­­­­

Asia Pacific College

Online Pre-Registration System

Project Documentation Submitted

To the Faculty of School of

Computer Science and Information Technology

Of

Asia Pacific College

In Partial Fulfillment of the Requirements for the subject

Applied Projects 2

By

Romulus Diego

Jairus Adrian G. Roguel

Angelica Laurene S. Ruiz

Table of Contents

[Project Title **Error! Bookmark not defined.**](#_Toc454543889)

[Abstract 3](#_Toc454543890)

[List of Figures, List of Tables, List of Notations 4](#_Toc454543891)

[I. Introduction **Error! Bookmark not defined.**](#_Toc454543892)

[1.1 Project Context **Error! Bookmark not defined.**](#_Toc454543893)

[1.2 Purpose and Description **Error! Bookmark not defined.**](#_Toc454543894)

[1.3 Objectives **Error! Bookmark not defined.**](#_Toc454543895)

[1.4 Scope and Limitations **Error! Bookmark not defined.**](#_Toc454543896)

[II. Related Literature 7](#_Toc454543897)

[III. Technical Background 9](#_Toc454543898)

[3.1 Include in-depth discussion on relevant technical aspects of this project **Error! Bookmark not defined.**](#_Toc454543899)

[IV. Design and Methodology 9](#_Toc454543900)

[4.1 Include discussions on conceptual design/system architecture/block diagrams and algorithms **Error! Bookmark not defined.**](#_Toc454543901)

[V. Results and Discussion 11](#_Toc454543902)

[VI. Conclusions and Recommendations 11](#_Toc454543903)

[VII. Appendices 11](#_Toc454543904)

# Abstract

Pre-registration is one of the many processes that APC student would have to do during his/her stay in Asia Pacific College. But the pre-registration process in APC is not the most convenient and efficient way. This project aims to provide convenient and efficient pre-registration services to the Asia Pacific College community. This project would make the current pre-registration process online to allow the students to customize the schedule automatically given to them by the system. This project would not only benefit students but also the advisers for the system would do most of their job. Advisers would only need to approve or disapprove requests that are sent by the students of the schools they are assigned to. The project would be created using the Cobalt Framework.

***Keywords:*** *pre-registration, students, advisers, online, cobalt, Asia Pacific College*

1. **Introduction**
   1. Project Context

Pre-registration inside a school or university is a process of registering in advance a subject that students need to add or drop in their load. There are also rules applied for pre-registering to avoid conflicts like clashing of schedules and inconsistencies in subjects. In Asia Pacific College, a manual pre-registration system exists wherein students can request for an addition or dropping of subject. Basically, students fill in a form and is sent to an adviser for approval. The steps in pre-registering may cause inconvenience because students must go to school just to pre-register, instead of just being one click away.

Thus, the project members proposed an online pre-registration system to provide convenience and satisfaction to the main users which are students and advisers. The proposed system aims to fulfill the satisfaction of the students because they greatly affect all aspects of the school. The system’s main feature is to add a subject wherein students can choose by using the dropdown of subject codes provided and submitted as a request. But, there is a rule provided inside the system which is the pre-requisites of subjects. For example, when a student requests for a subject that has a pre-requisite subject that he or she failed or not yet taken, the subject requested cannot be taken where the system notifies the student. Another feature is for advisers in which requests sent by students are directed to them for approval or disapproval and

Users can access APC’s Flavio website www.[apc.edu.ph/flavio](http://apc.edu.ph/flavio), where the pre-registration is located along with Flavio’s other features. Authentication (username and password) is required to log in and view other systems like Online Teachers Evaluations and Freshman to Junior Evaluation. Once logged in, users can proceed to pre-registration.

Presently, the proposed system is in the form of a prototype in which not all features envisioned by the project members are introduced. Once the system is ready, it would be integrated with APC’s Flavio system, a system used by the APC community and is accessible as long as it is opened in the school’s server. The proposed system is intended to be accessible outside the school and according to APC’s Information Technology Resources Office (ITRO), Flavio would soon be accessible outside the school.

* 1. Purpose and Description

Students are highly prioritized in APC for they have enrolled in order to receive the services that the school can offer like education, computer and science laboratories, libraries, and cafeterias. But more to that is what the school can offer when there are processes to be made like enrollment, petition, and pre-registration. Currently, the process of pre-registration is manual. The current system may cause inconvenience for a student because there can be conflicts in the schedule of an adviser, crowded offices and waiting time.

The main goal of the proposed system is to provide students a service wherein most inconveniences are eliminated thus giving them the satisfaction and convenience they need. The system will push the potential of APC’s online services and would equal to Philippine schools’ online systems.

The online pre-registration system provides a convenient process to add or drop a subject in a student’s load. Adding a subject is simplified because the system already has a Flowchart database table that identifies the subjects of a student from the first term to the last. It also provides all subject offerings available so that students can easily view what are the subjects they need to take because they have not taken it yet or failed it. Dropping a subject would be just like how the adding function works. In the side of an adviser/approver, processes are also simplified because all requests of students are directed to advisers filtered by the students and advisers’ department. For example, a CSIT student’s request is directed to a CSIT adviser/approver for approval.

* 1. Objectives

The objectives of the project are:

* To understand the flow of the current pre-registration process
* To create a process flow of the proposed online pre-registration system
* To build a prototype of the online pre-registration system
* To provide a service that will offer convenience and satisfy students and advisers
* To collect data from students about their feedback on the current pre-registration system
* To look for algorithms that may serve as a basis for working on the project
  1. Scope and Limitations

The proposed system has a feature for adding or removing subjects wherein the students send it as request to their respective advisers who approves or disapproves the request**.** But as of now, the existing features created were the adding of subject and approval of requests, there is no dropping of subjects yet. Although the proposed system is not yet final, the notifications of approvals of the advisers are envisioned to be sent through email.

Authentication is required in which username and password are used. Also, setup tables are provided for an administrator to add, edit and delete data from the database tables. When implemented, errors due to codes will be examined by the team while overloading of database will be examined by the ITRO. Currently, the system is not yet applied in the school’s network since it is still in the proposal/development stage.

When the application development team of ITRO implements that Flavio is accessible outside the school, the pre-registration system can also be accessed outside. When it is not implemented, users can access it inside the school given that there is an internet connection whether intranet or wireless fidelity provided by APC.

Also, more work can be done in the database tables because there can be added, changed or deleted tables and data as the development progresses. A flowchart is provided but as of now, it is not yet illustrated and does not contain data about grades has three colored indicators which are blue for not yet taken, white for passed and red for repeat or authorized withdrawal. To post added subjects to a student’s assessment form is not yet tackled. The course table, which handles the different courses of students is presently constant, meaning that records of students that may have shifted into another course are not yet counted and its record history is non-existent.

# Related Literature

To help develop the system, the team gathered ideas on how to develop a pre-registration system by gathering information from other school’s systems. The team decided to search for schools who are known for their excellency in education and started gathering procedures on how their current system works. The gathered data are used to compare the similarities of the chosen schools to the proposed system for a clear understanding so that the system will have better functionalities and user experience. Nowadays, schools are automating processes in their current systems and having an online system is necessary.

The adaptation of these related systems has been a big help to the team, it gave them the thought of how important and how beneficial it would be for the school. Hence, the project’s goal is to provide convenience for the APC Community and help the school to have a better and hassle-free experience of pre-registering. The team believes that having these ideas gathered together will help develop a well-functioning and user-friendly online pre-registration system.

* 1. Ateneo Integrated Student Information System (AISIS)

The Ateneo Integrated Student Information System (AISIS Online) is the portal for Ateneo students, faculty, and staff. Through the AISIS Online, officially enrolled Ateneo students may view pertinent school information including their IPS, grades, class schedules and the like. Students may also eventually enlist using AISIS Online. Ateneo faculty and staff with access to AISIS, on the other hand, may submit grades and access their class schedules from outside the campus.

Our project is similar in a sense that our system would like to use the web to provide the user information concerning his/her subjects that be modified. AISIS Online provides an Online Enlistment service that would give the students the convenience of modifying their subjects online. One of our project’s objectives is to provide convenience and satisfy the students, advisers, and APC Community. With this related literature, we can adopt or implement ideas that we have reviewed in the system and implement it to our proposed system. (Retrieved on August 19, 2016 from <http://aisisonline.ateneo.edu/index.php>)

* 1. De La Salle University Animo.sys Portal

De La Salle University has an online enrollment system which is Animo.sys portal that students can access for encoding the courses they want to take for the next term. Enrollment schedules are posted for students to know when to enroll. All students must have an activated My.LaSalle account to access the enrollment system open from 8 am until 8 pm from Mondays to Fridays in http://my.dlsu.edu.ph/. They must arrange all clearances first before or during the enrollment schedule. Students may have academic advising from the Graduate Program Coordinator of each course who is discussing about the subjects that students want to add. Then, students can check the course codes of the subjects they want to take. Schedules are given for students on when to enroll online. After enrolling, they may claim the Enrollment Assessment Form (EAF) which contains all the subjects they have encoded.

