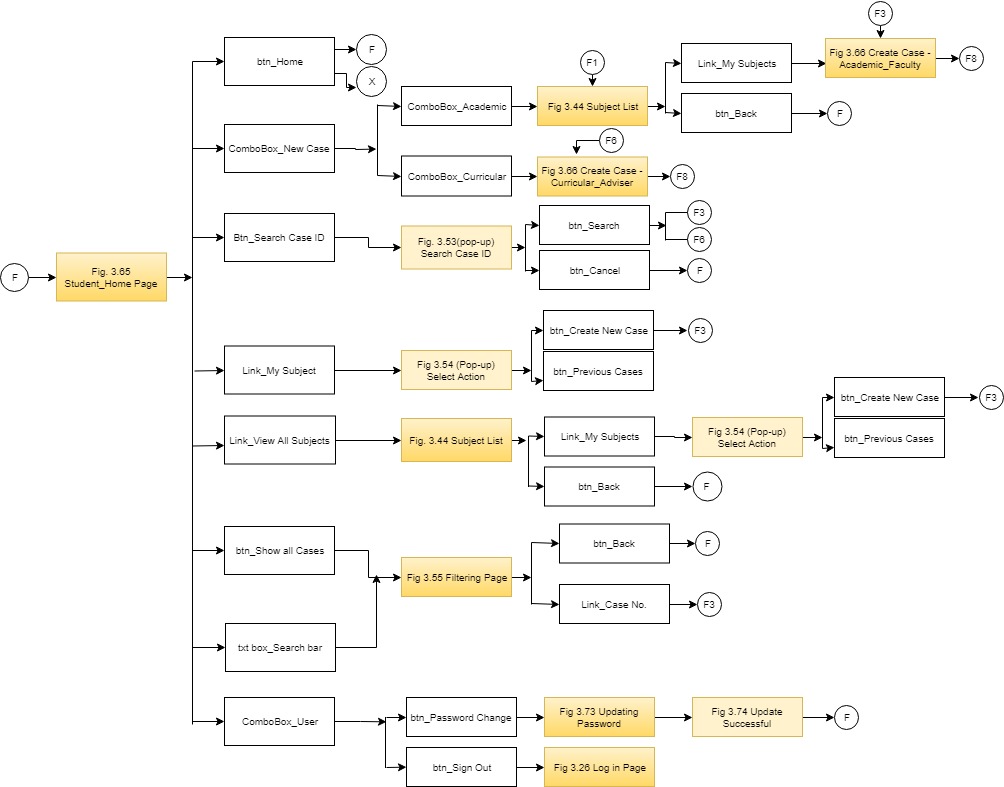


**Figure 3.16 Faculty Hierarchy (A)**

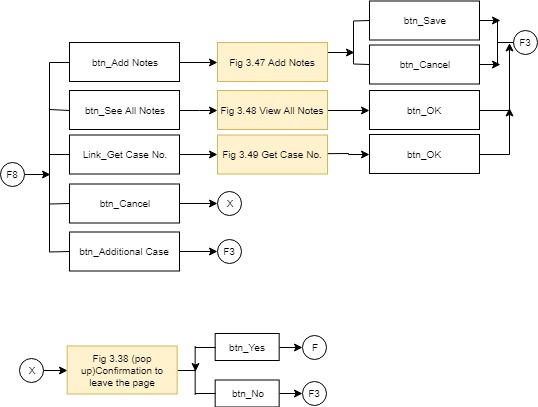


**Figure 3.17 Faculty Hierarchy (B)**

Figures 3.16 and 3.17 show the hierarchal processes of the faculty user. It contains the elements utilized from the faculty menu page as well as the case page.

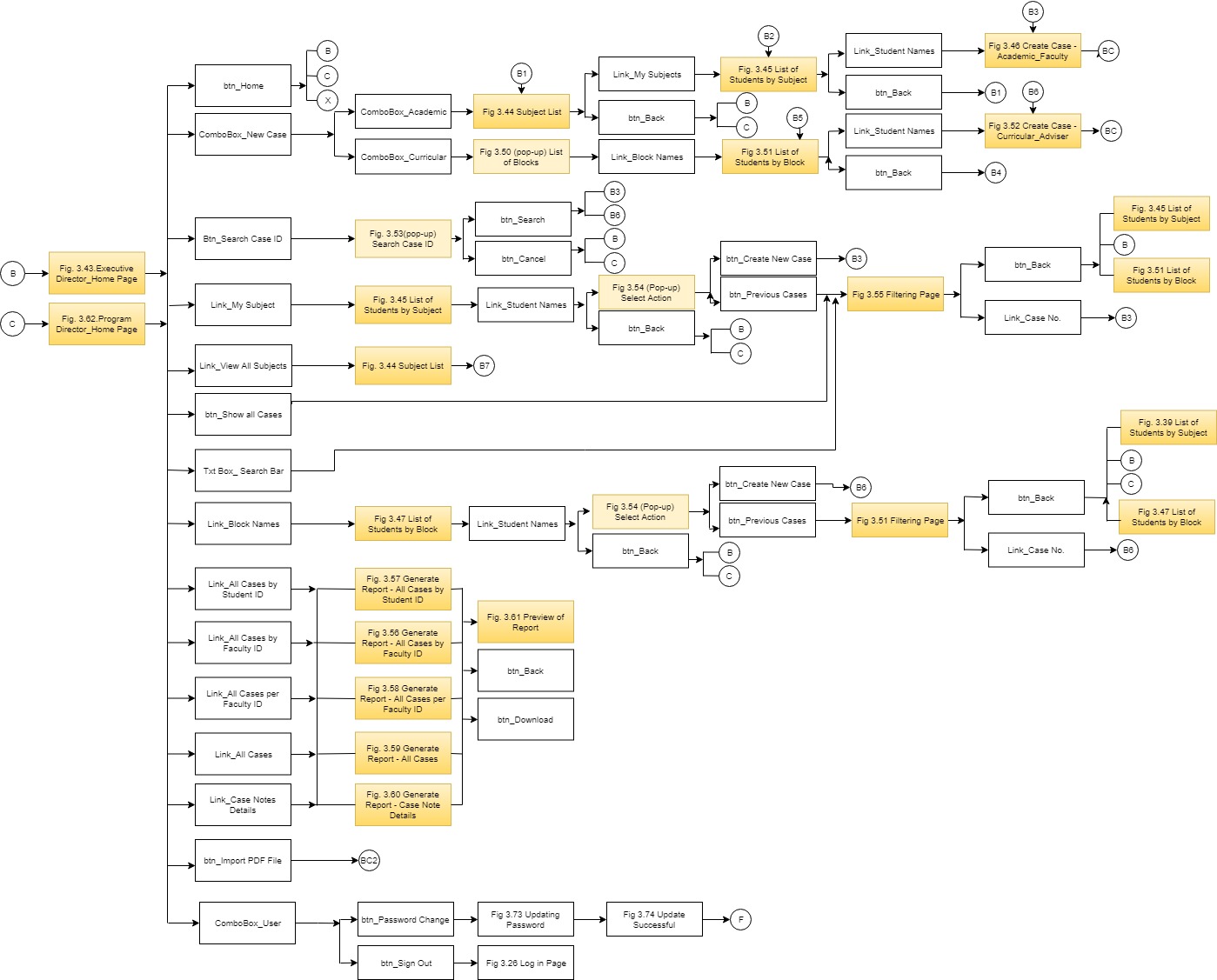


**Figure 3.18 Student Hierarchy (A)**

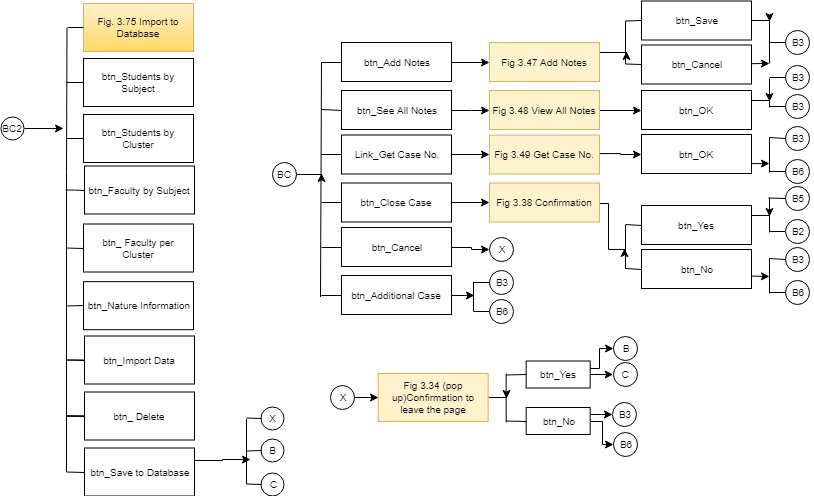


**Figure 3.19 Student Hierarchy (B)**

Figures 3.18 and 3.19 show the hierarchal processes of the student user. It contains the elements utilized from the student menu page as well as the case page.



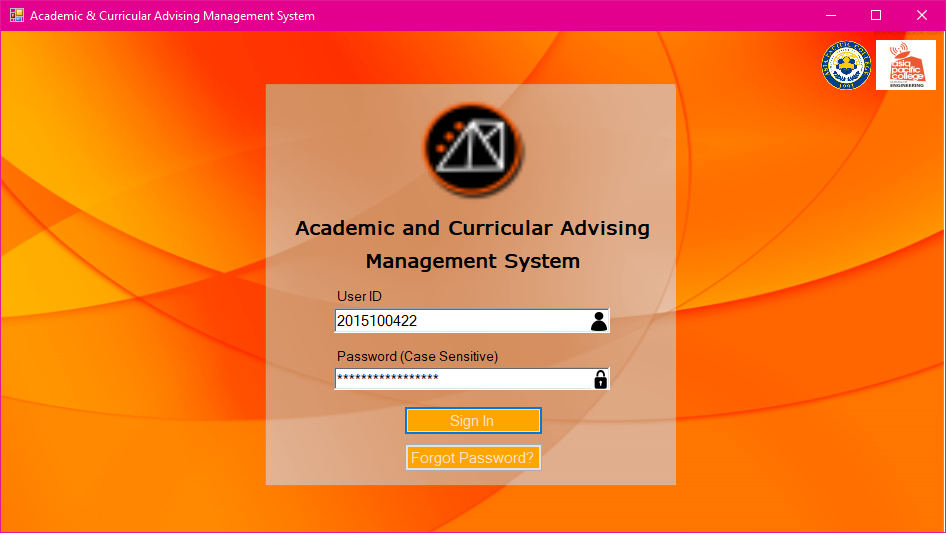
**Figure 3.20 Executive Director and Program Director Hierarchy (A)**



**Figure 3.21 Executive Director and Program Director Hierarchy (B)**

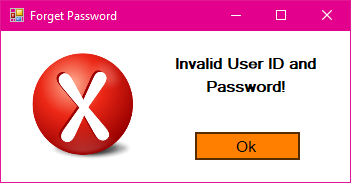
The figure 3.20 and 3.21 show the hierarchal processes of the XD or PD user. It contains the elements utilized from the XD or PD menu page as well as the case page, reports page, and import page.

**3.5 User Interface**



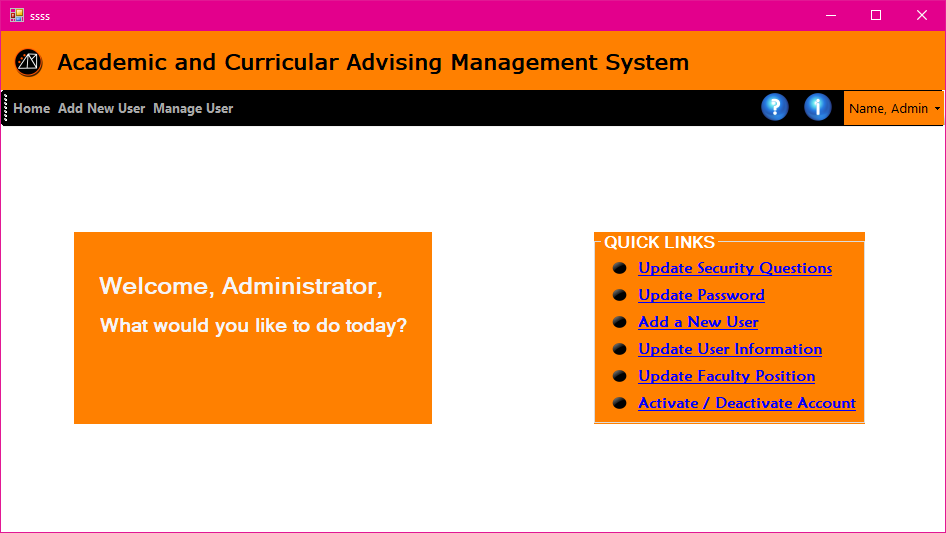
**Figure 3.22 Log in Page**

Figure 3.22 is the user login page. The page allows all users to enter login information such as User ID and Password. The user ID is automatically the school ID number of the user. There is also a hyper link available in case the user has forgot the password.



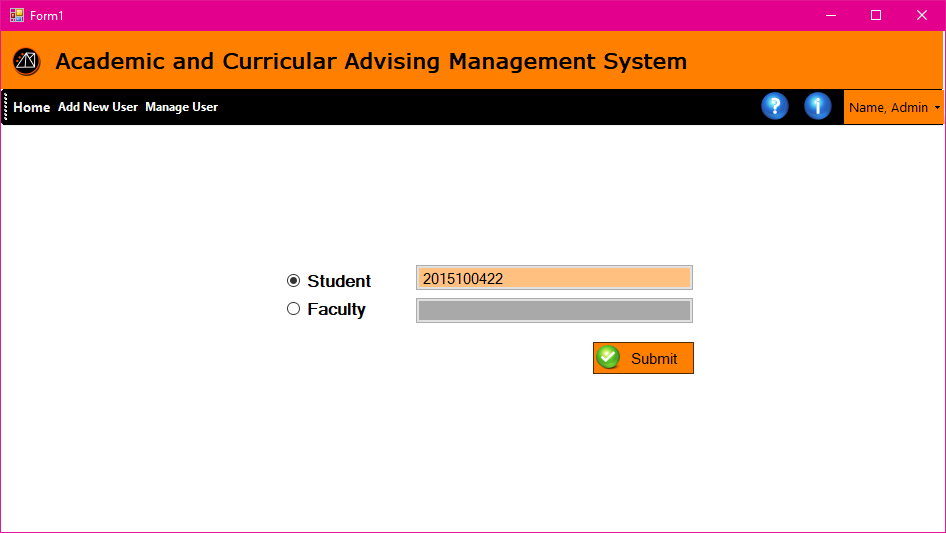
**Figure 3.23 Invalid Log in Information (Pop up)**

If the input in User ID and password is incorrect, this pop-up message will display.



**Figure 3.24 Admin Homepage**

Figure 3.24 is the admin homepage. This is the portal where admin can access the registration module and user information module. It also has quick links that directs to the most visited page.