The features that Animo.sys Portal has some features similar to the manual pre-registration system of Asia Pacific College. For example, both systems provide course codes for students to know what subjects they want to add. Another is that both have academic advising for students to be notified whether they can take the subject or not due to different cases like failed pre-requisites, unfinished clearances and unpaid tuition fees. By having a background of how Animo.sys Portal works, it will be a great basis for creating our proposed project which is Asia Pacific College Online Pre-Registration System. (Retrieved on August 19, 2016 from https://my.dlsu.edu.ph/faq/, <http://www.dlsu.edu.ph/offices/registrar/enrollment.asp>)

* 1. University of the Philippines Computerized Registration System (UP CRS)

The Electrical and Electronics Engineering Institute in the University of the Philippines made an explanation about how the University of the Philippines’ Computerized Registration System (UP CRS) works. Students usually go through advanced enlistment during general registration. They also have advanced enlistment in the middle of the period. During advanced enlistment, students would submit a list of their desired subjects to their respective colleges. These lists are submitted online through the UP CRS. The UP CRS then processes these lists according to the student’s enrollment status. The results are to be printed as the UP Form 5-A. During the registration period, the subjects can be added or dropped from the Form 5-A due to different reasons. Then the different colleges would allocate new slots to those subjects that have been dropped by the students. These slots can be viewed in the UP CRS.

The UP CRS has the flexibility we would like our Pre-registration to have. The UP CRS has an option to add, drop and petition subjects online. The UP CRS can also show available slots to all subjects offered just like what APC has which is the Masters List. The UP CRS would be a great basis on what functionalities our Pre-registration system would have. (Retrieved on August 19, 2016 from <http://www.eee.upd.edu.ph/academic-programs/enlistment/computerized-registration-system>)

# Technical Background

Flavio is a system divided into two parts which are the main system and the Online Inquiry that are used by APC students and faculty. It was created with the use of Cobalt and since the proposed project is to be integrated with Flavio, the team members decided to use Cobalt Mark V, a code-generator framework which eliminates repetitive tasks and automates the add, edit, view and delete modules.

Cobalt uses XAMPP, a cross-platform web server that includes the Apache and MySQL modules which are used for the framework to run. The programming languages used are Hyper Text Markup Language (HTML) and Cascading Style Sheets (CSS) for the design and Hypertext Preprocessor (PHP) and MySQL for the logic. The project team members use Atom, a cross-platform source code editor to have an organized and proper platform for coding.

The hardware requirements of Cobalt were followed for the proposed system. Laptops and computers with any operating system that can use any web-server for PHP can be used to access the system.

When implemented, APC students can access the system by logging in to https://www.apc.edu.ph/flavio using their existing accounts. Flavio accounts are created at the start of their stay at APC.

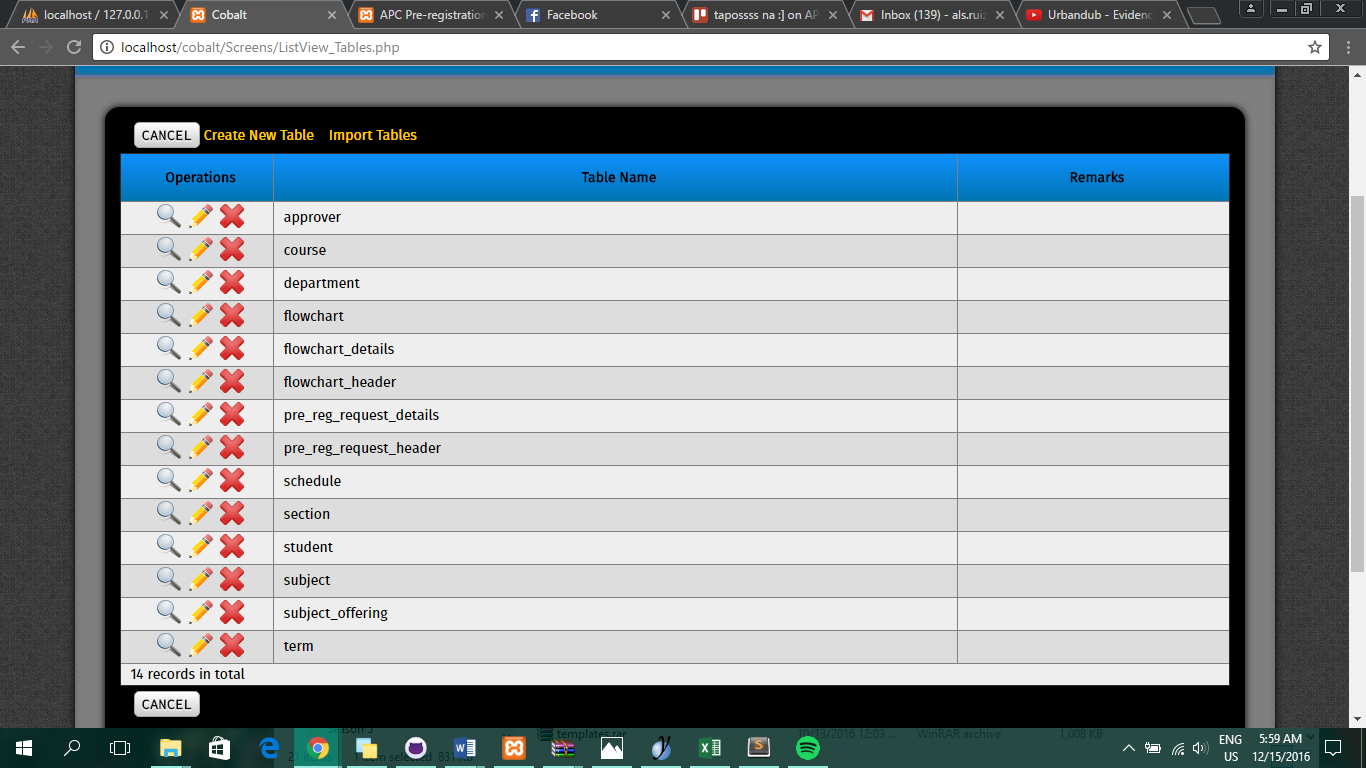
# Design and Methodology

To create the prototype of the proposed system, Cobalt was used for the front-end and the back-end. The web-server XAMPP was used all throughout the development because Cobalt requires a localhost server with the Apache and MySQL modules. The prototype was ran in the team’s preferred browser Google Chrome, but is also accessible in other browsers. The graphical user-interface was created using Cobalt’s built-in HTML designs and skins.

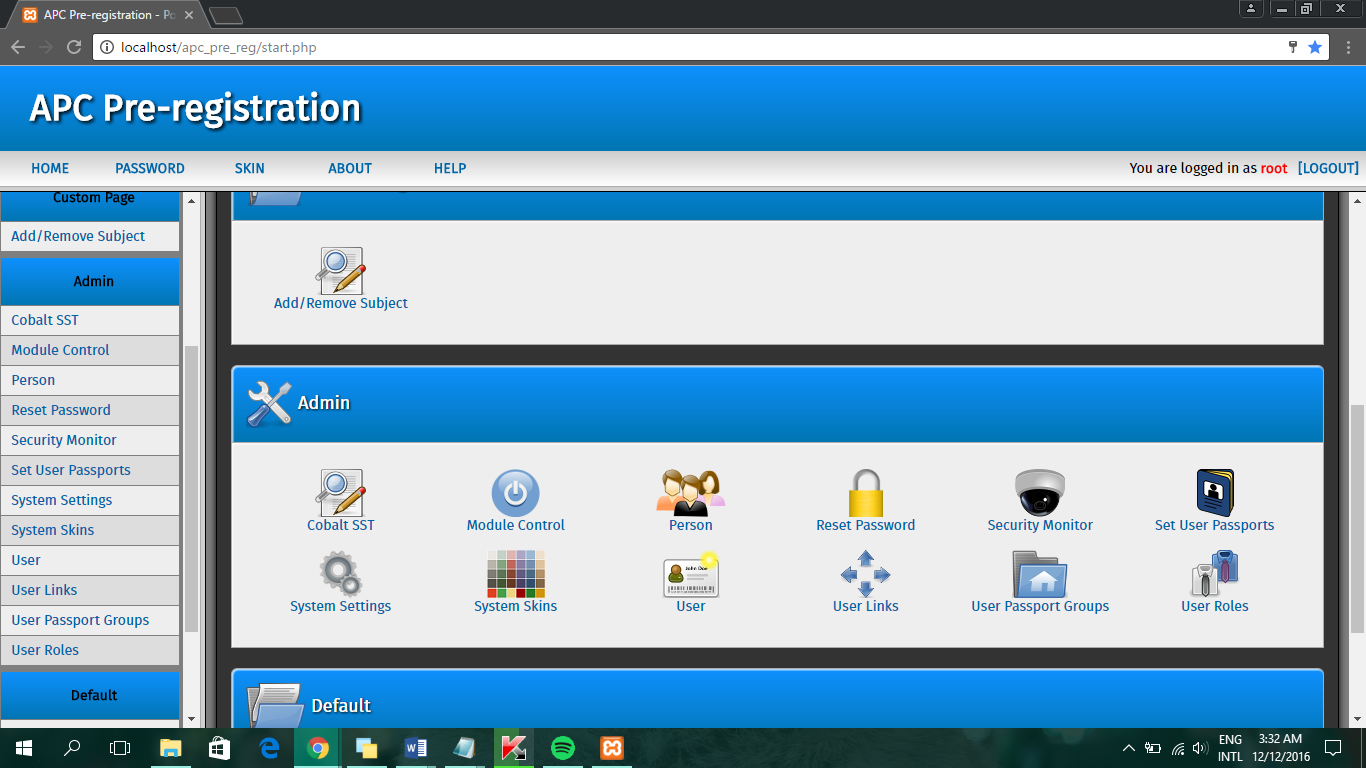
For the database, phpMyAdmin was used which is accessible by running XAMPP. All tables and data needed for the prototype was stored inside the database. Data are pre-created for presentation purposes in order to visualize the process of the system clearly. Once the proposed system is implemented, it would be integrated with Flavio, therefore the data from Flavio would also be used in the pre-registration system.