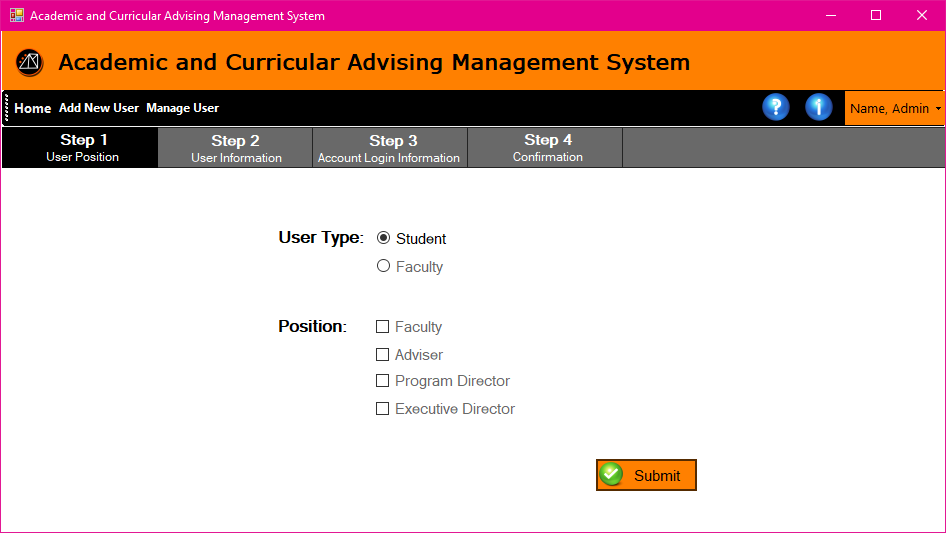
**Figure 3.25 Registration Account Verification**

Registration Account Verification page allows the admin to enter the desired school ID for registration. It has a radio button to identify either the user is a student or a faculty. The system is programmed to search for existing account before proceeding to the registration to avoid multiple account under same ID number.



**Figure 3.26 Existing Account Detected (Pop up)**

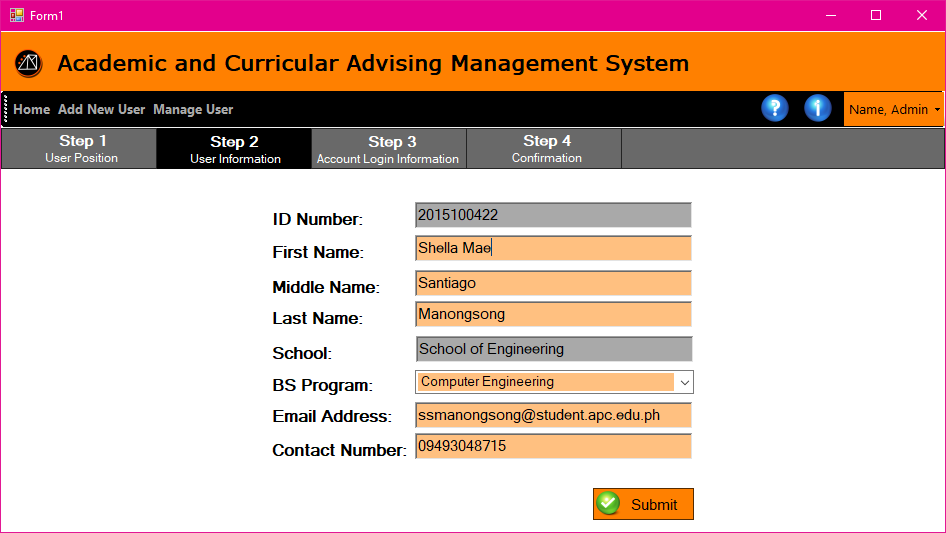
If the account registering is already registered, this pop-up message will display.



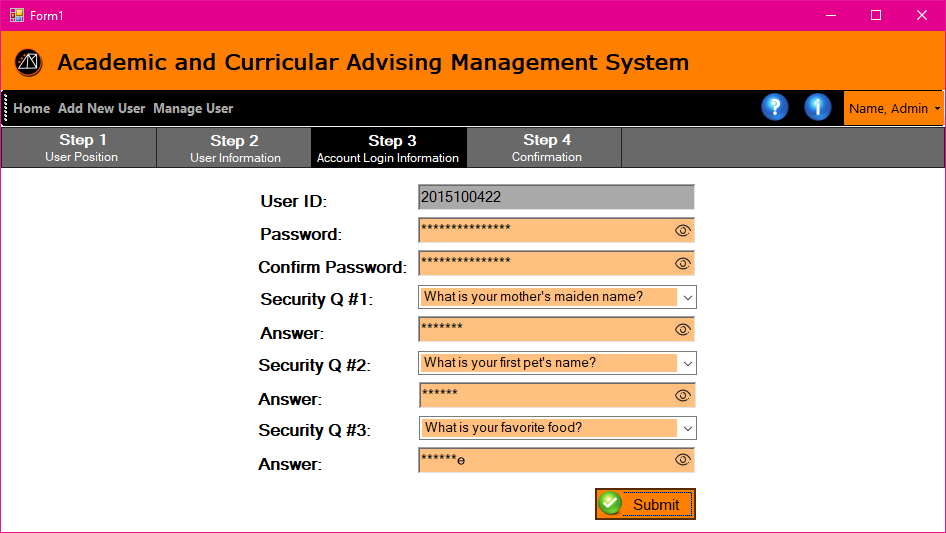
**Figure 3.27 Step 1 Account Type**

Step 1 of registration allows the administrator to select the user type. If student is selected the position will not be available for selection.

In figure 3.28, the Registration page step 2 will allow the user to enter the user information. The ID and the School is not editable from this part of the registration. The rest are user defined information which are required to fill out before proceeding to the next steps

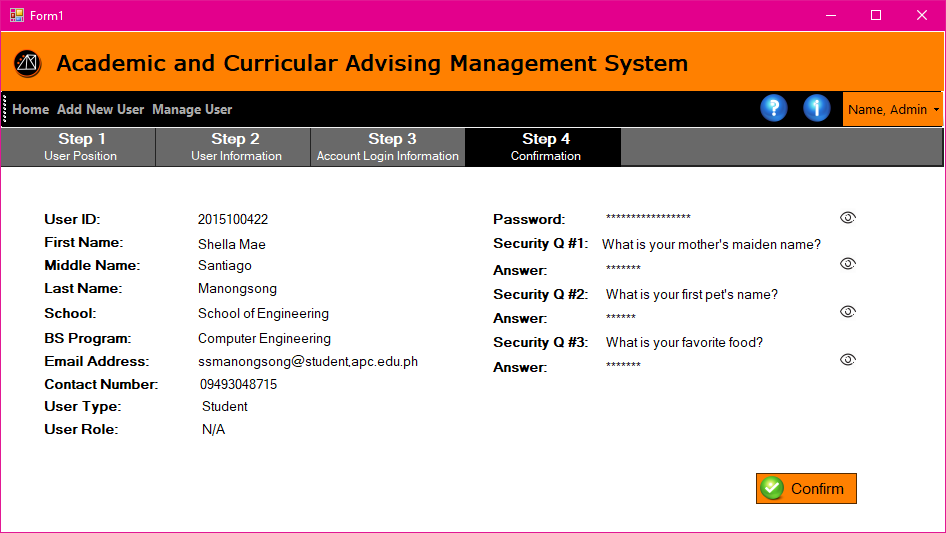


**Figure 3.28 Step 2 User Information**



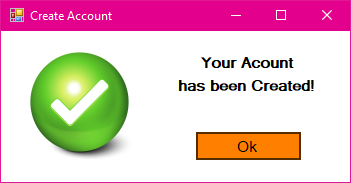
**Figure 3.29 Step 3 Account Log in Information**

Registration page step 3 – Account login information allows the user to setup his password and reconfirm it. There is also an area to setup for 3 security question and password. This information will be used in resetting password in case forgotten.



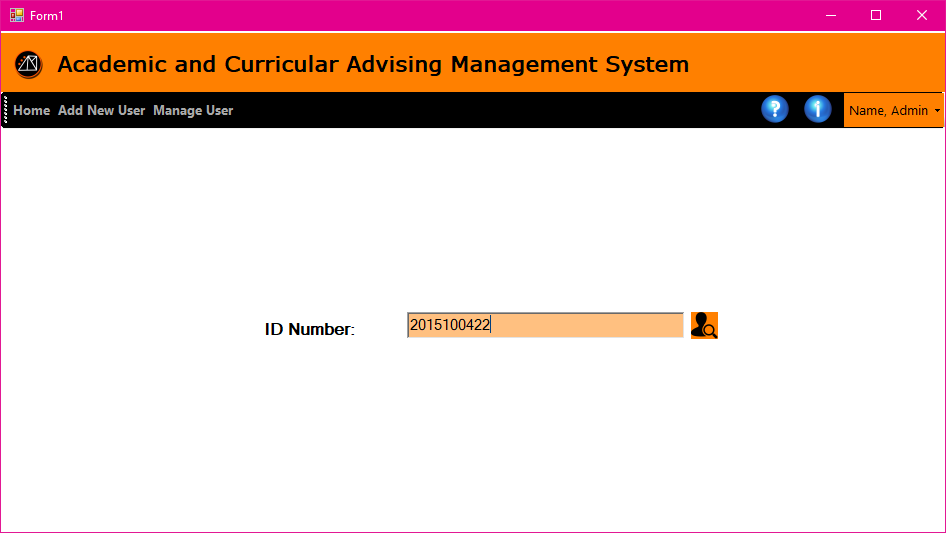
**Figure 3.30 Step 4 Account Log in Information**

Registration Page Step 4 – Account Login Information page allows the user to review all the user's information before completing the registration.



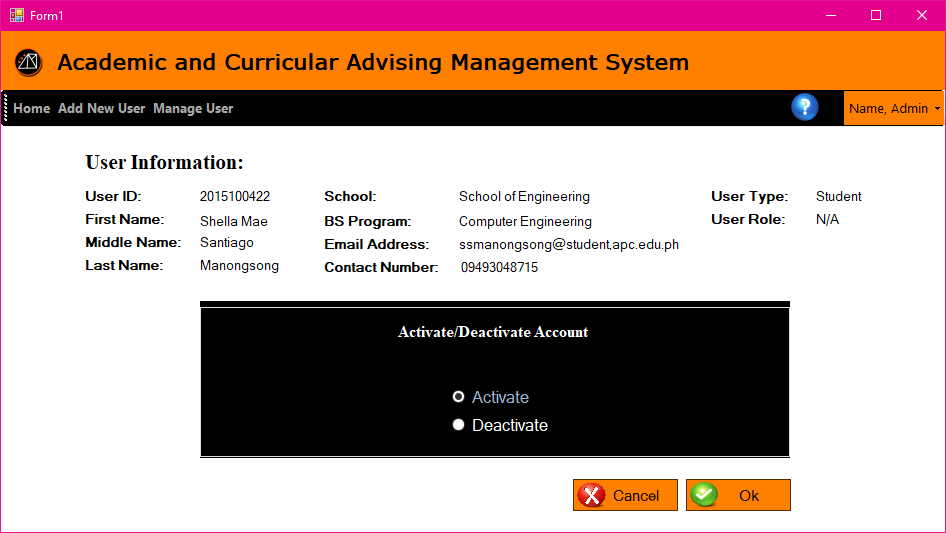
**Figure 3.31 Account Created (Pop up)**

When the user is finished registering, this pop-up message will display indicating the account is created successfully.



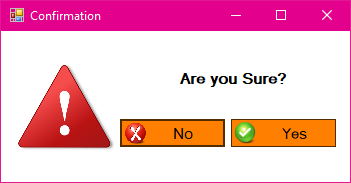
**Figure 3.32 Search School ID**

The Search School ID page allows the admin user to enter a user ID to search for a specific student.



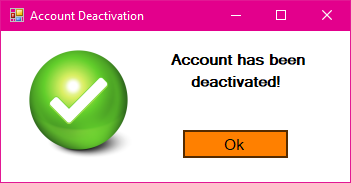
**Figure 3.33 Activate/Deactivate Account**

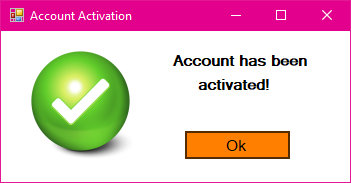
The Activate/Deactivate account Page allows the administrator to change the account status.



**Figure 3.34 Confirmation (Pop up)**

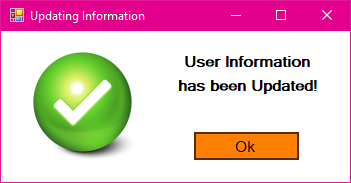
This page is to confirm if the user is certain before proceeding on the next step.





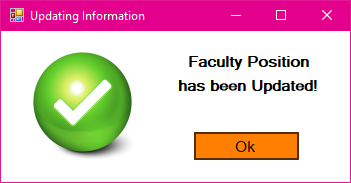
**Figure 3.35 Account activated/deactivated Confirmation (Pop up)**

This confirmation screen is to notify the admin that the request in changing the account status has successfully been processed.



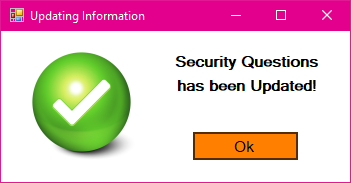
**Figure 3.36 User Information Updated (Pop up)**

This popup will notify the admin that the user information edited has been updated.



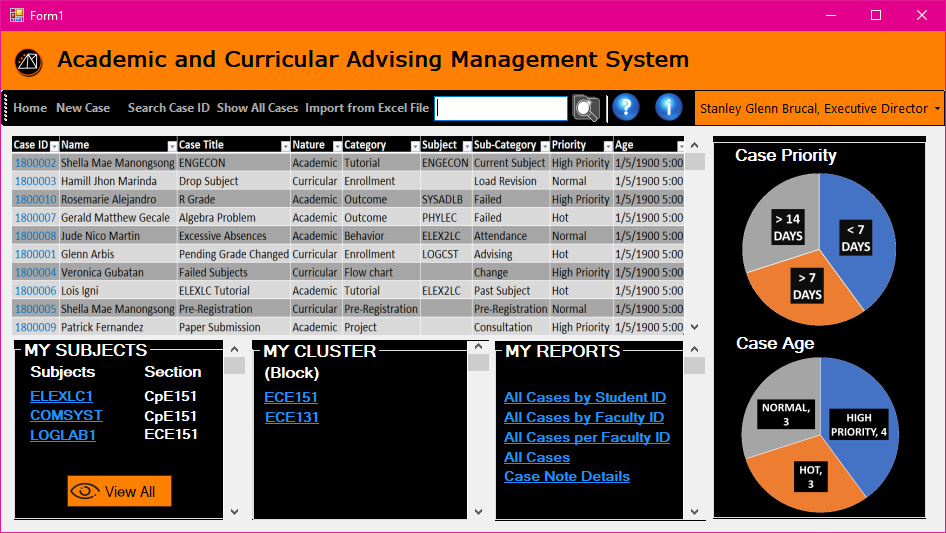
**Figure 3.37 Faculty Position Updated (Pop up)**

After editing a faculty user’s position or account type with its privileges, this popup will notify the user that the position has been updated.



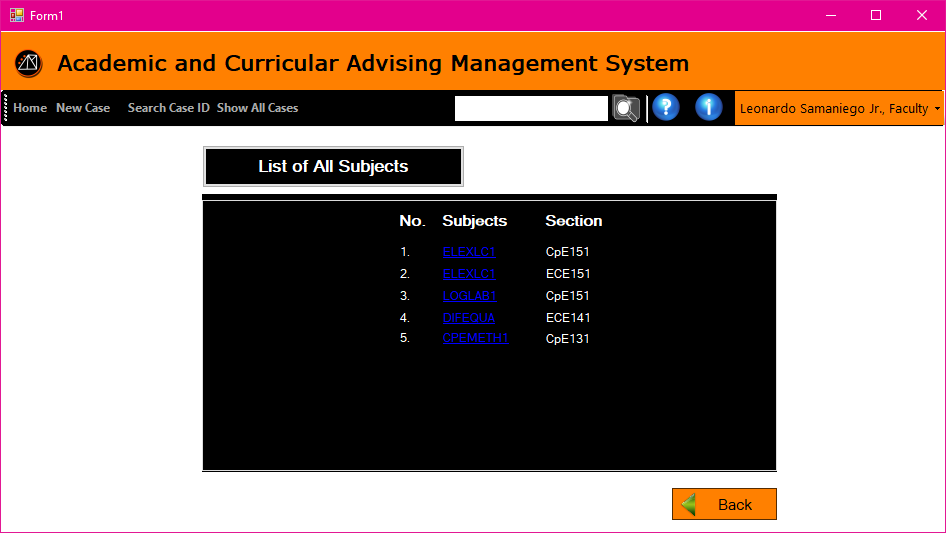
**Figure 3.38 Security Questions Updated (Pop up)**

After editing the security questions this pop-up message will display indicating the change is successful.