The system is well-secure because of Cobalt’s built-in security protocols. Passwords of any user are stored as a hash so that even when the database is accessed, the passwords are not seen as it is. Also, every log in a user does, a session is created and if a user logged out, the session ends so that even if there are other pages he or she opened, once the pages are reloaded, it would go back to the login page wherein the session expires. With that being defined, the system is secured from any threats especially hacking and infiltrations.

Current Tables:



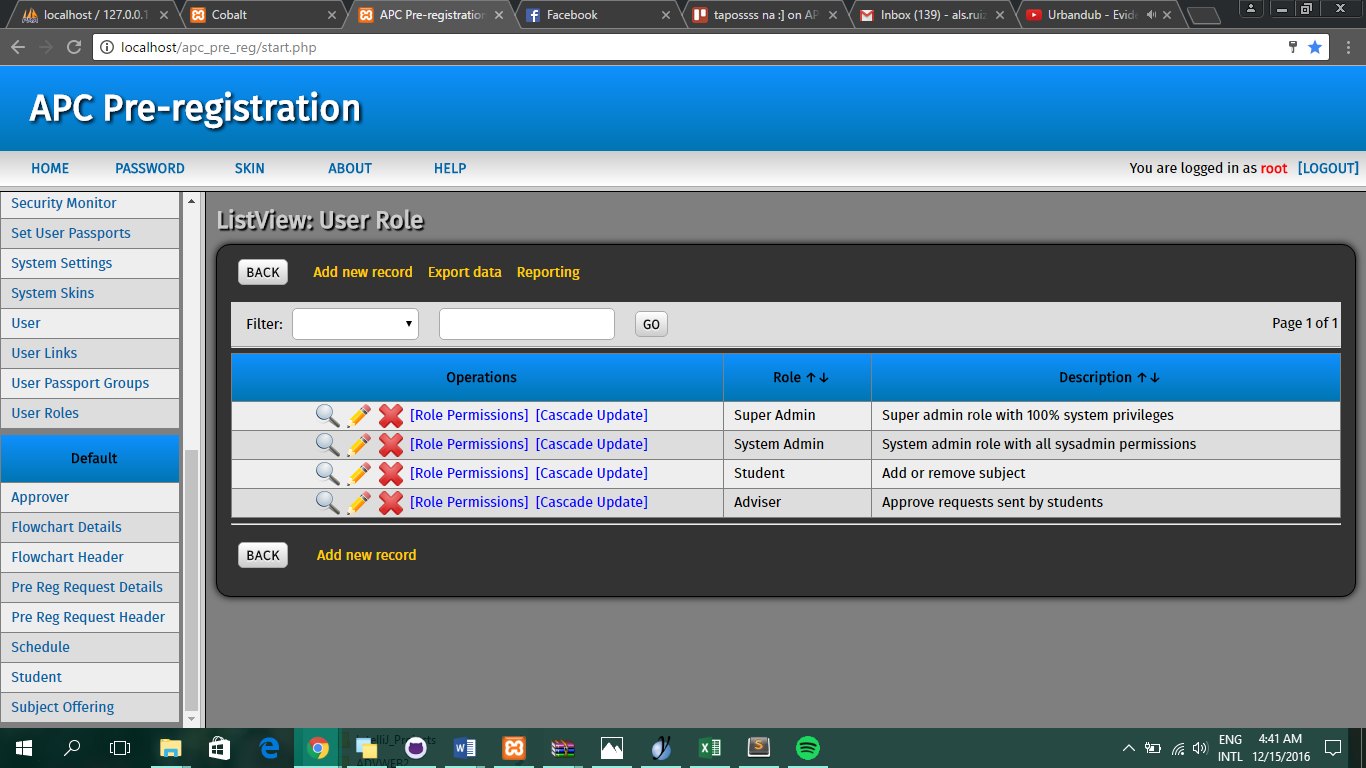
*Figure 1. Tables inside Cobalt*



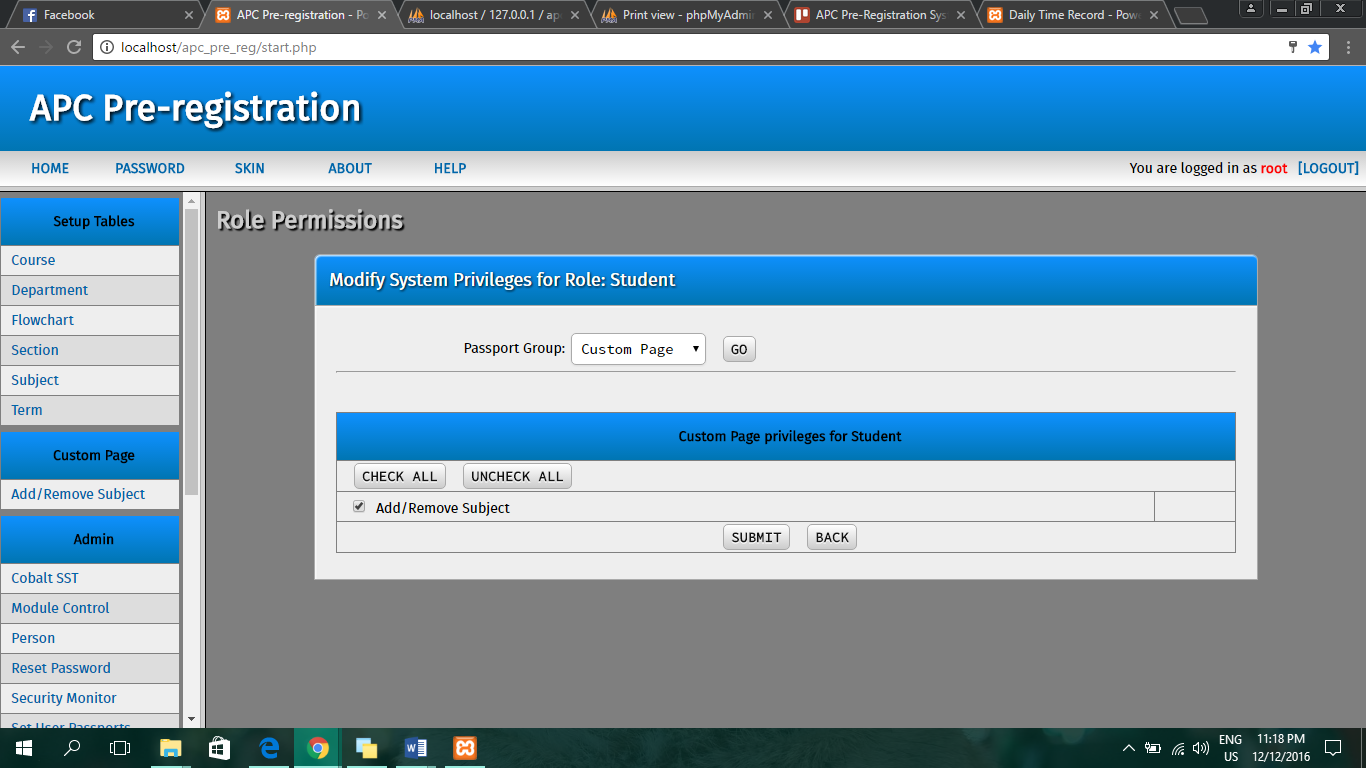
*Figure 2. Built-in tables inside Cobalt*

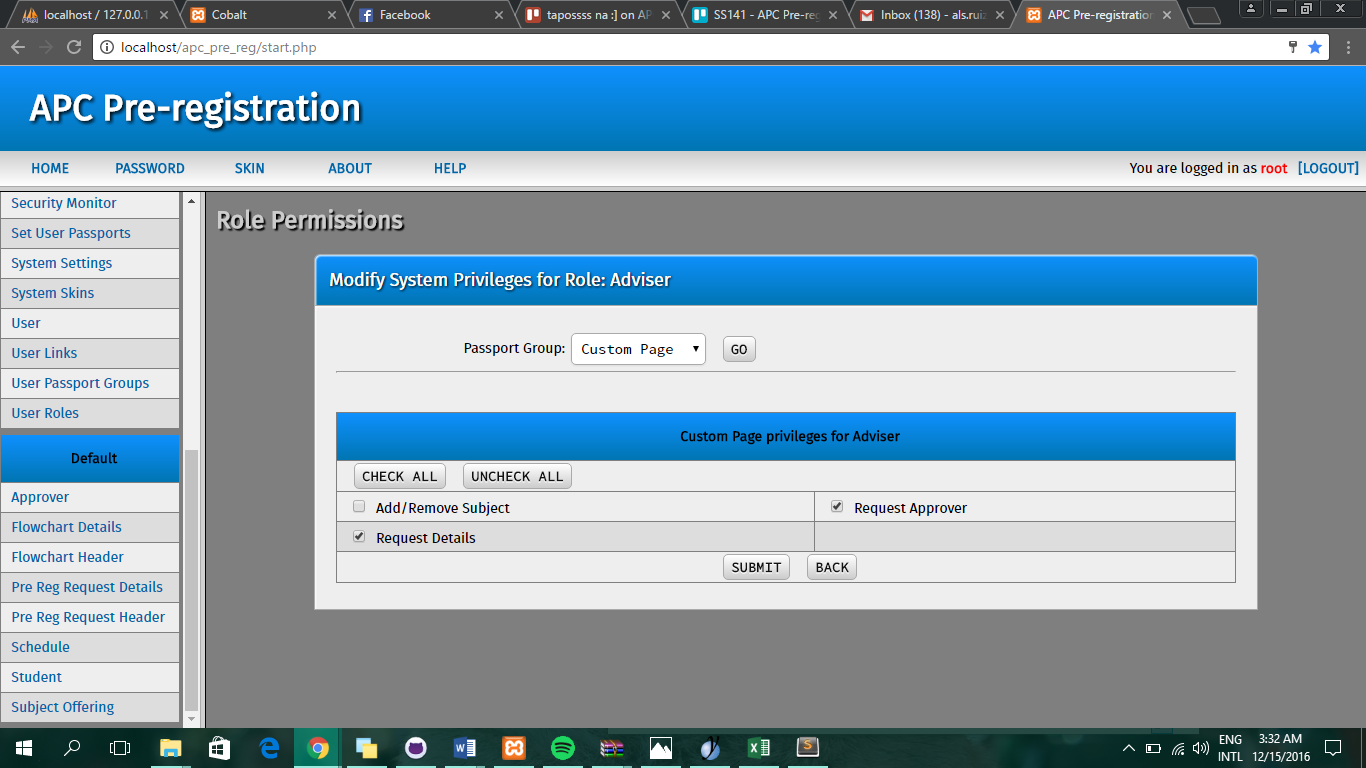
The users of the system are the administrator, students and advisers. The administrator has full control in the system wherein he or she can add, view, edit/update and delete data from the tables in Figure 1 and in Figure 2, which are incorporated with Cobalt.

The two main users are the students and advisers. The role of a student in the system is to add or remove a subject and send it as a request which is directed to the student’s respective adviser. The role of an adviser/approver is to receive and confirm the student’s request.



*All User Roles*

*User Role of Student*

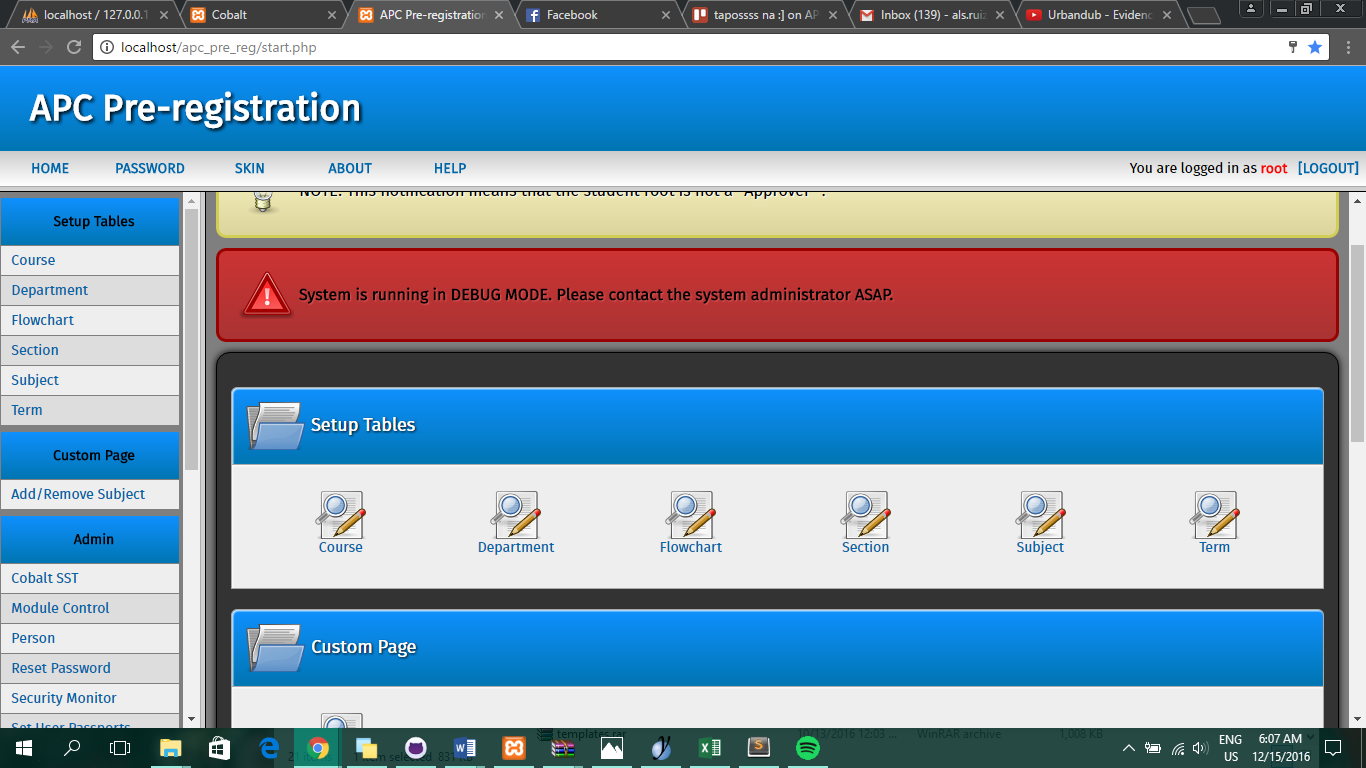


*User Role of Adviser/Approver*

# Results and Discussions

After the research, the team members started to develop a prototype for the online pre-registration system. The database was highly prioritized for it is essential for the whole system, thus it was ensured that the database would be as neat and organized as possible and data are centralized. The current database and its tables are seen in Figure 2. With the database, the creation of prototype was made possible. Setup tables as seen in Figure 3, are provided in order to show pre-created data in order to present the pre-registration process with data. Currently, the prototype functions with one process finished which is to add a subject then approve a subject.

Based on the gathered data, a high percentage of APC students state that there is a need for an online service of pre-registration because of their different reasons as seen in Figure 4. The data proved that the proposed online pre-registration system is a workable project and is a service that can satisfy students.



*Figure 3. Setup Tables*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Course | Year | Are you satisfied w/ the current pre-reg? | Comments/Recommendations/Suggestions/Violent Reactions: | Rate  (1-lowest, 5-highest) |
| *Hidden* | BSCS-SS | 3rd Yr College | Very unsatisfied | Please fix proper scheduling of encoder. | **2** |
| *Hidden* | BSCS-SS | 3rd Yr College | Unsatisfied | Sayang oras dahil madami pang pinupuntahan | **2** |
| *Hidden* | BSCS-SS | 3rd Yr College | Unsatisfied | Time is wasted because of the manual system,last time I registered a subject, it took about 1 and a half hour. | **1** |
| *Hidden* | BSCS-SS | 3rd Yr College | Unsatisfied | Ang tagal magsimula at matapos, then ang haba pa ng pilam sobrang unorganized tuwing pre-reg | **2** |
| *Hidden* | BSCS-SS | 3rd Yr College | No | Nagrereklamo yung ibang mga prof na nakakakita sa haba ng pila ng pre-reg. Dapat sisihin nila yung sistema, hindi ung mga estudyanteng gusto makakuha ng subject na gusto nila. | **1** |
| *Hidden* | BSCS-SS | 3rd Yr College | No | Hassle yung prereg nila kasi pupunta pa sa 4th floor para magprepreg, aapproach pa natin personally yung mga designated prof na magaapprove. Massuggest ko na tayo na mismong students mag add ng subjects ng online. | **2** |

*Figure 4. Snippet of feedback from APC Students*

# Conclusions and Recommendations

The research conducted along with the development of the system proved that an online pre-registration system is attainable. Based on the results, there is a high demand for an online system in pre-registering. Currently, the project’s feature is the requesting of students for subjects by choosing and adding it to their load. Those requests are sent to their respective advisers for approval. Once the request is verified, the students are notified whether their requests are approved or not. Also, students can view their flowchart to see what are their subjects.

More work can be done in order to optimize the system and by that, the project members recommend to future researchers to include a module wherein students can view their registration forms with the pre-registered subjects. In addition, a module for showing the schedule of pre-registration periods can be useful for students to be notified and lessen problems regarding schedules. A proper way to use the pre-registration would be

# Appendices

## Relevant Source Code

1. **main.php**

<?php

require\_once 'path.php';

init\_cobalt('ALLOW\_ALL',FALSE);

$html = new html;

$html->draw\_header('Welcome to your Control Center', $message, $message\_type, FALSE);

if(ENABLE\_SIDEBAR)

{

echo '

<script>

if (top.location == location)

{

window.location.replace("start.php");

}

</script>

';

}

/\*-------------------------------------------------------------------------

CUSTOMIZE FOR NOTIFICATION EACH ROLE IN THE SYSTEM.