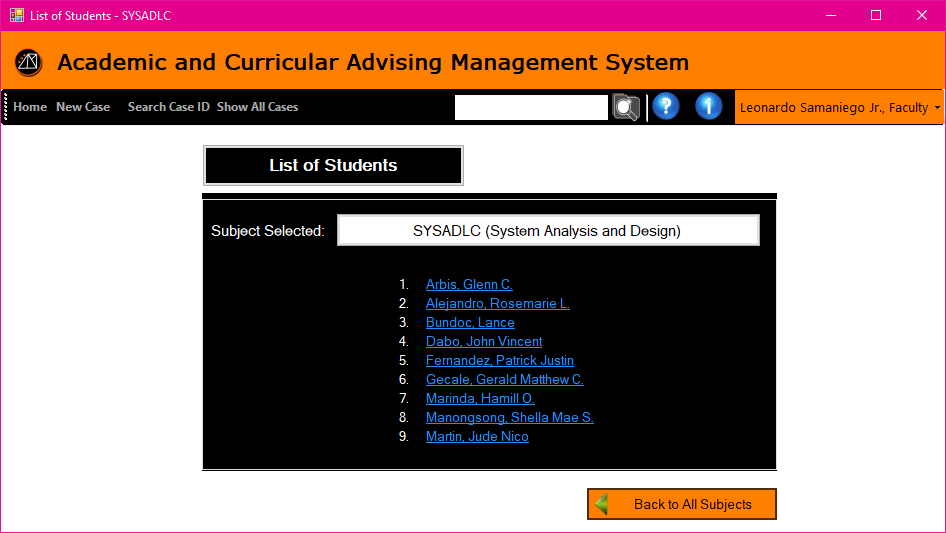
**Figure 3.39 Executive Director Homepage**

The form in figure 3.39 displays the home page of the Executive Director. This is where the summarized table of all cases is shown, and the pie chart of case priorities and its age. There is also a list of the subjects that the Executive Director is handling. There is also a cluster that has a list of all engineering students. And the reports have four types. And there is a toolbar for returning to home page, creating new case, search case ID, and showing all the cases created.



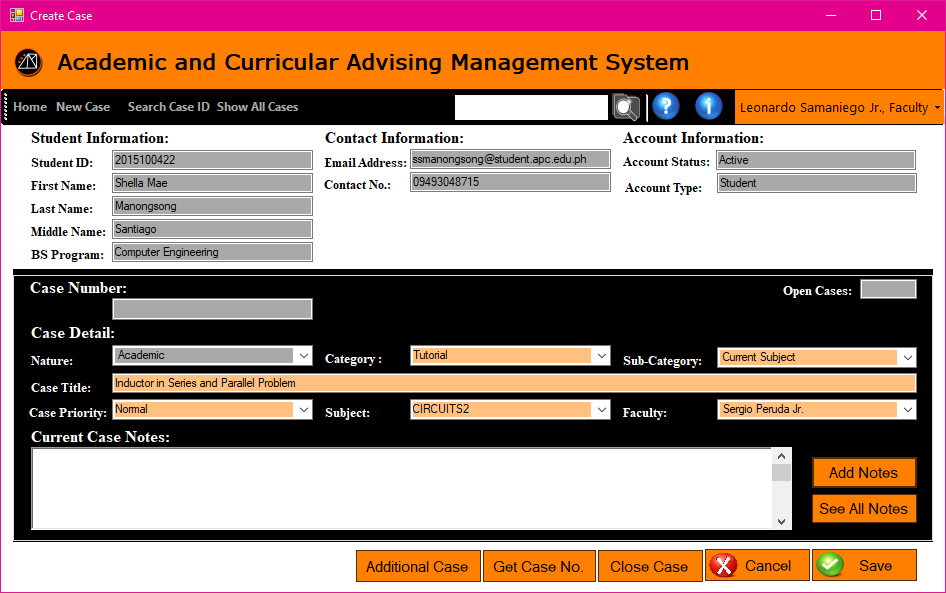
**Figure 3.40 Subject List**

The screen allows the user to view the list of subjects that he handles before proceeding to the next page which is the list of students enrolled in the subject.



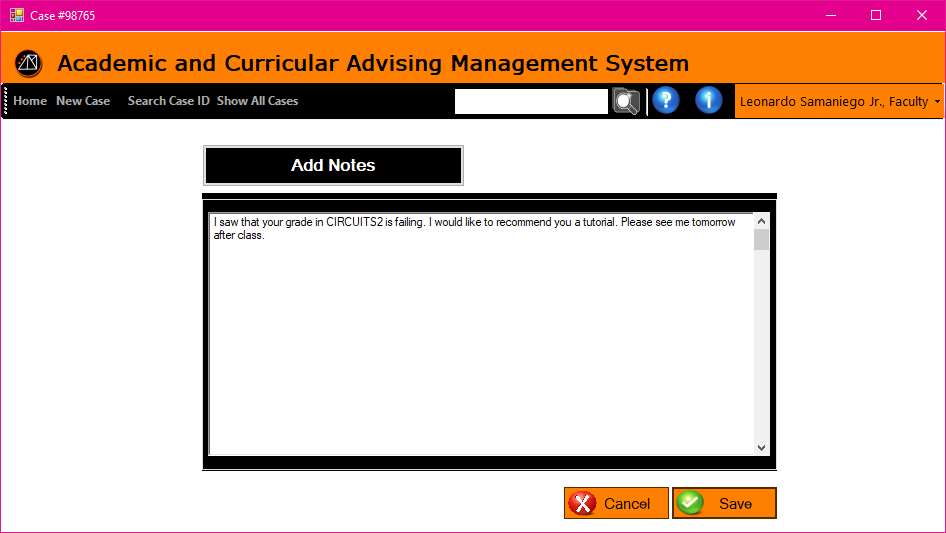
**Figure 3.41 List of Students by Subject**

This page allows the user to view the list of students who are currently enrolled in the subject.



**Figure 3.42 Case Page –Academic Faculty**

* This is where the user will create a new case. The text box that are tinted with grey color cannot be edited, and automatically generated by the searched student’s information. The Open Cases on the right corner of Case Detail displays the total cases that the student has.
* The Case Detail must be all filled before it allows to click done.
* The Nature indicates if the case is Academic or Curriculum.
* The Category is if it is about a subject tutorial, academic outcome, behavior concern, project, pre-registration, enrollment or flow chart.
* The Sub-Category indicates if it is a current subject or past subject, Passed or Failed subject, attendance, grade consultation, deliverables, project output, subject advising, load revision, flowchart advising, or flow chart changing.
* The Case Title is simply the title of the new case.
* Case Priority indicates if the case is in High Priority, Normal or Hot. Hot means the case created for the student is at risk. While the other priority depends on the faculty’s note.
* Subject is the list of subjects that the faculty is handling with the specific student.
* Faculty is the name of other faculty member that you will message for a tutorial or consultation, etc.
* Current Case Notes displays all the notes sent by the student and the faculty in a small note box.
* Add Note is when the faculty wants to send the student another note, the note will just add up to the notes sent.
* See All Notes will view all the notes on a larger window.
* Get Case Number is for generating a case number whenever the user is done filling up the form.
* Close Case is a command button when the case opened is already done.
* Done is for creating the case.
* Clear is for removing all the data inputted in Case Details.
* Back is for returning to the List of Students page.



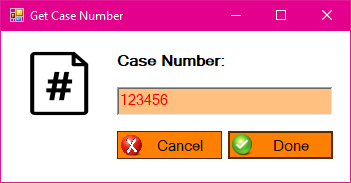
**Figure 3.43 Add Notes**

The figure 3.43 window lets the user construct a note and add to the history of their conversation. Save button is when the user is done typing and ready to add his/her note.  It also has a vertical scroll bar in case the user typed a long note.



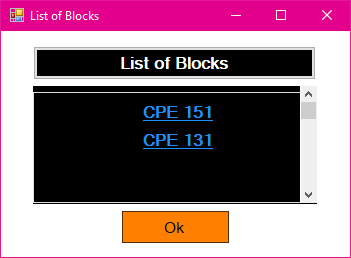
**Figure 3.44 View All Notes**

This page displays the history of the notes sent by the two users.



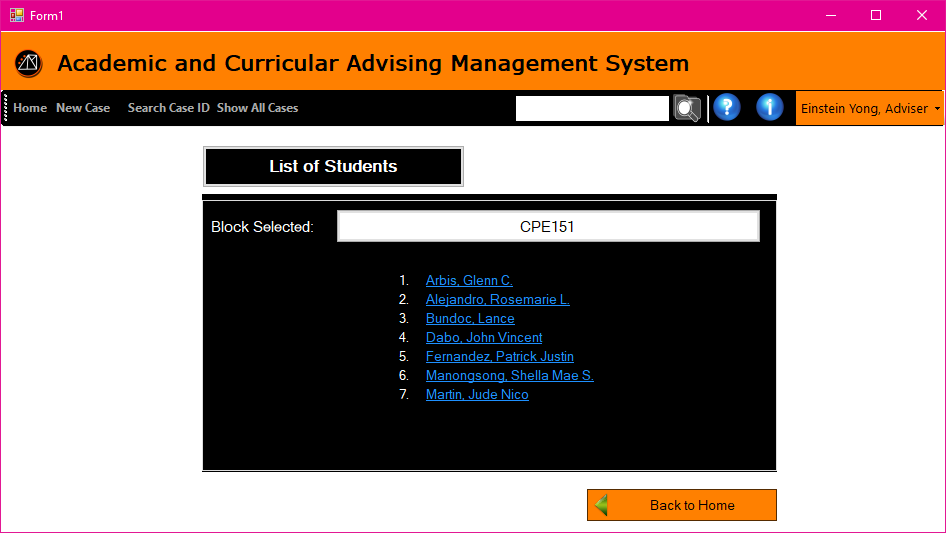
**Figure 3.45 Get Case Number (Pop up)**

This pop-up window will display whenever Get Case No. button is clicked to generate a unique case ID number when creating a new case. 



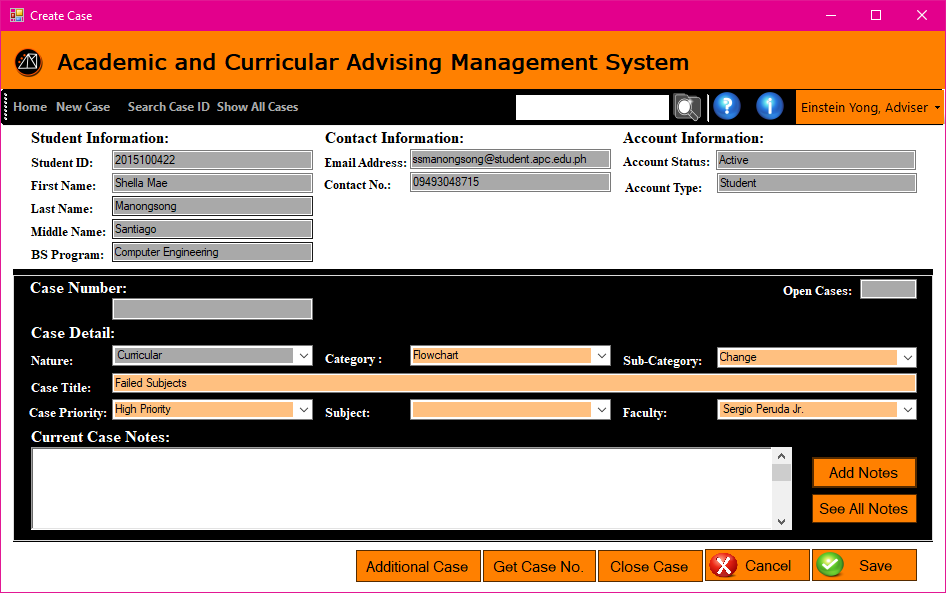
**Figure 3.46 List of Blocks (Pop up)**

This page allows the user to view all block he/she handles.



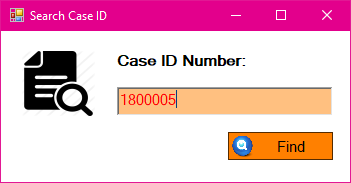
**Figure 3.47 List of Students by Block**

This window displays the list of students in a block that is hyperlinked to decide whether to create a new case to the student or view all his/her cases.



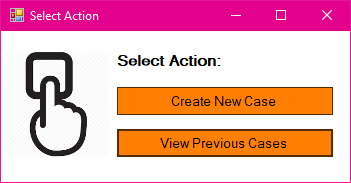
**Figure 3.48 Case Page – Curricular Adviser**

The buttons and displays are the same with figure 3.37. This page allows the adviser to create a case intended only in the nature of curricular.



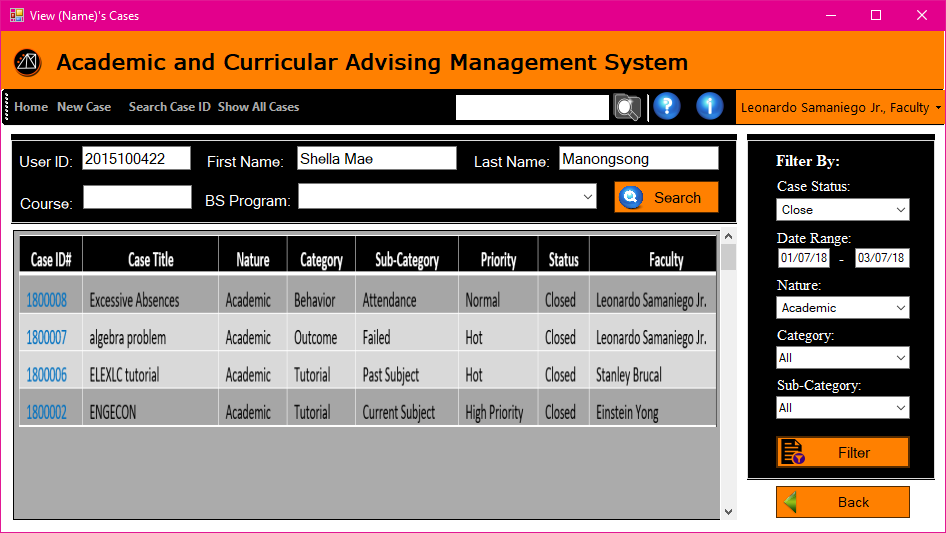
**Figure 3.49 Search Case ID (Pop up)**

The page allows the user to enter the case ID prior to pulling up the case.



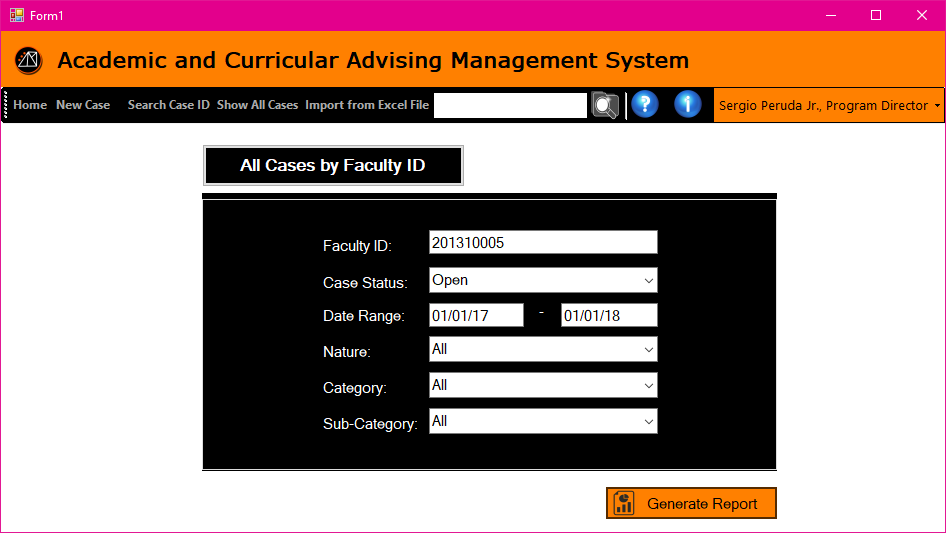
**Figure 3.50 Select Action (Pop up)**

This page allows the user to select he would like to create a new case or view previous cases.



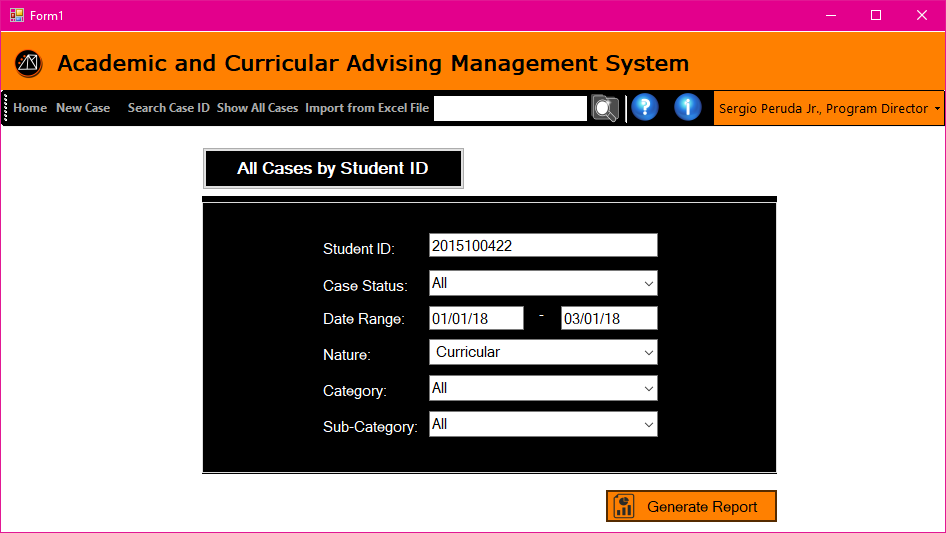
**Figure 3.51 Filtering Page**

This page allows any faculty level user to search for case by entering either the user ID, first name, last name, course, or BS Program. It will produce a queue of results that are related information provided. It also has a section to further filter the search result.



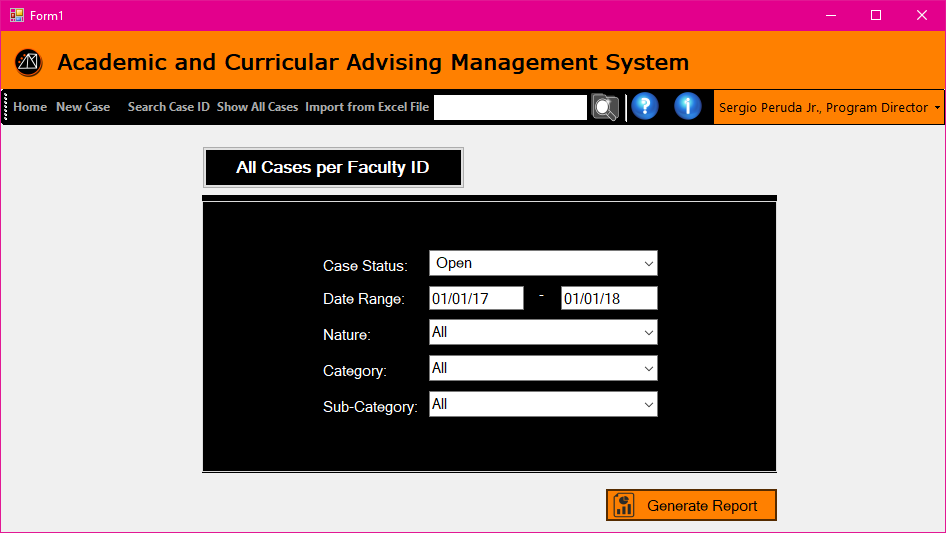
**Figure 3.52 Generate Report – All Cases by Faculty ID**

The page allows the PD/XD user to enter the information that will be needed in generating a report. The form needs to be filled up if the user is looking for case related to one faculty only.



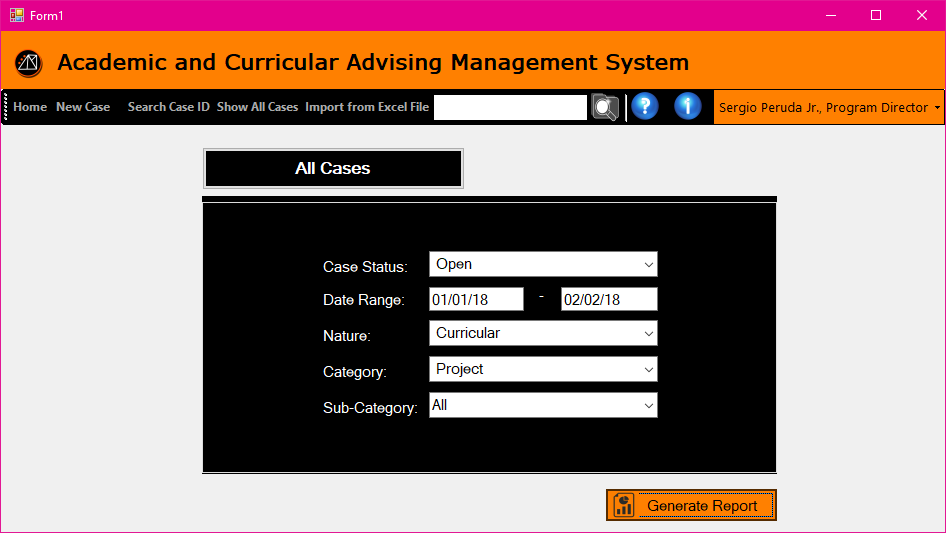
**Figure 3.53 Generate Report – All Cases by Student ID**

The page allows the PD/XD user to enter the information that will be needed in generating a report. The form needs to be filled up if the user is looking for case related to one student only.



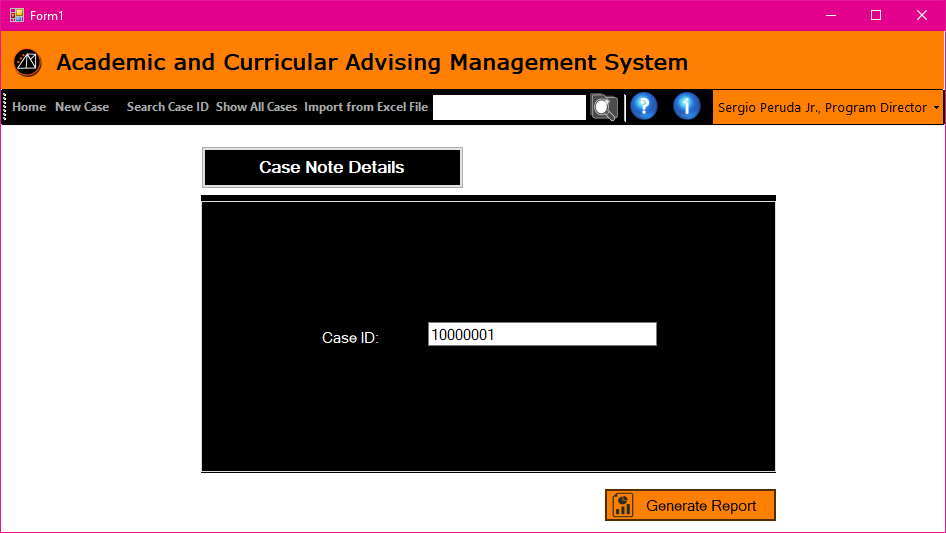
**Figure 3.54 Generate Report – All Cases per Faculty ID**

The page allows the PD/XD user to enter the information that will be needed in generating a report. The form needs to be filled up is the user is looking for all faculty related cases within a date range.



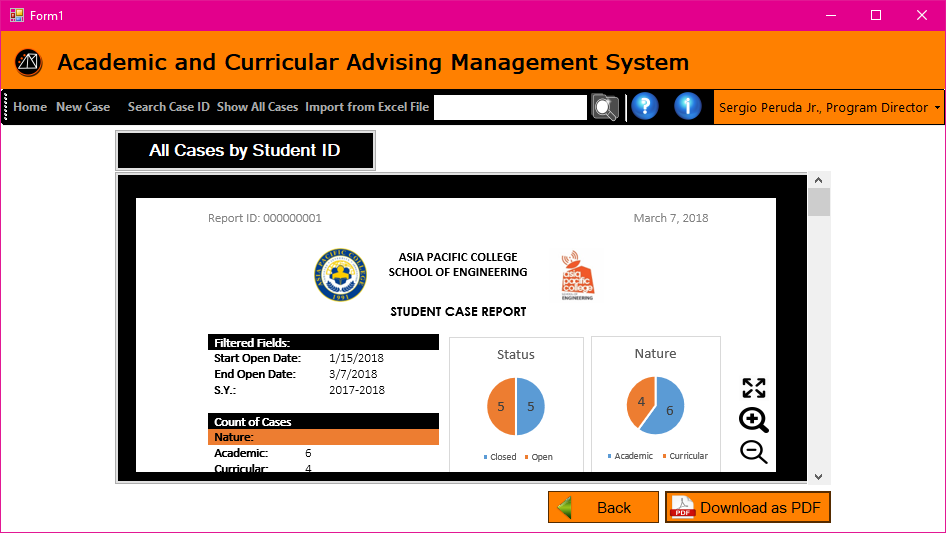
**Figure 3.55 Generate Report – All Cases**

The page allows the PD/XD user to enter the information that will be needed in generating a report. This form needs to be filled up if the user would like to pull up all case within a data range for a specific case status.



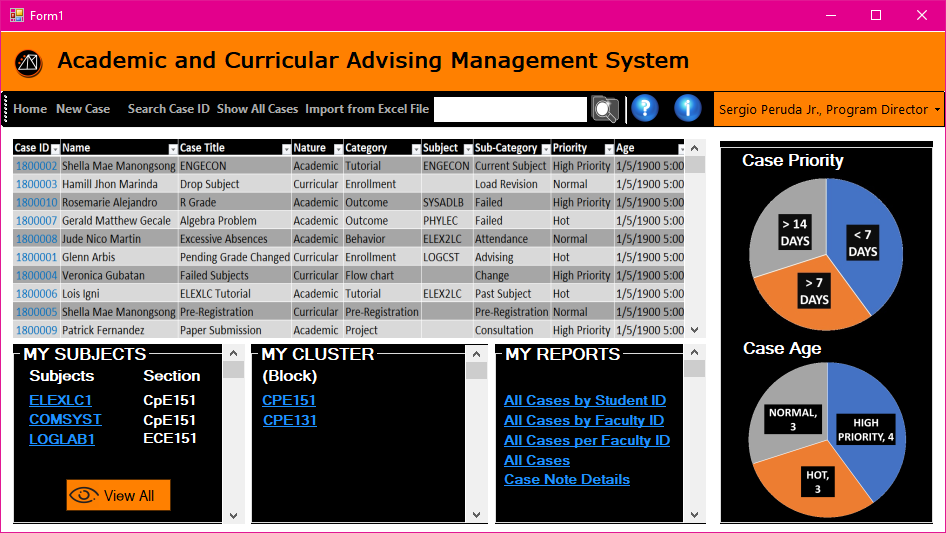
**Figure 3.56 Generate Report – Case Note Details**

This allows the PD/XD user to search for a specific case to create a report with.



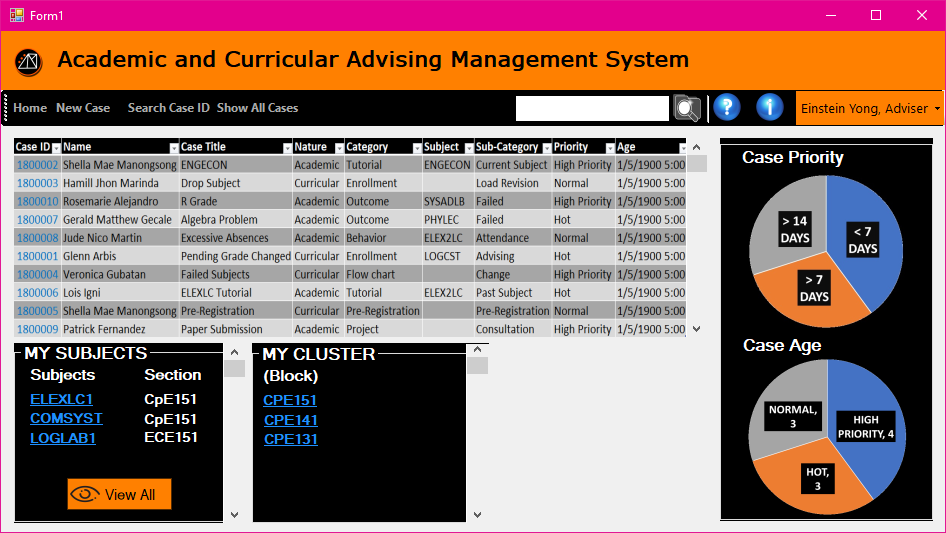
**Figure 3.57 View Report**

The page shows the preview of the report that the user created.



**Figure 3.58 Program Director Homepage**

Fig 3.56 is a form similar to the Executive Director's privilege, which displays the home page of the Program Director. This is where the summarized table of all cases is shown, and the pie chart of case priorities and its age. There is also a list of the subjects that the Program Director is handling. There is also a cluster that has a list of all engineering students. The reports have 4 types. The user can create a new case, search case ID, and show all the cases created.