-------------------------------------------------------------------------\*/

if($\_SESSION['role'] == "Student" || $\_SESSION['role'] == "Super Admin")

{

$html->display\_tip("NOTE: This notification means that the student {$\_SESSION['user']} is not a \"Approver\" .");

}

else

{

require\_once 'subclasses/pre\_reg\_request\_header.php';

$dbh\_pre\_reg = new pre\_reg\_request\_header;

$dbh\_pre\_reg = cobalt\_load\_class('pre\_reg\_request\_header');

$dbh\_pre\_reg->execute\_query("SELECT pre\_reg\_request\_header.is\_approved

FROM pre\_reg\_request\_header

WHERE pre\_reg\_request\_header.is\_approved = ''");

$result = $dbh\_pre\_reg->result;

$form\_status = array();

while($row = $result->fetch\_assoc())

{

$form\_status[] = $row;

}

for($x = 0; $x<count($form\_status); ++$x)

{

extract($form\_status[$x]);

}

init\_var($is\_approved);

if($is\_approved == '')

{

$html->display\_tip('You have some Request Forms to approve <a href = "modules/custom/request\_approver.php">Click Here </a>.');

}

elseif ($is\_approved == NULL)

{

$html->display\_info('There are no application requests');

}

}

/\*-------------------------------------------------------------------------

NOTHING FOLLOWS

-------------------------------------------------------------------------\*/

if(DEBUG\_MODE)

{

$html->display\_error('System is running in DEBUG MODE. Please contact the system administrator ASAP.');

}

$menu\_links = '';

$data\_con = new data\_abstraction;

$data\_con->set\_fields('a.link\_id, a.descriptive\_title, a.target, a.description, c.passport\_group, a.icon as link\_icon, c.icon as `group\_icon`');

$data\_con->set\_table('user\_links a, user\_passport b, user\_passport\_groups c');

$data\_con->set\_where("a.link\_id=b.link\_id AND b.username='" . quote\_smart($\_SESSION['user']) . "' AND a.passport\_group\_id=c.passport\_group\_id AND a.show\_in\_tasklist='Yes' AND a.status='On'");

$data\_con->set\_order('c.priority DESC, c.passport\_group, a.priority DESC, a.descriptive\_title');

if($result = $data\_con->make\_query()->result)

{

while($data = $result->fetch\_assoc())

{

extract($data);

$menu\_links[$passport\_group]['title'][] = $descriptive\_title;

$menu\_links[$passport\_group]['target'][] = $target;

$menu\_links[$passport\_group]['link\_id'][] = $link\_id;

$menu\_links[$passport\_group]['description'][] = $description;

$menu\_links[$passport\_group]['link\_icon'][] = $link\_icon;

$menu\_links[$passport\_group]['group\_icon'][] = $group\_icon;

}

$result->close();

}

else die("Fatal error: cannot retrieve modules");

unset($data\_con);

if(isset($\_SESSION['control\_center\_columns']) && $\_SESSION['control\_center\_columns'] > 0)

{

$columns\_per\_row = $\_SESSION['control\_center\_columns'];

}

elseif(defined('CONTROL\_CENTER\_COLUMNS'))

{

$columns\_per\_row = CONTROL\_CENTER\_COLUMNS;

}

else

{

$columns\_per\_row = 3; //just an arbitrary default value based on historical Cobalt setting

}

$cntr\_limit = $columns\_per\_row - 1; //subtraction needed due to 0-based counter

$column\_width = (100 / $columns\_per\_row);

$current\_group='';

$cntr=0;

if(is\_array($menu\_links))

{

$target\_frame='';

if(ENABLE\_SIDEBAR)

{

$target\_frame = 'target="content\_frame"';

}

echo '<fieldset class="container">';

foreach($menu\_links as $group => $link\_info)

{

if($current\_group=='')

{

$current\_group = $group;

menuGroupWindowHeader($group, $link\_info['group\_icon'][0]);

}

$num\_links = count($link\_info['title']);

for($a=0; $a<$num\_links; ++$a)

{

if($current\_group!= $group)

{

echo '</tr></table></div>';

$cntr=0;

menuGroupWindowFooter();

menuGroupWindowHeader($group, $link\_info['group\_icon'][$a]);

$current\_group = $group;

}

if($cntr==0)

{

echo '<div class="container\_icons\_CC">';

echo '<table width = "100%">';

echo '<tr>';

}

elseif($cntr > $cntr\_limit)

{

echo '</tr></table>';

echo '</div><div class="container\_icons\_CC">';

echo '<table width = "100%">';

echo '<tr>';

$cntr = 0;

}

++$cntr;

echo '<td width="' . $column\_width . '%" valign="top">

<a href="/' . BASE\_DIRECTORY . '/' . $link\_info['target'][$a] . '" $target\_frame class="linkCC">

<img src="images/' . $\_SESSION['icon\_set'] . '/' . $link\_info['link\_icon'][$a] . '"><br>' . $link\_info['title'][$a] . '

</a>

</td>';

}

//Just to be sure we have three columns before closing the table

for($z = $cntr; $z<=$cntr\_limit; ++$z)

{

echo '<td width="'. $column\_width . '%"> &nbsp; </td>';

}

}

echo '</tr></table></div>';

echo '</fieldset>';

}

else

{

$html->display\_error("You have no Control Center privileges in your account. Please contact your system administrator.");

}

menuGroupWindowFooter();

function menuGroupWindowHeader($group, $icon)

{

echo '<fieldset class="top">';

echo "<img src='images/" . $\_SESSION['icon\_set'] . "/$icon'> $group";

echo '</fieldset>';

echo '<fieldset class="middle">';

}

function menuGroupWindowFooter()

{

echo '</fieldset>';

}

$html->draw\_footer();

1. **login.php**

<?php

require\_once 'path.php';

init\_cobalt();

if(isset($\_SESSION['logged']) && $\_SESSION['logged'] == 'Logged')

{

redirect('start.php');

}

if(isset($\_GET['reason']))

{

if($\_GET['reason'] == 'ipchange')

{

$error\_message = 'You have been logged out because your IP address has changed. Please log in again.';

}

}

if(xsrf\_guard())

{

init\_var($\_POST['btnSubmit']);

if($\_POST['btnSubmit'])

{

require 'password\_crypto.php';

$error\_message = '';

extract($\_POST);

//Deal with passwords longer than MAX\_PASSWORD\_LENGTH (possible DoS vulnerability)

if(strlen($password) > MAX\_PASSWORD\_LENGTH)

{

//Reset password to an arbitrarily small string, thwarting any DoS attempt

$password = 'x';

}

$data\_con = new data\_abstraction;

$mysqli = $data\_con->connect\_db()->mysqli;

$clean\_username = $mysqli->real\_escape\_string($username);

$clean\_password = cobalt\_password\_hash('RECREATE', $password, $username);

//FIXME: remember to update this ancient code to use prepared statement

$mysqli->real\_query("SELECT `username`, `skin\_id`, `role`, `first\_name`, `middle\_name`, `last\_name`

FROM `user`, `user\_role`, `person`

WHERE `username`='$clean\_username'

AND `password`='$clean\_password'

AND `user`.`role\_id` = `user\_role`.`role\_id`

AND `user`.`person\_id` = `person`.`person\_id`");

if($result = $mysqli->use\_result())

{

if($data = $result->fetch\_assoc())

{

$result->close();

extract($data);

$\_SESSION['logged'] = 'Logged';

$\_SESSION['user'] = $username;

$\_SESSION['role'] = $role;

$\_SESSION['first\_name'] = $first\_name;

$\_SESSION['middle\_name'] = $middle\_name;

$\_SESSION['last\_name'] = $last\_name;

$\_SESSION['ip\_address'] = get\_ip();

$data\_con = new data\_abstraction;

$data\_con->set\_fields('skin\_name, header, footer, master\_css, colors\_css, fonts\_css, override\_css, icon\_set');

$data\_con->set\_table('system\_skins');

$data\_con->set\_where("skin\_id=?");

$bind\_params = array('i', $skin\_id);

$data\_con->stmt\_prepare($bind\_params);

$data\_con->stmt\_fetch('single');

if($data\_con->num\_rows==1)

{

extract($data\_con->dump);

$\_SESSION['header'] = $header;

$\_SESSION['footer'] = $footer;

$\_SESSION['skin'] = $skin\_name;

$\_SESSION['master\_css'] = $master\_css;

$\_SESSION['colors\_css'] = $colors\_css;

$\_SESSION['fonts\_css'] = $fonts\_css;

$\_SESSION['override\_css'] = $override\_css;

$\_SESSION['icon\_set'] = $icon\_set;

if(trim($\_SESSION['icon\_set'] == ''))

{

$\_SESSION['icon\_set'] = 'cobalt';

}

}

$data\_con->close\_db();

log\_action('Logged in');

//check if user must rehash his password due to updated method or work factor/iterations

if(cobalt\_password\_must\_rehash($username))

{

$hashed\_password = cobalt\_password\_hash('NEW',$password, $username, $new\_salt, $new\_iteration, $new\_method);

$data\_con = new data\_abstraction;

$data\_con->set\_query\_type('UPDATE');

$data\_con->set\_table('user');

$data\_con->set\_update("`password`=?, `salt`=?, `iteration`=?, `method`=?");

$data\_con->set\_where("username=?");

$bind\_params = array('ssiss', $hashed\_password, $new\_salt, $new\_iteration, $new\_method, $username);

$data\_con->stmt\_prepare($bind\_params);

$data\_con->stmt\_execute();

}

redirect('start.php');

}

else $error\_message = "Check username and password.";

}

else die($mysqli->error);

$data\_con->close\_db();

}

}

$html = new html;

?>

<!DOCTYPE html>

<html>

<head>

<script>

if (top.location != location)

{

top.location.href = document.location.href ;