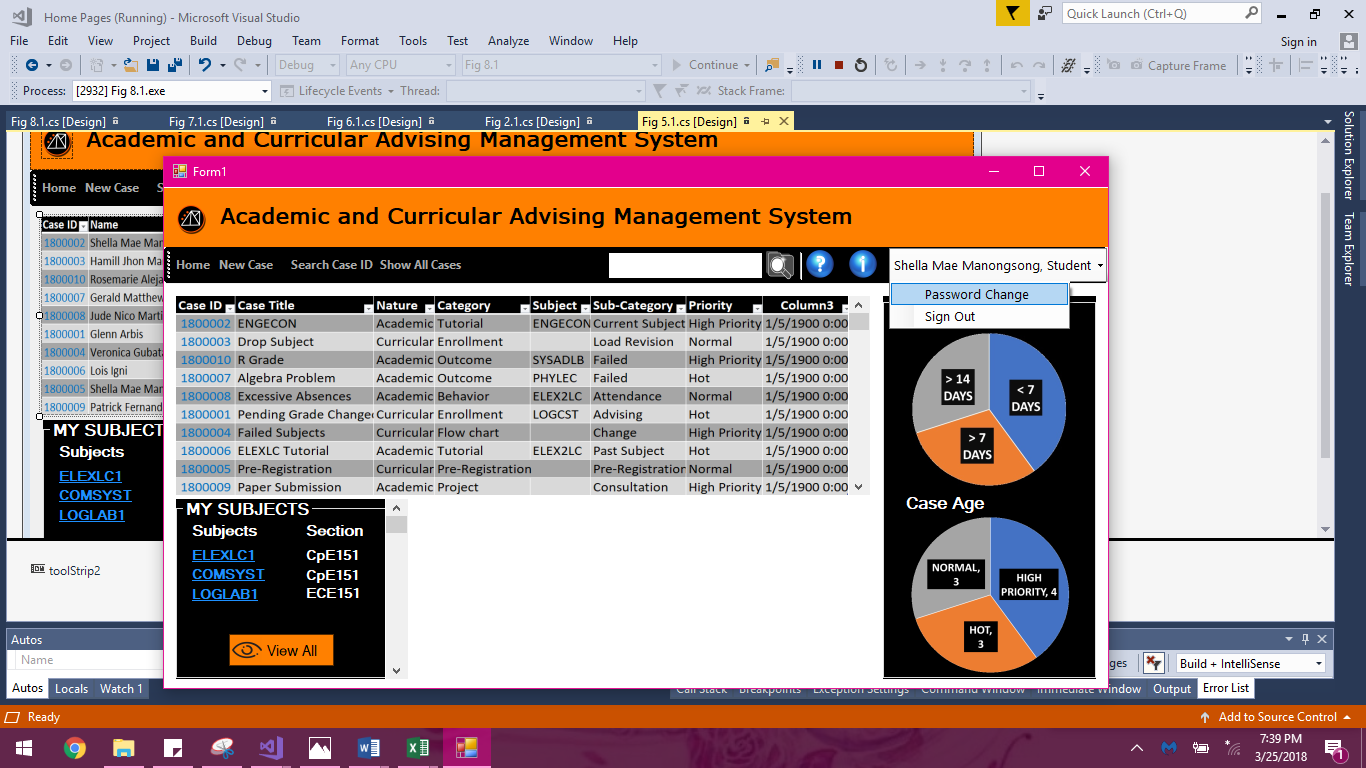
**Figure 3.59 Adviser Homepage**

The page shows the homepage for adviser. This page is similar to the faculty page. The added section is My Cluster that displays the hyperlinks of the blocks he/she handles.



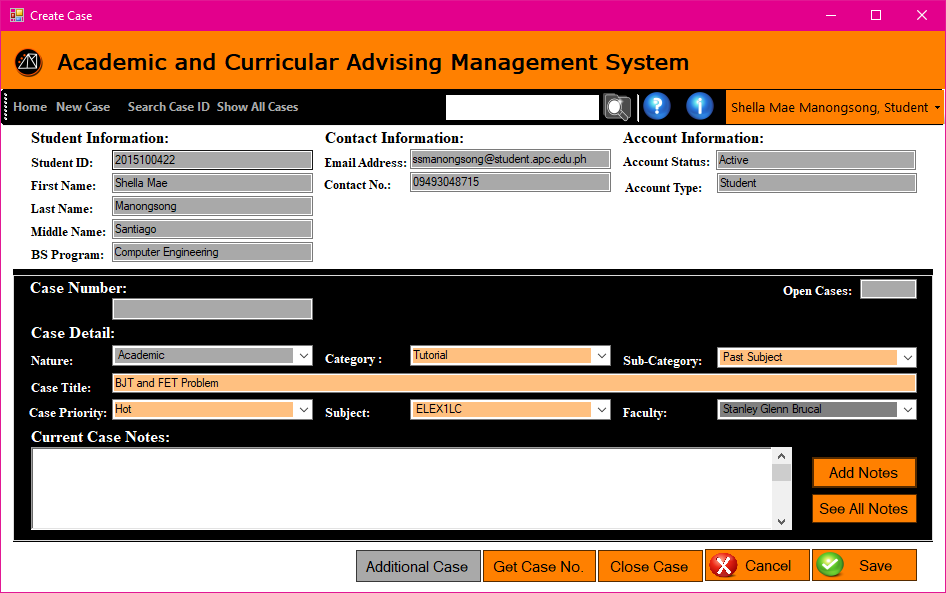
**Figure 3.60 Faculty Homepage**

This is the faculty’s homepage after logging in successfully.



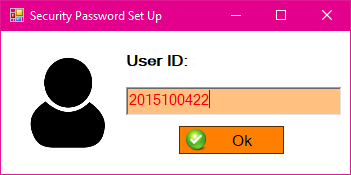
**Figure 3.61 Student Homepage**

This is the homepage ofthe student with my subject box and pie graph to see the summary of case priority and case age. He can create a case, search case ID and show all cases.



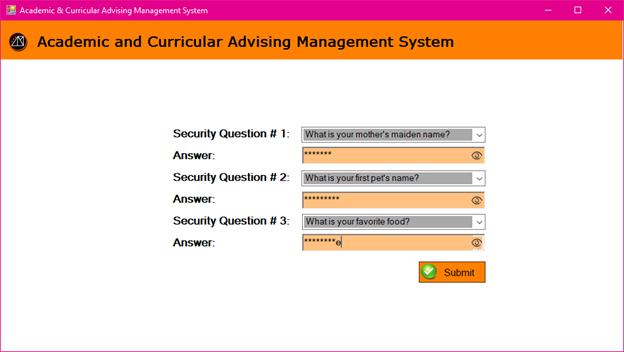
**Figure 3.62 Case Page – Academic / Curricular Student**

Fig 3.62 is the case page where in the student allows to create a case but cannot create additional case, because the faculty can only decide if it is an additional case and the case created by student is still in request until the faculty accepts it.

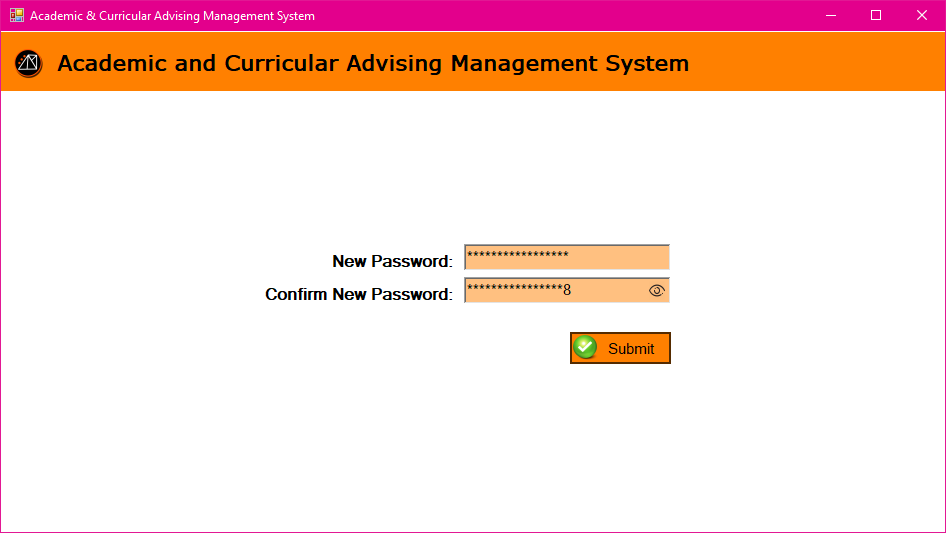


**Figure 3.63 Forgot Password – School ID (Pop-up)**

When the user clicked the Forgot Password button, this pop-up window will display. The user have to enter his/her user ID to identify which account is it.

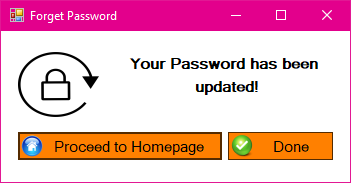
****

**Figure 3.64 Forgot Password – Security Question**

The user is required to answer the security questions he/she has chosen during his/her registration. Answering them all correctly will give the user the privilege in changing his/her password.  
   
****

**Figure 3.65 Forgot Password – Entering the New Password**

The user can now enter his/her new desired password twice for confirmation/case sensitivity purposes.



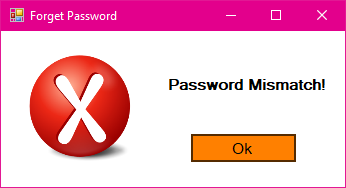
**Figure 3.66 Forgot Password – Update Successful (Pop up)**

Confirmation and a notification popup where the user will be redirected to their homepage.



**Figure 3.67 Forgot Password – Invalid Answer(s) Prompt (Pop up)**

A popup notifying the user that his/her answers in the security questions are incorrect.



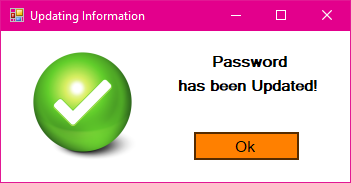
**Figure 3.68 Forgot Password – New Password Input Mismatch (Pop up)**

A popup notifying the user that his/her newly entered passwords are mismatched. 



**Figure 3.69 Updating Password**

When the user wants to change his/her password, the user needs to comply this form for security purposes.



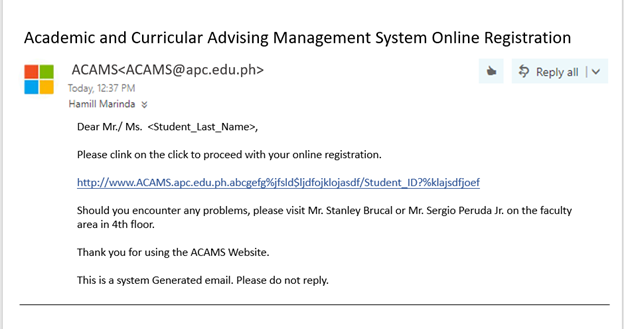
**Figure 3.70 Updating Password – Update Successful (Pop up)**

When the user is done with the form this pop-up message will display indicating the user’s password is updated successfully.



**Figure 3.71 Import from Excel File to Database**

This is the sample of user interface when the user picked a section where he/she wanted to add the information from the excel file he/she imported.



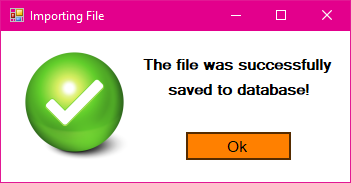
**Figure 3.72 Email**

After importing the list of students by the Executive and Program Directors, a default email will be sent to the APC email of the students listed. The students will receive an email containing a link direct to the registration of the system.



**Figure 3.73 Step 2 User Information not edited**

After clicking the link from the email sent by the system, this page will display. The text boxes with grey highlights are not editable, the information stored in it is bases on the data imported by the executive/program director, and the student will only fill the data for their contact number.



**Figure 3.74 Successful Saving of data (Pop-up)**

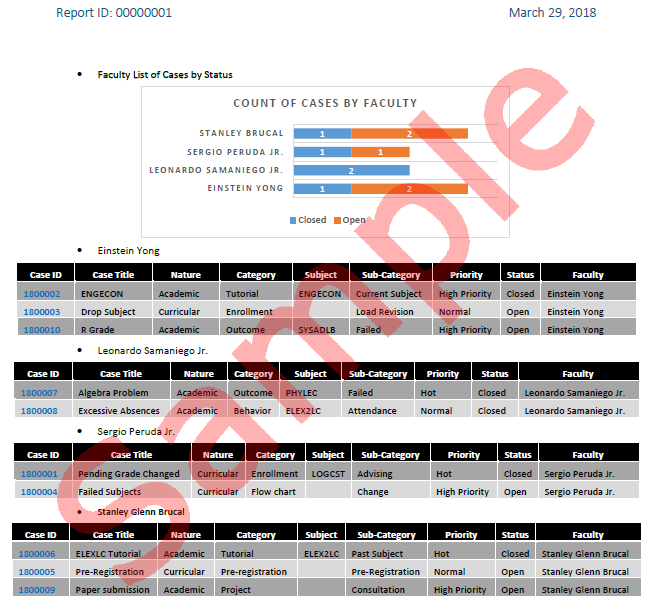
After importing of excel file containing data that will be added to the database, this pop-up window will display indicating that the data uploaded is saved successfully.

**3.6 Reports**

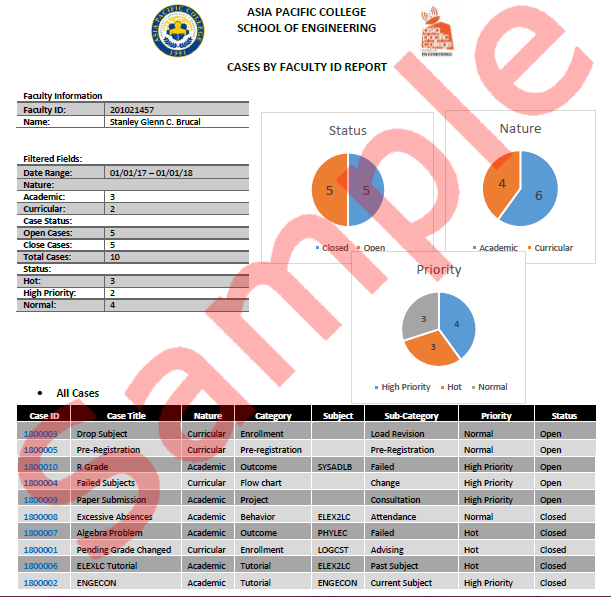


**Figure 3.75 All Cases A**

The figures 3.75 and 3.76 are examples of a multiple case generated report. It consists of the list of cases selected by the XD and PD from their inputted date range and categories.

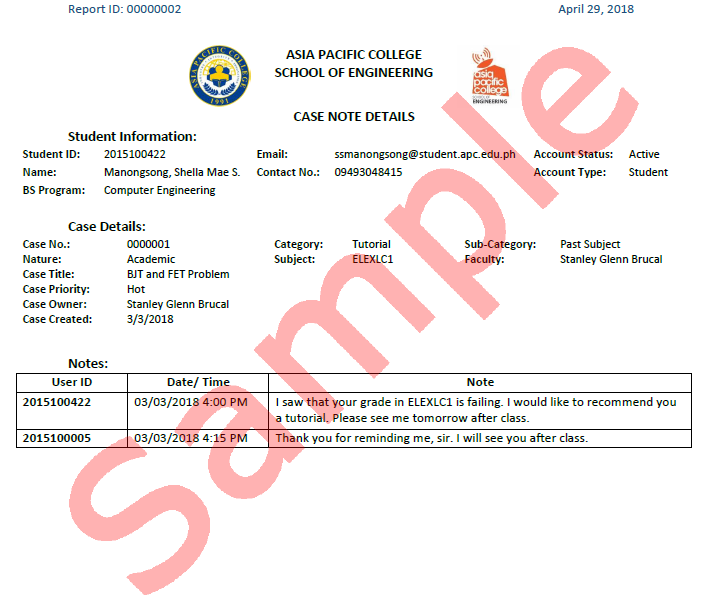
****

**Figure 3.76 All Cases B**

****

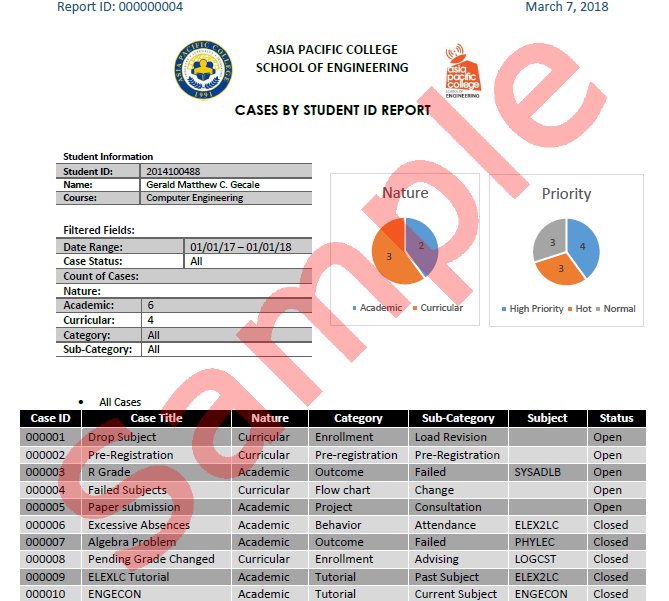
**Figure 3.77 Cases by Faculty ID Report**

In figure 3.77, the report generated a report from the XD, or PD user. A report that shows all the reports open or closed involving themselves**.**

****

**Figure 3.78 Case Note Details Report**

The generated report in figure 3.79 is a report of an individual student selected by either the XD, or PD, showing all the case details along with the interaction via case notes between the case owner and the selected student.

****

**Figure 3.79 Cases by Student ID Report**

The figure 3.80 is a generated report that shows both open and closed cases of a selected student with the case details. The report shows only the cases from the selected date range.

**3.7 Methods, Tools, and Techniques**

**Table 3.2. Methods**

|  |  |  |
| --- | --- | --- |
| **Methods** | **Tools** | **Techniques** |
| Automated | * Computer * Internet Connection * Database Server | User Database Management System |
| Automated | * Computer * Internet Connection * Case Database Server | Case Database Management System |
| Automated | * MySQL Workbench * Draw.io | Entity Relationship Diagram: Crowfoot Notation |
| Automated | * Draw.io | Business Process Model: Data Flow Diagram (LvL0, LvL1, Lvl2) |
| Automated | * Microsoft Excel | Data Dictionary: Normalization |

**3.7.1. Tools**

* MySQL Workbench – the group used MySQL Workbench to create the system's Entity Relationship Diagram. This software helped us in properly placing the mandatory forms of the entities with the crowfoot method.

* Draw.io - this online drawing tool was used for creating the extended Entity Relationship Diagram, and the Data Dictionary, from the Context Diagram (LvL0), LvL1, and Lvl2.

* Microsoft Visual Studio (2013-2016) - this software was our vital tool in creating the group's user interfaces.

* Microsoft OneNote – this online file collection and collaboration tool was used for the organization and arrangement of our documentation, as well as our images or figures for the documentation.

**3.6.2. Techniques**

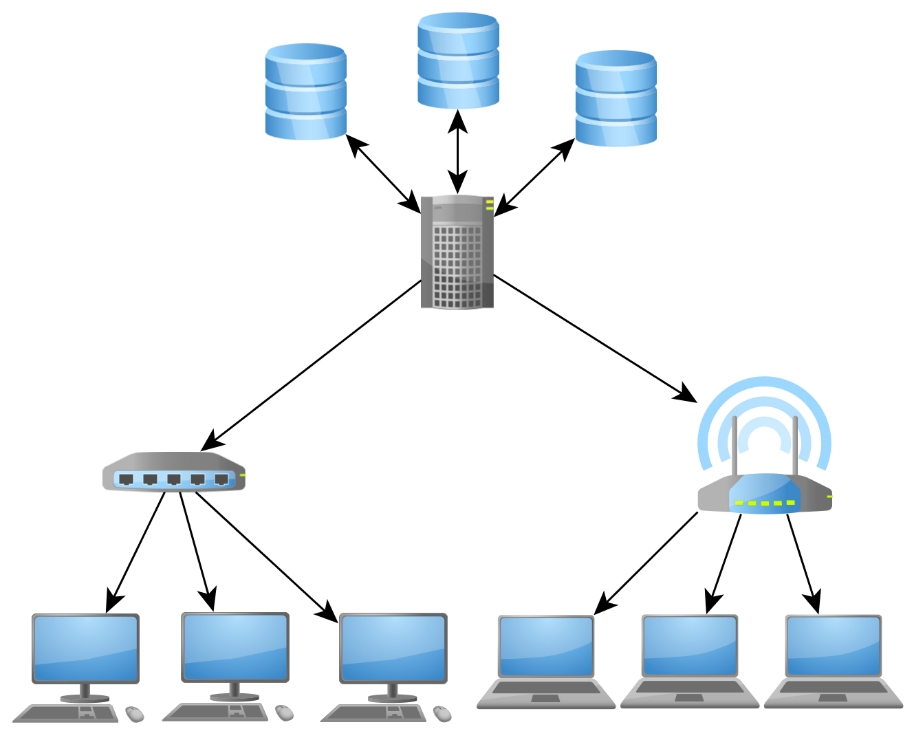
The proponent conducted interviews with the client in order to fully understand the processes and functions needed in the system. People who are involved in the system was distinguished from the interviews as well. For the refinement of the processes and specifications of the system, follow-up consultation and interviews were actively conducted in order for the additional changes that were needed are clarified and verified.

* Crowfoot Notation – a database management method used in the Entity Relationship Diagram to assign the relationship of entities with other entities. The lines used to connect the entities have certain figures at the ends of the lines depending on the nature of the relationship. Figures that looks like a crow's foot getting the name Crowfoot Notation.

* Normalization of Tables – the processes of reconstructing a database system's tables to reduce data redundancy and makes the tables more organized and accessible in an orderly fashion.

**3.8 Infrastructure**

From the figure 3.81, the minimum requirements of the Academic & Curricular Advising Management System that being the system to be an intranet, a server, typically a basic workstation server for schools with sufficient internet speeds for optimal data transfer. Yes, the system is an intranet web-based system, but it is still necessary to have proper internet connection inside the network for maximum throughput of the networks file transfer speed. 



**Figure 3.80 Infrastructure**

**Internet and Network Speed Minimum System Requirements (For a small population e.g. 100 Users)**

Internet:

* Upload Speed: 15 Megabits per second
* Download Speed: 15 Megabits per second

Network: 

* Cable: CAT5E with the rating of 1000 Megabits per second, Ethernet 1000Base-T.

**Server's Minimum System Specifications Required for Possible Outside Deployment**

**OS:**Linux (Oracle, Slackware, etc. Any basic server-type Linux OS would suffice)

**Processor Specifications:** Multicore/Multi-threaded CPU (4 Cores/8 Threads)

The more threads the better depending on the quantity of the population that will use the system.

**RAM:**16GB, frequency higher than the old 1600MHz. DDR3 is still good for optimal performance but if the platform of the motherboard and processor supports DDR4, DDR4 would be most optimal for the system.

**Storage:**Hard Disk Drive with the rating of 7200RPM would be the minimum, but the utilization of an SSD, although expensive, would provide the best performance for the system.

**Graphics Card:** The system being a basic file and reports system, no powerful graphics card is necessary. The Server's System Unit needs a graphics processing unit because majority of server processors don't come with integrated graphics processing unit.

Possible minimum requirements for graphics are the following:

* NVidia GTX1030 2GB GDDR5
* AMD RX550 2GB GDDR5

**Power Supply:**Depends on the power required to power the CPU + GPU + Motherboard.    
Recommended Wattage: 600W or higher, with 80% Efficiency Rating preferably with a Bronze Rating or Higher.

**User's Minimum System Specification Requirements for his/her computer**

**Processor:**Intel Pentium 4th generation or higher,

Recommended Processor: Intel i3-4th generation or higher.

**RAM:**4GB DDR3 1600MHz+ or DDR4 2133MHz+

Recommended RAM Capacity: 8GB

**GPU:** Intel HD Graphics (Integrated) / AMD Radeon Graphics (Integrated)

Recommended: Any Discrete Graphics whether the user is using a desktop computer or portable computer (Notebook/Laptops, etc.).

**OS:**Windows 7 (64-bit) or higher versions of windows

**Web Browser:**Microsoft Edge 10, Google Chrome

**CHAPTER IV**

**MANAGERIAL PROCESS PLAN**

**4.1 Start-Up Plan**

**4.1.1 Estimates**

**Table 4.1 Estimates**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PERSONNEL** | **RATE/HR** | **# PERSONNEL** | **HOUR RENDERED** | **COST** |
| Project Manager | 500 | 1 | 315 | 157500 |
| Analyst | 400 | 1 | 306 | 122400 |
| Database Manager | 400 | 3 | 306 | 122400 |
| Programmer | 300 | 5 | 503 | 150900 |
| Graphic Designer | 200 | 1 | 660 | 132000 |
| Trainer | 100 | 1 | 4 | 400 |
| QA Analyst | 100 | 1 | 115 | 11500 |
| Documentarist | 100 | 1 | **22** | 2200 |
| **TOTAL** |  | **14** | **2231** | **699300** |

Table 4.1 specifies the estimated cost of the personnel, the estimated wages, headcount, hours, and cost in conducting the project. The estimation is based on analogy, rule of thumb, standard unit of size, cost mode, and historical data.

**4.1.2 Staffing**

**Table 4.2 Staffing**

|  |  |
| --- | --- |
| **PERSONNEL** | **# PERSONNEL** |
| Project Manager | 1 |
| Analyst | 1 |
| Database Manager | 3 |
| Programmer | 5 |
| Graphic Designer | 1 |
| Trainer | 1 |
| QA Analyst | 1 |
| Documentarist | 1 |
| **TOTAL** | **14** |

Table 4.2 specifies the number of staff required per position for the whole project. The estimation is based from the number of personnel needed per project phase and duration of personnel requirement. Data from 4.3 is the main basis for estimation.

**4.1.3 Resource Acquisition**

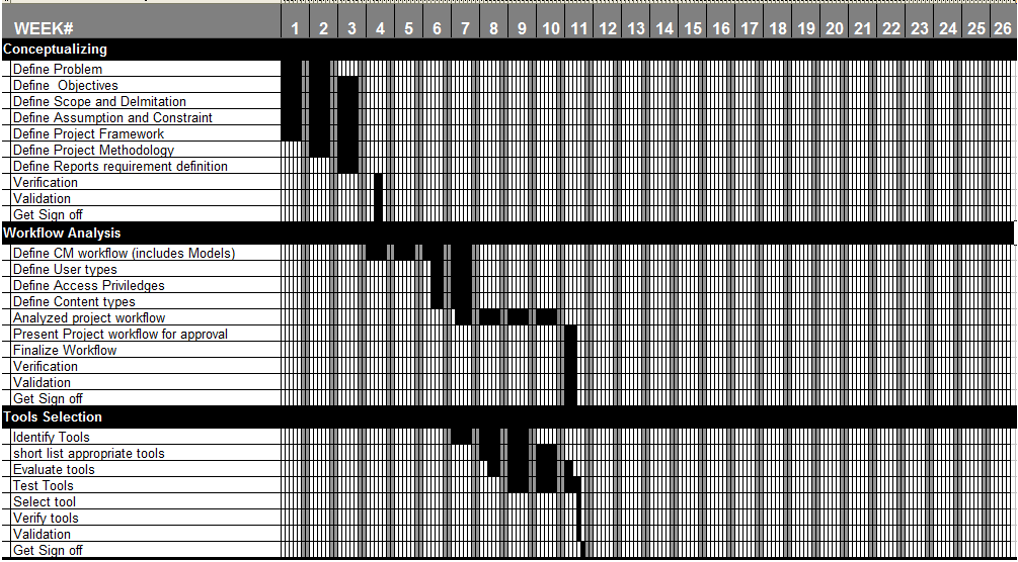
Since the system is a web-based system with PHP as its language, the proponent will utilize any Phpstorm as the integrated development environment. MySQL is used for the Entity Relationship Diagram, but the actual database will utilize Microsoft Access or MySQL Server. The resources have installers available in the market or in the internet. The server that the system will use once implemented will be provided by the Information and Technology Resource Office (ITRO) of Asia Pacific College.

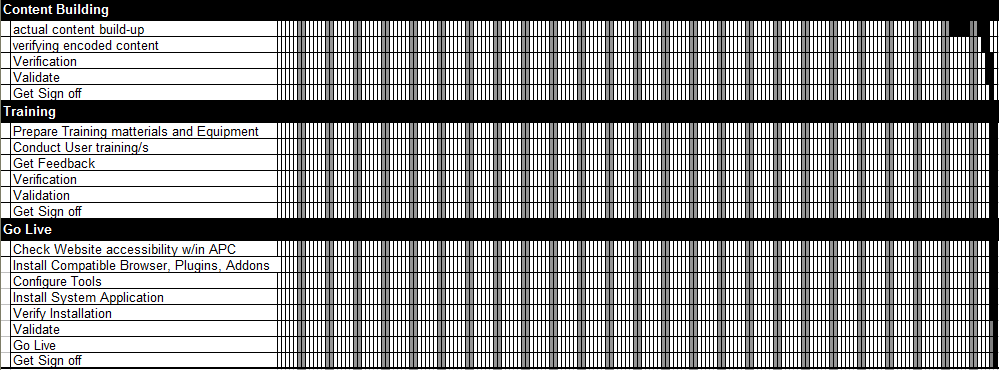
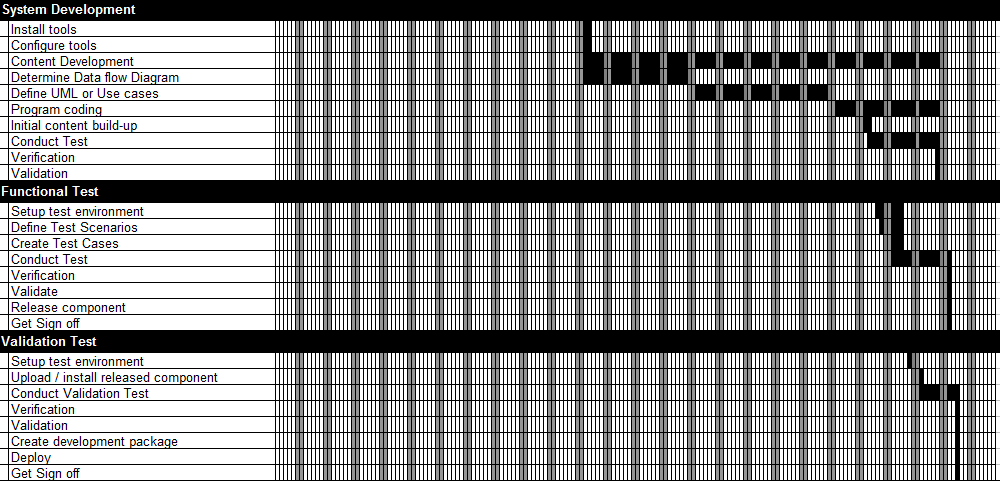
**4.2 Work Plan**

**4.2.1 Work Breakdown Structure**

Table 4.3 shows the Work Breakdown Structure in a graphical illustration of the phases of the project. The rows in the tables shows the major activities and its actual activities. The column is divided by weeks and a week is divided by days with corresponding date. If an activity is expected to start and end to date range, it will be highlighted on the table allowing the reader to easily manage the project and view the project status.