}

</script>

<title> <?php echo GLOBAL\_PROJECT\_NAME;?> - Powered by Cobalt</title>

<link href="css/login.css" rel="stylesheet" type="text/css">

<meta http-equiv="content-type" content="text/html; charset=<?php echo MULTI\_BYTE\_ENCODING; ?>" />

</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0" onload="document.getElementById('username').focus();">

<?php

echo '<form method="POST" action="' . basename($\_SERVER['SCRIPT\_NAME']) . '">';

$form\_key = generate\_token();

$form\_identifier = $\_SERVER['SCRIPT\_NAME'];

$\_SESSION['cobalt\_form\_keys'][$form\_identifier] = $form\_key;

echo '<input type="hidden" name="form\_key" value="' . $form\_key .'">';

?>

<div class="left\_container">

<div class="title">

<?php echo GLOBAL\_PROJECT\_NAME;?>

</div>

<div class="flavor\_text">

<?php

$enable\_flavor\_text = TRUE;

require 'login\_flavor\_text.php';

?>

</div>

</div>

<div class="right\_container">

<fieldset class="right\_panel">

<table border="0" width="100%" cellspacing="1">

<tr>

<td align="center">

<img src="images/login\_badge.png">

</td>

</tr>

<tr>

<td align="center">

<?php $html->draw\_text\_field('','username',FALSE,'text',FALSE, 'id="username" size="37" autocomplete="off" placeholder="Username"'); ?>

</td>

</tr>

<tr>

<td align="center">

<?php $html->draw\_text\_field('','password',FALSE,'password',FALSE,'maxlength="' . MAX\_PASSWORD\_LENGTH . '" size="37" autocomplete="off" placeholder="Password"'); ?>

</td>

</tr>

<tr>

<td align="center">

<input type=submit value="LOG IN" name="btnSubmit">

<?php

init\_var($error\_message);

$\_SESSION['icon\_set'] = 'cobalt';

$html->display\_error($error\_message);

?>

</td>

</tr>

</table>

</fieldset>

</div>

</form>

</body>

</html>

1. **add\_subject.php**

<?php

require 'path.php';

init\_cobalt('Add subject');

require\_once 'subclasses/subject\_html.php ';

require\_once 'subclasses/subject.php ';

$student = cobalt\_load\_class('student');

$request = cobalt\_load\_class('pre\_reg\_request\_details');

$header = cobalt\_load\_class('pre\_reg\_request\_header');

$html = new subject\_html;

init\_var($\_POST['code\_subject']);

init\_var($\_POST['list\_section']);

init\_var($\_POST['btn\_submit']);

init\_var($\_POST['btn\_go']);

init\_var($\_POST['btn\_add']);

init\_var($\_POST['btn\_remove']);

init\_var($\_POST['id']);

init\_var($\_POST['btn\_remove\_all']);

$code\_subject = ($\_POST['code\_subject']);

$list\_section = ($\_POST['list\_section']);

$query\_code = "SELECT subject\_id AS id\_subject, subject\_code FROM subject WHERE is\_active ='Yes' ";

$list\_value\_code = 'id\_subject';

$list\_items\_code = array('subject\_code');

$query\_section = "SELECT section\_id AS id\_section, section\_name FROM section WHERE is\_active ='Yes' ";

$list\_value\_section = 'id\_section';

$list\_items\_section = array('section\_name');

$dbh\_subject = new subject;

$object\_name = 'dbh\_subject';

if($\_POST['btn\_remove\_all'])

{

unset($\_SESSION['arr\_container']);

}

if($\_POST['code\_subject'] == $code\_subject && $\_POST['list\_section'] == $list\_section )

{

$query = "SELECT subject.subject\_code,

subject.subject\_name,

subject.subject\_unit,

subject.subject\_id,

section.section\_id,

section.department\_id,

section.course\_id,

section.section\_name,

schedule.schedule\_id,

schedule.schedule\_day,

schedule.schedule\_from,

schedule.schedule\_to

FROM subject\_offering

LEFT JOIN subject

ON subject.subject\_id = subject\_offering.subject\_id

LEFT JOIN section

ON section.section\_id = subject\_offering.section\_id

LEFT JOIN schedule

ON schedule.schedule\_id = subject\_offering.schedule\_id

WHERE subject.subject\_id LIKE '$code\_subject'

AND section.section\_id LIKE '$list\_section'

";

}

if(isset($\_GET['notice']))

{

$notice = TRUE;

}

else

{

$notice = FALSE;

}

if($notice)

{

$html->display\_message('Pre-Registered Subjects are submitted, please wait for the adviser to review it.');

}

if($\_POST['code\_subject'] == '' || $\_POST['list\_section'] == '')

{

// $html->display\_error("Please select a Subject followed by a Section.");

$query= "SELECT subject\_offering\_id FROM subject\_offering WHERE subject\_offering\_id = '0'";

}

$result = $dbh\_subject->execute\_query($query)->result;

$arr\_results = array();

// debug($dbh\_subject->query);

$row = $result->fetch\_assoc();

if($\_POST['btn\_add'])

{

if($row != NULL)

{

if(!isset($\_POST['counter']))

{

$counter = 0;

}

else

{

$counter = $\_POST['counter'];

}

$\_SESSION['arr\_container'][$counter] = $row;

$counter++;

}

}

// debug($row);

if($\_POST['btn\_remove'])

{

$row\_number = $\_POST['id'];

array\_splice($\_SESSION['arr\_container'],$row\_number,1);

}

// debug($\_POST);

// $\_SESSION['arr\_container'] = "";

// debug($\_SESSION['arr\_container']);

if(isset($\_SESSION['arr\_container']))

{

$arr\_results = $\_SESSION['arr\_container'];

$submit\_true = TRUE;

}

else {

$submit\_true = FALSE;

}

// debug($student->get\_student\_id\_by\_username($\_SESSION['user']));

if($\_POST['btn\_submit'])

{

// debug($arr\_results);

require\_once 'subclasses/pre\_reg\_request\_header.php';

$dbh\_pre\_reg\_request\_header = new pre\_reg\_request\_header;

$dbh\_pre\_reg\_request\_header->execute\_query("SELECT COUNT(pre\_reg\_request\_header\_id) AS req\_id FROM pre\_reg\_request\_header");

$result = $dbh\_pre\_reg\_request\_header->result;

$row = $result->fetch\_assoc();

extract($row);

if($row == '')

{

$counter = 1;

}

else

{

$counter = ++$req\_id;

}

$req\_tag = 'RN';

$req\_count = str\_pad($counter, '3','0', STR\_PAD\_LEFT);

$code = "{$req\_tag}{$req\_count}";

if($submit\_true)

{

$param\_header = array(

'course\_id' => $student->get\_course\_id\_by\_username($\_SESSION['user']),

'department\_id' => $student->get\_department\_id\_by\_username($\_SESSION['user']),

'student\_id'=>$student->get\_student\_id\_by\_username($\_SESSION['user']),

'date\_applied' => date('Y-m-d h:i:s'),

'registration\_number' => $code

);

$header -> add($param\_header);

$pre\_reg\_request\_header\_id = $header->auto\_id;

for($b = 0; $b<$\_POST['counter']; ++$b)

{

$param\_request = array(

'pre\_reg\_request\_header\_id'=> $pre\_reg\_request\_header\_id,

'subject\_id'=>$arr\_results[$b]['subject\_id'],

'section\_id'=>$arr\_results[$b]['section\_id'],

'schedule\_id'=>$arr\_results[$b]['schedule\_id'],

);

$request->add($param\_request);

}

// debug($param\_header);

log\_action('Pressed submit button');

redirect('add\_subject.php?notice=1');

}

else

{

$html->display\_error("Please select a Subject followed by a Section.");

}

}

// debug($arr\_results);

$html->draw\_header();

$html->draw\_container\_div\_start();

$html->draw\_fieldset\_header('Add/Remove Subject');

$html->draw\_fieldset\_body\_start();

// debug($counter);

echo '<input type="hidden" name="counter" value="'.count($arr\_results).'">';

// debug($\_POST);

$html->draw\_select\_field\_from\_query($query\_code,$list\_value\_code,$list\_items\_code,'Subject Code','code\_subject',FALSE,TRUE,'','');

$html->draw\_select\_field\_from\_query($query\_section,$list\_value\_section,$list\_items\_section,'Section','list\_section',FALSE,TRUE,'','');

$html->draw\_button('special', 'btn\_add', 'btn\_add', 'Add', FALSE, "", '');

$html->draw\_button('special', 'btn\_remove\_all', 'btn\_remove\_all', 'Remove All', FALSE, "", '');

// $html->draw\_text\_field('Change # of Subject', 'change\_num',FALSE, 'text', TRUE,'size ="3"');

// $html->draw\_button('Go', 'btn\_submit', 'btn\_submit', 'Go', FALSE, "", '');

echo'<center><table class = "listView" border = "1" width = 1300 >';

echo'<tr class = "listRowHead">';

echo'<center><td>Subject Code </td></center>';

echo'<td>Subject Name </td>';

echo'<td>Section</td>';

echo'<td>Schedule</td>';

echo'<td>Action</td>';

echo'</tr>';

for($i = 0; $i < count($arr\_results); ++$i)

{

if(isset($arr\_results[$i]))

{

extract($arr\_results[$i]);

}

$table\_row = ($i % 2 == 0)? "<tr class='listRowEven'>": "<tr class='listRowOdd'>" ;