**Table 4.3 Work Breakdown Structure**





**4.6. Schedule Allocation**

**Table 4.4 Work Breakdown Structure per Major Activity**

|  |  |  |  |
| --- | --- | --- | --- |
| **MAJOR ACTIVITY** | **START DATE** | **END DATE** | **DAYS** |
| Conceptualizing | 15-Jan-18 | 26-Jan-18 | 12 |
| Workflow Analysis | 05-Feb-18 | 28-Mar-18 | 52 |
| Tools Selection | 26-Feb-18 | 30-Mar-18 | 33 |
| System Development | 02-Apr-18 | 01-Jul-18 | 91 |
| Functional Test | 14-Jun-18 | 02-Jul-18 | 19 |
| Validation Test | 22-Jun-18 | 04-Jul-18 | 13 |
| Content Building | 02-Jul-18 | 12-Jul-18 | 11 |
| Training | 12-Jul-18 | 12-Jul-18 | 1 |
| Go Live | 12-Jul-18 | 13-Jul-18 | 2 |

Table 4.4 shows the specified start and end estimated date per major activity and the expected days for completion.

**Table 4.5 Work Breakdown Structure per with Actual Activity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Conceptualizing** | **Start Date** | **End Date** | **Man Hours** |
| Define Problem | 15-Jan-18 | 01-Feb-18 | 12 |
| Define Project Objectives | 15-Jan-18 | 02-Feb-18 | 12 |
| Determine requirements | 15-Jan-18 | 02-Feb-18 | 12 |
| Define Business Process | 15-Jan-18 | 02-Feb-18 | 12 |
| Define Project Framework | 15-Jan-18 | 02-Feb-18 | 12 |
| Define Project Methodology | 19-Jan-18 | 02-Feb-18 | 24 |
| Define Reports requirement definition | 22-Jan-18 | 02-Feb-18 | 12 |
| Verification | 07-Feb-18 | 08-Feb-18 | 1 |
| Validation | 07-Feb-18 | 08-Feb-18 | 1 |
| get sign off | 07-Feb-18 | 08-Feb-18 | 1 |
| **Workflow Analysis** | **Start Date** | **End Date** | **Man Hours** |
| Define CM workflow | 05-Feb-18 | 02-Mar-18 | 96 |
| Define User types | 21-Feb-18 | 02-Mar-18 | 12 |
| Define Access privileges | 22-Feb-18 | 02-Mar-18 | 18 |
| Define Content types | 23-Feb-18 | 02-Mar-18 | 18 |
| Analysed project workflow | 27-Feb-18 | 23-Feb-18 | 138 |
| Present Project workflow for approval | 26-Mar-18 | 28-Mar-18 | 3 |
| Finalize Workflow | 26-Mar-18 | 28-Mar-18 | 15 |
| Verification | 26-Mar-18 | 28-Mar-18 | 8 |
| Validation | 26-Mar-18 | 28-Mar-18 | 8 |
| get sign off | 26-Mar-18 | 28-Mar-18 | 1 |
| **Tools Selection** | **Start Date** | **End Date** | **Man Hours** |
| Identify Tools | 26-Feb-18 | 16-Mar-18 | 81 |
| short list appropriate tools | 05-Mar-18 | 23-Mar-18 | 81 |
| Evaluate tools | 07-Mar-18 | 27-Mar-18 | 81 |
| Test Tools | 12-Mar-18 | 29-Mar-18 | 72 |
| Select tool | 29-Mar-18 | 29-Mar-18 | 3 |
| Verify tools | 29-Mar-18 | 29-Mar-18 | 3 |
| Validation | 29-Mar-18 | 29-Mar-18 | 3 |
| get sign off | 30-Mar-18 | 30-Mar-18 | 1 |
| **System Development** | **Start Date** | **End Date** | **Man Hours** |
| Install tools | 02-Apr-18 | 03-Apr-18 | 7 |
| Configure tools | 02-Apr-18 | 03-Apr-18 | 7 |
| Content Development | 02-Apr-18 | 29-Jun-18 | 669 |
| Determine Data Flow Diagram | 02-Apr-18 | 27-Apr-18 | 162 |
| Define UML or Use cases | 30-Apr-18 | 08-Nov-08 | 210 |
| Program coding | 02-Jun-18 | 29-Jun-18 | 159 |
| Initial content build-up | 11-Jun-18 | 12-Jun-18 | 4 |
| Conduct Test | 12-Jun-18 | 29-Jun-18 | 46 |
| Verification | 29-Jun-18 | 29-Jun-18 | 3 |
| Validation | 29-Jun-18 | 29-Jun-18 | 1 |
| **Functional Test** | **Start Date** | **End Date** | **Man Hours** |
| Setup test environment | 14-Jun-18 | 20-Jun-18 | 27 |
| Define Test Scenarios | 15-Jun-18 | 20-Jun-18 | 25 |
| Create Test Cases | 18-Jun-18 | 20-Jun-18 | 20 |
| Conduct Test | 18-Jun-18 | 02-Jul-18 | 38 |
| Verification | 02-Jul-18 | 02-Jul-18 | 3 |
| Validate | 02-Jul-18 | 02-Jul-18 | 1 |
| Release component | 02-Jul-18 | 02-Jul-18 | 1 |
| Get Sign off | 02-Jul-18 | 02-Jul-18 | 1 |
| **Validation Test** | **Start Date** | **End Date** | **Man Hours** |
| Setup test environment | 22-Jun | 22-Jun | 4 |
| Upload / install released component | 26-Jun | 26-Jun | 4 |
| Conduct Validation Test | 26-Jun | 04-Jul | 18 |
| Verification | 04-Jul | 04-Jul | 3 |
| Validation | 04-Jul | 04-Jul | 1 |
| Create development package | 04-Jul | 04-Jul | 6 |
| Deploy | 04-Jul | 04-Jul | 1 |
| Get Sign off | 04-Jul | 04-Jul | 1 |
| **Content Building** | **Start Date** | **End Date** | **Man Hours** |
| Actual content build-up | 03-Jul-18 | 11-Jul-18 | 18 |
| Verifying encoded content | 10-Jul-18 | 11-Jul-18 | 3 |
| Verification | 11-Jul-18 | 12-Jul-18 | 3 |
| Validate | 11-Jul-18 | 12-Jul-18 | 3 |
| Get sign off | 13-Jul-18 | 13-Jul-18 | 1 |
| **Training** | **Start Date** | **End Date** | **Man Hours** |
| Prepare Training materials and Equipment | 12-Jul-18 | 13-Jul-18 | 8 |
| Conduct User training/s | 12-Jul-18 | 13-Jul-18 | 3 |
| Get Feedback | 12-Jul-18 | 13-Jul-18 | 3 |
| Verification | 12-Jul-18 | 13-Jul-18 | 3 |
| Validation | 12-Jul-18 | 13-Jul-18 | 3 |
| Get sign off | 12-Jul-18 | 13-Jul-18 | 1 |
| **Go Live** | **Start Date** | **End Date** | **Man Hours** |
| Check Website accessibility w/in APC | 12-Jul-18 | 13-Jul-18 | 4 |
| Install Compatible Browser, Plugins, Addons | 12-Jul-18 | 13-Jul-18 | 2 |
| Configure Tools | 12-Jul-18 | 13-Jul-18 | 2 |
| Install System Application | 12-Jul-18 | 13-Jul-18 | 2 |
| Verify Installation | 12-Jul-18 | 13-Jul-18 | 4 |
| Validate | 12-Jul-18 | 13-Jul-18 | 3 |
| Go Live | 13-Jul-18 | 13-Jul-18 | 1 |
| Get Sign off | 13-Jul-18 | 13-Jul-18 | 1 |

Table 4.5 shows the specified start and end estimated date of completion per actual activity. The actual activity is grouped by major activity.

**Table 4.6. Resource Allocation**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Project Manager** | **Analyst** | **Database Manager** | **Programmer** | **Graphic Designer** | **Trainer** | **QA Analyst** | **Documentarist** |
|
|
|
|  | | **Manpower** | | | | | | | |
| **Conceptualizing** | | **PM** | **AN** | **DM** | **PR** | **GD** | **TR** | **QA** | **DO** |
|  | Define Problem | 1 |  |  |  |  |  |  |  |
|  | Define Project Objectives | 1 |  |  |  |  |  |  |  |
|  | Determine requirements | 1 |  |  |  |  |  |  |  |
|  | Define Business Process | 1 |  |  |  |  |  |  |  |
|  | Define Project Framework | 1 |  |  |  |  |  |  |  |
|  | Define Project Methodology | 1 |  |  |  |  |  |  |  |
|  | Define Reports requirement definition | 1 |  |  |  |  |  |  |  |
|  | Verification | 1 |  |  |  |  |  |  |  |
|  | Validation | 1 |  |  |  |  |  |  |  |
|  | Get Sign off | 1 |  |  |  |  |  |  |  |
| **Workflow Analysis** | | **PM** | **AN** | **DM** | **PR** | **GD** | **TR** | **QA** | **DO** |
|  | Define CM workflow | 1 | 1 | 1 |  |  |  |  |  |
|  | Define User types | 1 | 1 | 1 |  |  |  |  |  |
|  | Define Access privileges | 1 | 1 | 1 |  |  |  |  |  |
|  | Define Content types | 1 | 1 | 1 |  |  |  |  |  |
|  | Analysed project workflow | 1 | 1 | 1 |  |  |  |  |  |
|  | Present Project workflow for approval | 1 |  |  |  |  |  |  |  |
|  | Finalize Workflow | 1 | 1 | 1 |  |  |  |  |  |
|  | Verification | 1 | 1 | 1 |  |  |  |  |  |
|  | Validation | 1 | 1 | 1 |  |  |  |  |  |
|  | Get Sign off | 1 |  |  |  |  |  |  |  |
| **Tools Selection** | | **PM** | **AN** | **DM** | **PR** | **GD** | **TR** | **QA** | **DO** |
|  | Identify Tools | 1 | 1 | 1 |  |  |  |  |  |
|  | short list appropriate tools | 1 | 1 | 1 |  |  |  |  |  |
|  | Evaluate tools | 1 | 1 | 1 |  |  |  |  |  |
|  | Test Tools | 1 | 1 | 1 |  |  |  |  |  |
|  | Select tool | 1 | 1 | 1 |  |  |  |  |  |
|  | Verify tools | 1 | 1 | 1 |  |  |  |  |  |
|  | Validation | 1 | 1 | 1 |  |  |  |  |  |
|  | Get Sign off | 1 |  |  |  |  |  |  |  |
| **System Development** | | **PM** | **AN** | **DM** | **PR** | **GD** | **TR** | **QA** | **DO** |
|  | Install tools | 1 | 1 | 1 | 1 |  |  |  |  |
|  | Configure tools | 1 | 1 | 1 | 1 |  |  |  |  |
|  | Content Development | 1 |  |  |  | 5 |  |  |  |
|  | Determine Data flow Diagram | 1 | 1 | 1 | 3 |  |  |  |  |
|  | Define UML or Use cases | 1 | 1 | 1 | 3 |  |  |  |  |
|  | Program coding | 1 | 1 | 1 | 3 |  |  |  |  |
|  | Initial content build-up |  |  |  |  |  |  |  | 1 |
|  | Conduct Test | 1 | 1 | 1 | 1 |  |  | 1 |  |
|  | Verification | 1 | 1 | 1 |  |  |  |  |  |
|  | Validation | 1 |  |  |  |  |  |  |  |
| **Functional Test** | | **PM** | **AN** | **DM** | **PR** | **GD** | **TR** | **QA** | **DO** |
|  | Setup test environment | 1 | 1 | 1 |  |  |  | 1 |  |
|  | Define Test Scenarios | 1 | 1 | 1 |  |  |  | 1 |  |
|  | Create Test Cases | 1 | 1 | 1 |  |  |  | 1 |  |
|  | Conduct Test | 1 | 1 | 1 |  |  |  | 1 |  |
|  | Verification | 1 | 1 | 1 |  |  |  |  |  |
|  | Validate | 1 |  |  |  |  |  |  |  |
|  | Release component |  |  |  |  |  |  | 1 |  |
|  | Get Sign off | 1 |  |  |  |  |  |  |  |
| **Validation Test** | | **PM** | **AN** | **DM** | **PR** | **GD** | **TR** | **QA** | **DO** |
|  | Setup test environment |  |  |  |  |  |  | 1 |  |
|  | Upload / install released component |  |  |  |  |  |  | 1 |  |
|  | Conduct Validation Test |  |  |  |  |  |  | 1 |  |
|  | Verification | 1 | 1 | 1 |  |  |  |  |  |
|  | Validation | 1 |  |  |  |  |  |  |  |
|  | Create development package |  |  |  | 3 |  |  |  |  |
|  | Deploy |  |  |  | 1 |  |  |  |  |
|  | Get Sign off | 1 |  |  |  |  |  |  |  |
| **Content Building** | | **PM** | **AN** | **DM** | **PR** | **GD** | **TR** | **QA** | **DO** |
|  | actual content build-up |  |  |  |  |  |  |  | 1 |
|  | verifying encoded content | 1 | 1 | 1 |  |  |  |  |  |
|  | Verification | 1 | 1 | 1 |  |  |  |  |  |
|  | Validate | 1 | 1 | 1 |  |  |  |  |  |
|  | Get Sign off | 1 |  |  |  |  |  |  |  |
| **Training** | | **PM** | **AN** | **DM** | **PR** | **GD** | **TR** | **QA** | **DO** |
|  | Prepare Training matterials and Equipment |  |  |  | 3 |  | 1 |  |  |
|  | Conduct User training/s |  |  |  | 1 |  | 1 |  |  |
|  | Get Feedback | 1 | 1 | 1 |  |  |  |  |  |
|  | Verification | 1 | 1 | 1 |  |  |  |  |  |
|  | Validation | 1 | 1 | 1 |  |  |  |  |  |
|  | Get Sign off | 1 |  |  |  |  |  |  |  |
| **Go Live** | | **PM** | **AN** | **DM** | **PR** | **GD** | **TR** | **QA** | **DO** |
|  | Check Website accessibility w/in APC |  |  |  | 1 |  |  |  |  |
|  | Install Compatible Browser, Plugins, Addons |  |  |  | 1 |  |  |  |  |
|  | Configure Tools |  |  |  | 1 |  |  |  |  |
|  | Install System Application |  |  |  | 1 |  |  |  |  |
|  | Verify Installation | 1 | 1 | 1 | 1 |  |  |  |  |
|  | Validate | 1 | 1 | 1 |  |  |  |  |  |
|  | Go Live | 1 |  |  |  |  |  |  |  |
|  | Get Sign off | 1 |  |  |  |  |  |  |  |