$schedule\_from = date('h:i A', strtotime($schedule\_from));

$schedule\_to = date('h:i A', strtotime($schedule\_to));

$schedule = $schedule\_day . ' ' . $schedule\_from . ' - ' . $schedule\_to;

echo $table\_row;

echo'<td>' . $subject\_code . '</td>';

echo'<td>' . $subject\_name . '</td>';

echo'<td>' . $section\_name . '</td>';

echo'<td>' . $schedule . '</td>';

echo'<td><input type="hidden" name="id['.$i.']" value="'.$i.'">';

$html->draw\_button("special",'btn\_remove','btn\_remove','Remove',FALSE,'');

echo '</td></tr>';

}

// debug($arr\_results);

echo '</table>';

$html->draw\_fieldset\_body\_end();

$html->draw\_fieldset\_footer\_start();

$html->draw\_button('submit');

$html->draw\_fieldset\_footer\_end();

?>

1. **request\_approver.php**

<?php

require 'path.php';

init\_cobalt('Request approver');

require'subclasses/pre\_reg\_request\_header\_html.php';

$html = new pre\_reg\_request\_header\_html;

$html->draw\_header('');

$html->draw\_container\_div\_start();

$html->draw\_fieldset\_header('Request Approval');

$html->draw\_fieldset\_body\_start();

$query = "SELECT \*

FROM pre\_reg\_request\_details

LEFT JOIN subject

ON pre\_reg\_request\_details.subject\_id = subject.subject\_id

LEFT JOIN section

ON pre\_reg\_request\_details.section\_id = section.section\_id

LEFT JOIN schedule

ON pre\_reg\_request\_details.schedule\_id = schedule.schedule\_id

LEFT JOIN pre\_reg\_request\_header

LEFT JOIN student

ON pre\_reg\_request\_header.student\_id = student.student\_id

ON pre\_reg\_request\_details.pre\_reg\_request\_header\_id = pre\_reg\_request\_header.pre\_reg\_request\_header\_id

WHERE pre\_reg\_request\_header.is\_approved = ''";

require\_once 'subclasses/pre\_reg\_request\_header.php ';

$dbh\_request = new pre\_reg\_request\_header;

$dbh\_request->execute\_query($query);

$result = $dbh\_request->result;

$arr\_app\_form = array();

while($row = $result->fetch\_assoc())

{

$arr\_app\_form[] = $row;

}

echo'<table>';

echo'<center><table class = "listView" border = "1" width = 1300>';

echo'<tr class = "listRowHead">';

echo '<td> Student Number </td>';

echo '<td> Student Name </td>';

echo '<td> Subject </td>';

echo '<td> Section </td>';

echo '<td> Schedule </td>';

// echo '<td> </td>';

echo'</tr>';

for($x = 0; $x<count($arr\_app\_form); ++$x)

{

extract($arr\_app\_form[$x]);

$table\_row = ($x % 2 == 0)? "<tr class='listRowEven'>": "<tr class='listRowOdd'>" ;

echo $table\_row;

$schedule = $schedule\_day . ' ' . $schedule\_from . ' - ' . $schedule\_to;

$name = $last\_name . ', ' . $first\_name . ' ' . $middle\_name;

echo"<td>

<a href = 'request\_details.php?req\_num=".$registration\_number."'> "

. $registration\_number .

"</a>

</td>";

echo '<td>' . $name . '</td>';

echo '<td>' . $subject\_name . '</td>';

echo '<td>' . $section\_name . '</td>';

echo '<td>' . $schedule . '</td>';

// echo '<td>' . $ . '</td>';

}

?>

1. **request\_details.php**

<?php

require 'path.php';

init\_cobalt('Request details');

require'subclasses/pre\_reg\_request\_header.php';

require'subclasses/pre\_reg\_request\_header\_html.php';

init\_var($\_POST['btn\_cancel']);

init\_var($\_POST['btn\_submit']);

init\_var($\_POST['status\_approval']);

init\_var($\_GET['req\_num']);

$pre\_reg\_request = cobalt\_load\_class('pre\_reg\_request\_header');

$approver = $\_SESSION['last\_name'] . ', ' . $\_SESSION['first\_name'];

if($\_POST['btn\_cancel'])

{

log\_action('Pressed cancelled button');

redirect('level\_one\_approval.php');

}

if(isset($\_GET["req\_num"]))

{

$req\_num = $\_GET["req\_num"];

if($\_POST['btn\_submit'])

{

/\*

|-----------------------------------------------------------------------

| Edit any "Add" behavior functionalities in this block.

|-----------------------------------------------------------------------

\*/

$param\_req = array(

'approved\_by'=>$\_POST['approver'],

'remarks'=>$\_POST['comments'],

'is\_approved'=>$\_POST['status\_approval'],

'pre\_reg\_request\_header\_id'=>$\_POST['req\_id'][0]

);

$pre\_reg\_request->edit\_req($param\_req);

log\_action('Pressed submit button');

redirect('request\_approver.php');

// redirect('request\_approver.php?pre\_reg\_request\_header\_id=' . $\_POST['pre\_reg\_request\_header\_id'][0]);

/\*

|-----------------------------------------------------------------------

| NOTHING FOLLOWS

|-----------------------------------------------------------------------

\*/

}

$html = new pre\_reg\_request\_header\_html;

$html->draw\_header('');

$html->draw\_container\_div\_start();

$html->draw\_fieldset\_header('Request Details');

$html->draw\_fieldset\_body\_start();

$query = "SELECT \*

FROM pre\_reg\_request\_details

LEFT JOIN subject

ON pre\_reg\_request\_details.subject\_id = subject.subject\_id

LEFT JOIN section

ON pre\_reg\_request\_details.section\_id = section.section\_id

LEFT JOIN schedule

ON pre\_reg\_request\_details.schedule\_id = schedule.schedule\_id

LEFT JOIN pre\_reg\_request\_header

LEFT JOIN student

ON pre\_reg\_request\_header.student\_id = student.student\_id

ON pre\_reg\_request\_details.pre\_reg\_request\_header\_id = pre\_reg\_request\_header.pre\_reg\_request\_header\_id

WHERE pre\_reg\_request\_header.is\_approved = ''

AND pre\_reg\_request\_header.registration\_number = '".$req\_num."'";

require\_once 'subclasses/pre\_reg\_request\_header.php ';

$dbh\_request = new pre\_reg\_request\_header;

$dbh\_request->execute\_query($query);

$result = $dbh\_request->result;

$arr\_app\_form = array();

while($row = $result->fetch\_assoc())

{

$arr\_app\_form[] = $row;

}

echo'<table>';

echo'<center><table class = "listView" border = "1" width = 1300>';

echo'<tr class = "listRowHead">';

echo '<td> Student Number </td>';

echo '<td> Student Name </td>';

echo '<td> Subject </td>';

echo '<td> Section </td>';

echo '<td> Schedule </td>';

// echo '<td> </td>';

echo'</tr>';

for($x = 0; $x<count($arr\_app\_form); ++$x)

{

extract($arr\_app\_form[$x]);

$table\_row = ($x % 2 == 0)? "<tr class='listRowEven'>": "<tr class='listRowOdd'>" ;

echo $table\_row;

echo '<input type = "text" name = "req\_id['.$x.']" value = "'.$pre\_reg\_request\_header\_id.'" hidden >';

$schedule = $schedule\_day . ' ' . $schedule\_from . ' - ' . $schedule\_to;

$name = $last\_name . ', ' . $first\_name . ' ' . $middle\_name;

echo"<td>

$registration\_number

</td>";

echo '<td>' . $name . '</td>';

echo '<td>' . $subject\_name . '</td>';

echo '<td>' . $section\_name . '</td>';

echo '<td>' . $schedule . '</td>';

// echo '<td>' . $ . '</td>';

}

echo '</table> <br>';

$approved\_status = array('per\_line' =>FALSE,

'items' =>array('Yes', 'No'),

'values'=>array('Yes', 'No'));

$html->draw\_text\_field('Approve By', 'approver', FALSE, 'text', TRUE, 'readonly');

echo'<br>';

$html->draw\_radio\_buttons($approved\_status, 'Approval Status', 'status\_approval', TRUE,'');

echo'<br>';

$html->draw\_text\_field('Remarks', 'comments', FALSE, 'textarea', TRUE, '');

}

$html->draw\_fieldset\_body\_end();

$html->draw\_fieldset\_footer\_start();