**4.2.4. Budget Allocation**

**Table 4.7 Budget Allocation**

|  |  |  |
| --- | --- | --- |
| **Conceptualizing** | | **Cost** |
| Define Project Objectives | | **3,600** |
| Define Project Objectives | | **3,600** |
| Determine requirements | | **3,600** |
| Define Business Process | | **3,600** |
| Define Project Framework | | **3,600** |
| Define Project Methodology | | **7,200** |
| Define Reports requirement definition | | **3,600** |
| Verification | | **300** |
|  | Validation | **300** |
|  | Get Sign Off | **300** |
|  | **Total Conceptualizing** | **29,700** |
| **Workflow Analysis** | | **Cost** |
|  | Define CM workflow | **24,000** |
|  | Define User types | **3,000** |
|  | Define Access privileges | **4,500** |
|  | Define Content types | **4,500** |
|  | Analysed project workflow | **34,500** |
|  | Present Project workflow for approval | **900** |
|  | Finalize Workflow | **3,600** |
|  | Verification | **1,950** |
|  | Validation | **1,950** |
|  | Get sign Off | **300** |
|  | **Total Workflow Analysis** | **79,200** |
| **Tools Selection** | | **Cost** |
|  | Identify Tools | **18,900** |
|  | Short List Appropriate Tools | **18,900** |
|  | Evaluate tools | **18,900** |
|  | Test Tools | **16,800** |
|  | Select tool | **750** |
|  | Verify tools | **750** |
|  | Validation | **750** |
|  | Get Sign Off | **300** |
|  | **Total Selection** | **76,050** |
| **System Development** | | **Cost** |
|  | Install tools | **1,550** |
|  | Configure tools | **1,550** |
|  | Content Development | **101,700** |
|  | Determine Data Flow Diagram | **33,300** |
|  | Define UML or Use cases | **43,500** |
|  | Program coding | **32,850** |
|  | Initial content build-up | **400** |
|  | Conduct Test | **6,300** |
|  | Verification | **750** |
|  | Validation | **300** |
|  | **Total System Development** | **222,200** |
| **Functional Test** | | **Cost** |
|  | Setup test environment | **4,800** |
|  | Define Test Scenarios | **4,750** |
|  | Create Test Cases | **3,800** |
|  | Conduct Test | **5,600** |
|  | Verification | **750** |
|  | Validate | **300** |
|  | Release component | **100** |
|  | Get Sign off | **300** |
|  | **Total Functional Test** | **20,400** |
| **Validation Test** | | **Cost** |
|  | Setup test environment | **400** |
|  | Upload / install released component | **400** |
|  | Conduct Validation Test | **1,800** |
|  | Verification | **750** |
|  | Validation | **300** |
|  | Create development package | **1,200** |
|  | Deploy | **200** |
|  | Get Sign off | **300** |
|  | **Total Validation Test** | **5,350** |
| **Content Building** | | **Cost** |
|  | actual content build-up | **1,800** |
|  | verifying encoded content | **750** |
|  | Verification | **750** |
|  | Validate | **750** |
|  | get sign off | **300** |
|  | **Total Content Building** | **4,350** |
| **Training** | | **Cost** |
|  | Prepare Training materials and Equipment | **1,400** |
|  | Conduct User training/s | **400** |
|  | Get Feedback | **750** |
|  | Verification | **750** |
|  | Validation | **750** |
|  | get sign off | **300** |
|  | **Total Training** | **4,350** |
| **Go Live** | | **Cost** |
|  | Check Website accessibility w/in APC | **800** |
|  | Install Compatible Browser, Plugins, Addons | **400** |
|  | Configure Tools | **400** |
|  | Install System Application | **400** |
|  | Verify Installation | **950** |
|  | Validate | **750** |
|  | Go Live | **300** |
|  | Get Sign off | **300** |
|  | **Total Go Live** | **4,300** |
|  | **GRAND TOTAL** | **445,900** |

**Table 4.8 Summary of Project Cost**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PERSONNEL** | **RATE/HR** | **# PERSONNEL** | **HOUR RENDERED** | **COST** |
| Project Manager | 500 | 1 | 315 | 157500 |
| Analyst | 400 | 1 | 306 | 122400 |
| Database Manager | 400 | 3 | 306 | 122400 |
| Programmer | 300 | 5 | 503 | 150900 |
| Graphic Designer | 200 | 1 | 660 | 132000 |
| Trainer | 100 | 1 | 4 | 400 |
| QA Analyst | 100 | 1 | 115 | 11500 |
| Documentarist | 100 | 1 | **22** | 2200 |
| **TOTAL** |  | **14** | **2231** | **699300** |

**Table 4.9 Staff Salary**

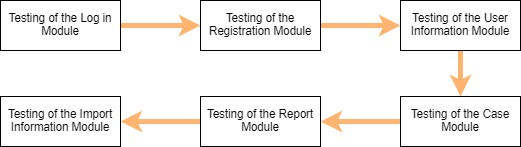
|  |  |  |
| --- | --- | --- |
| **Position** | **Position Code** | **Hourly rate** |
| Project Manager | **PM** | 300 |
| Analyst | **AN** | 250 |
| Database Manager | **DM** | 200 |
| Programmer | **PR** | 200 |
| Graphic Designer | **GD** | 150 |
| Trainer | **TR** | 100 |
| QA Analyst | **QA** | 100 |
| Documentarist | **DO** | 100 |

Table 4.9 Shows the staff position, the position code, and the corresponding hourly rate.

**4.3 Test Plan**

**4.3.1 Development Test Plan**

**4.3.1.1 Test Phases**



**Figure 4.1 Test Phases**

Figure 4.1 shows the major test phases. There will be 6 major phases in our Academic & Curricular Advising Module System. From the Login Module where the sign in, and password recovery are used. Next, we have the Registration Module where the Admin user creates new accounts. After that, we have the User Information Module which is one of the Admin's modules where account information can be updated. Another module which can be accessed by all users is the Case Module. It is a module where users can create, and view cases. Also, it is where the case notes are utilized. And we have the Reporting Module where only the Executive Director (XD) and Program Director (PD) has. It is a module where they can generate reports depending on how they want the attributes to be sorted and compiled the reports. Lastly, we have the Information Import Module where again, the XD and PD can import information from the databases: subjects, users(students), and cases.

**4.3.1.2 Test Cases and Test Scripts**

**Table 4.10 Test Cases and Test Scripts**

|  |  |
| --- | --- |
| **Test Phase 1:** | |
|  | **Description** |
| **Test Case 1.1-** Admin Login Test | This test is where the admin tests his/her login information. |
| **Test Case 1.2-** Student Login Test | This test is where the student test his/her login information. |
| **Test Case 1.3-** Adviser/Standard Faculty Login Test | This test is where the adviser or standard faculty test their login information. |
| **Test Case 1.4-** XD/PD Login Test | This test is where the XD or PD test their login information. |
| **Test Case 1.5-** Admin Login Failed Test (Wrong User ID) | This test is for the checking of the incorrect login popup in the admin login test. |
| **Test Case 1.6-** Student Login Failed Test (Wrong Password) | This test is for the checking of the popup for incorrect login using student information test account. |
| **Test Case 1.7-** Adviser/Standard Faculty Login Failed Test (Blank/Empty) | This test is for the adviser and or standard faculty incorrect login popup test. |
| **Test Case 1.8-** XD/PD Login Failed Test (Case Sensitivity) | This test is for the XD/PD incorrect login popup test. |
| **Test Phase 2:** | |
| **Test Case 2.1-** Add New Student User Test (1) | The first out of six steps in the user registration module for student users. The checking of the functionality of a quick link: Add New User |
| **Test Case 2.2-** Add New Student User Test (2) | The second out of six steps in the user registration module for student users. This is where the admin selects the account type as student and enters the new student ID. |
| **Test Case 2.3-** Add New Student User Test (3) | The third out of six steps in the user registration module for student users. Testing the button for submission. |
| **Test Case 2.4-** Add New Student User Test (4) | The fourth out of six steps in the user registration module for student users. This is where the admin enters the new student information. And the test of the button for submission. |
| **Test Case 2.5-** Add New Student User Test (5) | The fifth out of six steps in the user registration module for student users. Additional account information such as security questions and the user’s password. |
| **Test Case 2.6-** Add New Student User Test (6) | The last step out of six steps in the user registration module for student users. It tests the button for confirmation. |
| **Test Case 2.7-** Add New Faculty(Standard)/Adviser/XD/PD User Test (1) | The first out of six steps in the user registration module for Faculty(Standard)/Adviser/XD/PD users. The checking of the functionality of a quick link: Add New User |
| **Test Case 2.8-** Add New Faculty(Standard)/Adviser/XD/PD User Test (2) | The second out of six steps in the user registration module for Faculty(Standard)/Adviser/XD/PD users. This is where the admin selects the account type as Faculty(Standard)/Adviser/XD/PD and enters the new faculty ID. |
| **Test Case 2.9-** New Faculty(Standard)/Adviser/XD/PD User Test (3) | The third out of six steps in the user registration module for Faculty(Standard)/Adviser/XD/PD users. Testing the button for submission. |
| **Test Case 2.10-** Add New Faculty(Standard)/Adviser/XD/PD User Test (4) | The fourth out of six steps in the user registration module for Faculty(Standard)/Adviser/XD/PD users. This is where the admin enters the new faculty information. And the test of the button for submission. |
| **Test Case 2.11-** Add New Faculty(Standard)/Adviser/XD/PD User Test (5) | The fifth out of six steps in the user registration module for Faculty(Standard)/Adviser/XD/PD users. Additional account information such as security questions and the user’s password. |
| **Test Case 2.12-** Add New Faculty(Standard)/Adviser/XD/PD User Test (6) | The last step out of six steps in the user registration module for Faculty(Standard)/Adviser/XD/PD users. It tests the button for confirmation. |
| **Test Case 2.13-** Add New Student User Test: Case Sensitivity | This test is for the checking of the case sensitivity of the input of the user ID of a student user account. |
| **Test Case 2.14-** Add New Faculty(Standard)/Adviser/XD/PD User Test: Case Sensitivity | This test is for the checking of the case sensitivity of the input of the user ID of a Faculty(Standard)/Adviser/XD/PD user account. |
| **Test Case 2.15-** Add New Student User Test: Blank/Empty Input | This test is for the checking of the blank and empty fields in the user ID input upon entering the user ID. |
| **Test Case 2.16-** Add New Faculty(Standard)/Adviser/XD/PD User Test: Blank/Empty Input | This test is for checking the input if the fields are blank or empty. |
| **Test Case 2.17-** Invalid Student User Email Input Test (Wrong format) | This test is for the format of the email required and is checked in the algorithm of the program. A label message will popup once this test is enabled. |
| **Test Case 2.18-** Invalid Faculty(Standard)/Adviser/XD/PD Email Input Test (Does not exist) | This test is to check whether the email address exists in the email database. |
| **Test Case 2.19-** Blank input for Security Questions Test | This test is when the fields in the security questions are empty. A label message popup type will appear once the test is enabled. |
| **Test Phase 3:** | |
| **Test Case 3.1-** Update Password Test | This test is for the utilization of buttons in the update password quick link of the admin user. |
| **Test Case 3.2-** Update Security Questions and Password Test | This test is to see the transition of the security module of the admin page. Correct information is inputted in this test. |
| **Test Case 3.3-** Update Password Test: Mismatch | This test is when the admin incorrectly enters the password twice in the new password and confirm password textbox, an error input popup will be triggered once the button submit is clicked. |
| **Test Case 3.4-** Update Security Questions Test: Blank Input | This test is when the admin left a text empty in the new password and confirm password textbox, an error popup will be triggered once the button submit is clicked. |
| **Test Phase 4:** | |
| **Test Case 4.1-** Create a Case Test (Step 1) | This test is for the test of a tool button to launch the Case Module. |
| **Test Case 4.2-** Create a Case Test (Step 2) | This test is for the input of the new case information. |
| **Test Case 4.3-** Create a Case Test (Step 3) | This test is for the generation of a case number. |
| **Test Case 4.4-** Create a Case Test (Step 4) | This test is for the checking of the functionality of the Case Notes tool. |
| **Test Case 4.5-** Create a Case Test (Step 5) | This test is for the checking of the functionality of the popup of the Add Case button. |
| **Test Case 4.6-** Create a Case Test (Step 2): Blank Title | This test is for the checking for invalid input such as empty fields, etc. A label message will appear when the test is triggered. |
| **Test Case 4.7-** View Case by Case ID Test (Step 1) | This test is where the Case ID is checked for validity and existence. |
| **Test Case 4.8-** View Case by Case ID Test (Step 2) | This test is where the system is tested if it will generate a new Case ID. |
| **Test Case 4.9-** View Case by Case ID Test (Step 2): Case Sensitivity | This test is where the Case ID is checked for validity, format, and existence. |
| **Test Case 4.10-** View Case by Case ID Test (Step 2): Blank Input | This test is where the Case ID is checked for validity, format, and existence. |

|  |  |
| --- | --- |
| **Test Phase 5:** | |
| **Test Case 5.1-** View All Cases(Selected) Test | This test is for report generation depending on the specifications entered by the XD/PD. Specification: All Cases Per Faculty ID |
| **Test Case 5.2-** View All Cases Per Faculty ID Test | This test is for report generation depending on the specifications entered by the XD/PD. Specification: All Cases by Faculty ID |
| **Test Case 5.3-** View All Cases by Faculty ID Test | This test is for report generation depending on the specifications entered by the XD/PD. Specification: All Cases by Faculty ID |
| **Test Case 5.4-** View All Cases by Student ID Test | This test is for report generation depending on the specifications entered by the XD/PD. Specification: All Cases by Student ID |
| **Test Case 5.5-** View All Cases by Student ID (Can also be for Faculty ID): Case Sensitivity Test | This test is for case sensitivity test. |
| **Test Phase 6:** | |
| **Test Case 6.1-** View Case by Case ID Test (Step 1) | This is to test the importing of the cases depending on the specifications entered by a user. And exported via Generate Case button in the Report Module. |
| **Test Case 6.2-** View Case by Case ID Test (Step 2) | This test is the second step of the generation of cases for exporting. |
| **Test Case 6.3-** View Case by Case ID Test (Step 2): Case Sensitivity | This test is to check for case sensitivity in the input of the Case ID (special characters, etc.). The label message for invalid input is triggered. |
| **Test Case 6.4-** View Case by Case ID Test (Step 2): Blank Input | This test is to check for the label message if it will be triggered when not entering anything in the setting of the case specifications for export. |

**4.3.2 Functional Test Plan**

**Log in Module**

Test Case: Login

**Table 4.11 Test # 1 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected Output** |
| The user logs in his/her account using their User ID (School ID) and password. | The user will be directed to their designated homepage. |

Test Case: Forgot Password

**Table 4.12 Test # 2 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected Output** |
| The user clicks the “forgot password” button and answers the security questions | The user will be notified with a popup that their security question answers are correct and will be redirected to the enter new password page. |

Test Case: Admin creating a new account

**Table 4.13 Test # 3 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected Output** |
| Admin clicks the “create user” button and fills it with new user information. | The admin will be notified with a popup that the account is successfully created. |

Test Case: Verifying the creation of a new account

**Table 4.14 Test # 4 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected Output** |
| Usually the admin, but any user can check whether the new account that was created will show in the account/users list. | The newly created account will be shown in the users list. |

**User Information Module**

Test Case: Updating a user’s information

**Table 4.15 Test # 1 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected Output** |
| The admin selects the user he/she will update and clicks the corresponding “update information” button in the admin menu. The admin will then input the new account information. | The admin will be notified with a popup that the account is successfully updated. |

Test Case: Verifying whether the information of the user was updated

**Table 4.16 Test # 2 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected Output** |
| The admin will select the user he/she updated and check whether the newly inputted information will properly show. | Once searched by the admin what he/she updated recently, the new and correct information will show in the corresponding user’s account. |

**Case Module**

Test Case: Creating a case

**Table 4.17 Test # 1 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected Output** |
| Any faculty member will click the “create a case” button and will enter the case information along with the student involved. | A popup will notify the faculty member that their case is successfully created. |

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

**Table 4.18 Test # 2 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected 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 will show the correct information inputted by its creator. |

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

**Table 4.19 Test # 3 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected 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 will show because of the newly created case that they are involved with. |

Test Case: Viewing of the cases selected

**Table 4.20 Test # 4 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected 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 will be showed to the user who called upon the cases. |

Test Case: Adding Case Notes

**Table 4.21 Test # 5 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected 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 will show that the case notes is successfully added to the case notes thread of that 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 4.22 Test # 6 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected 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 will appear in their corresponding menus. |

**Report Module**

Test Case: Generating a report

**Table 4.23 Test # 1 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected 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 will notify the X.D./P.D. that their report is successfully generated. |

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

**Table 4.24 Test # 2 Test Instruction and Expected Output**

|  |  |
| --- | --- |
| **Test Instruction** | **Expected 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. |