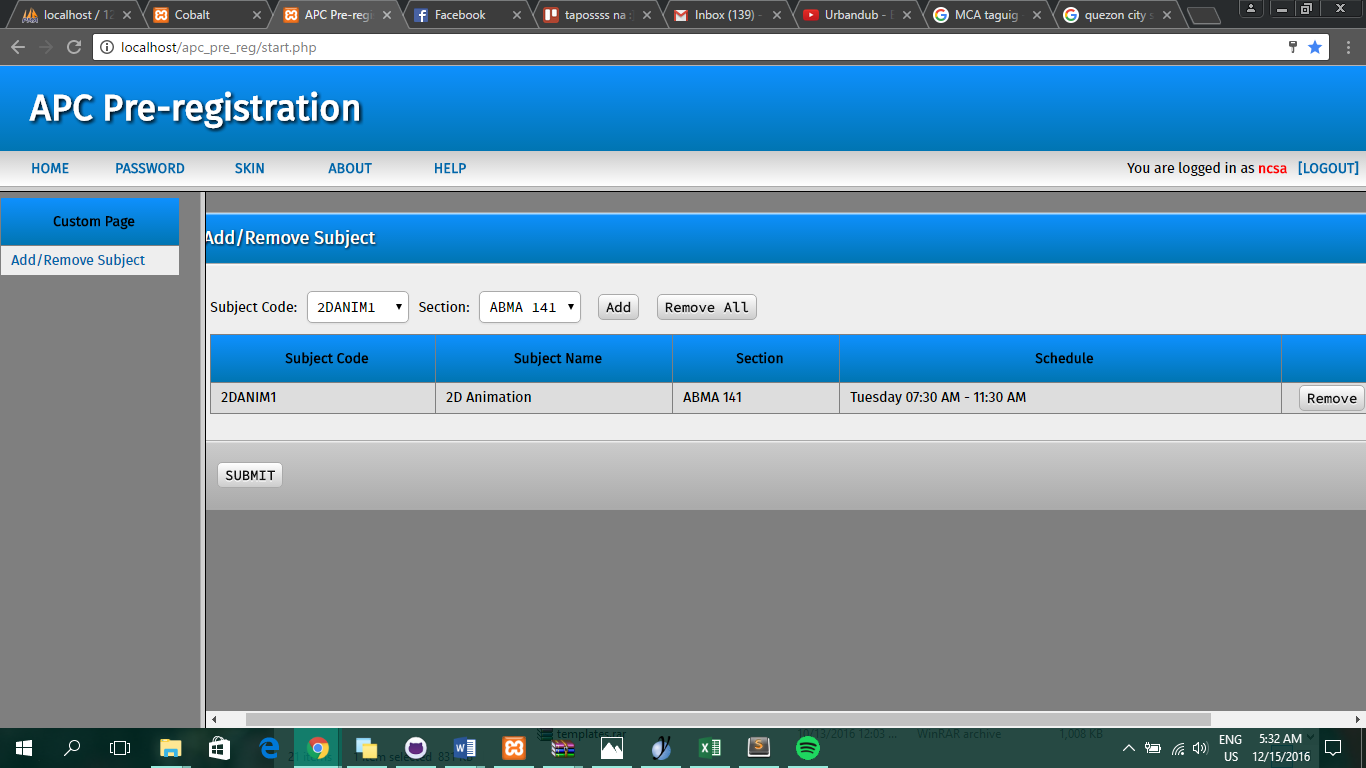
$html->draw\_submit\_cancel();

$html->draw\_fieldset\_footer\_end();

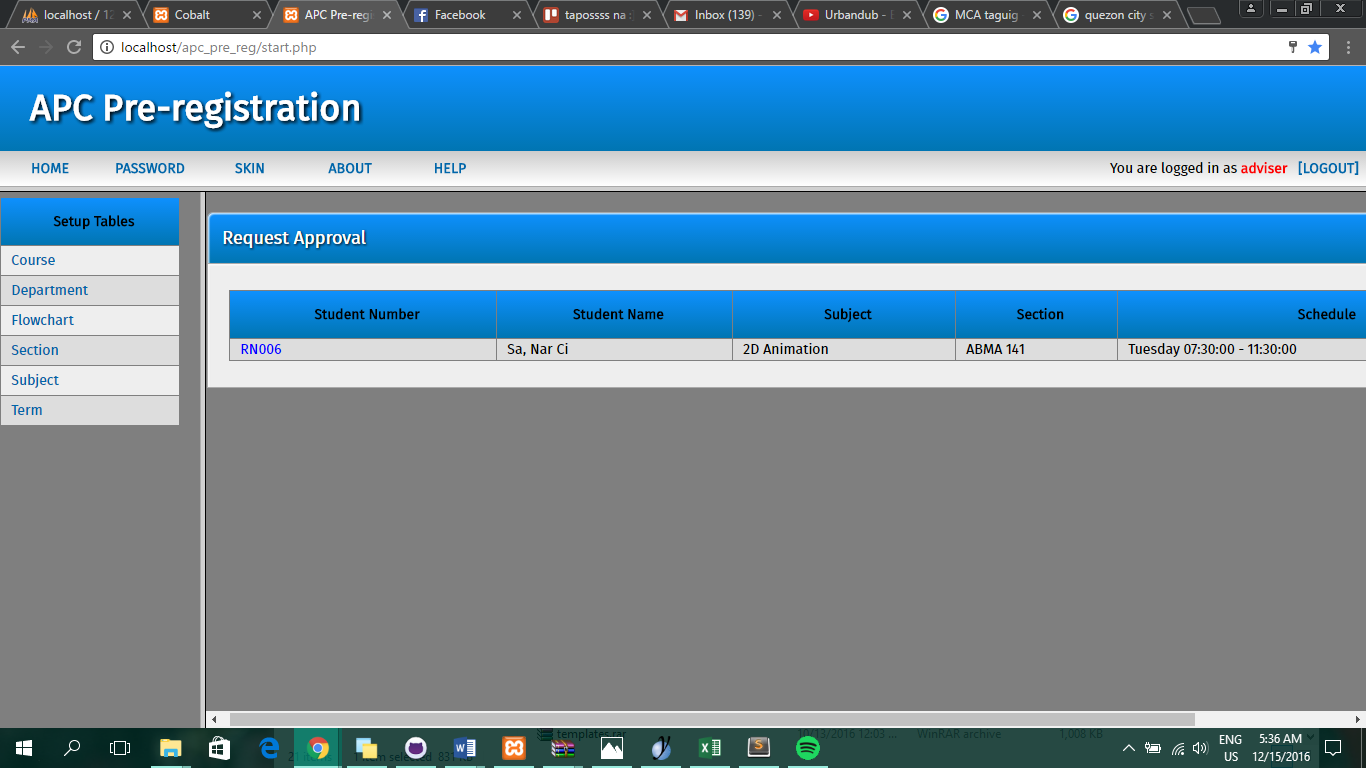
?>

## Evaluation Tool or Test Document

Student Side



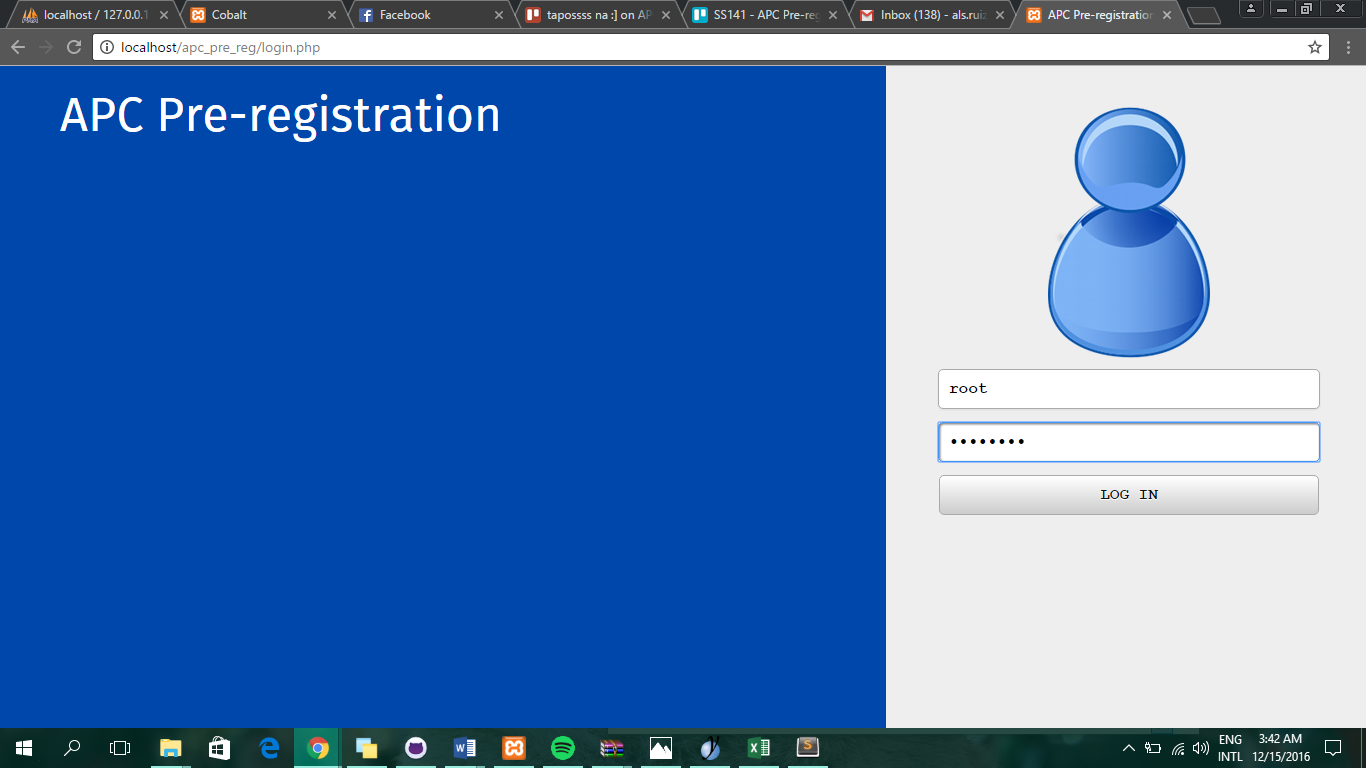
Adviser/Approver Side



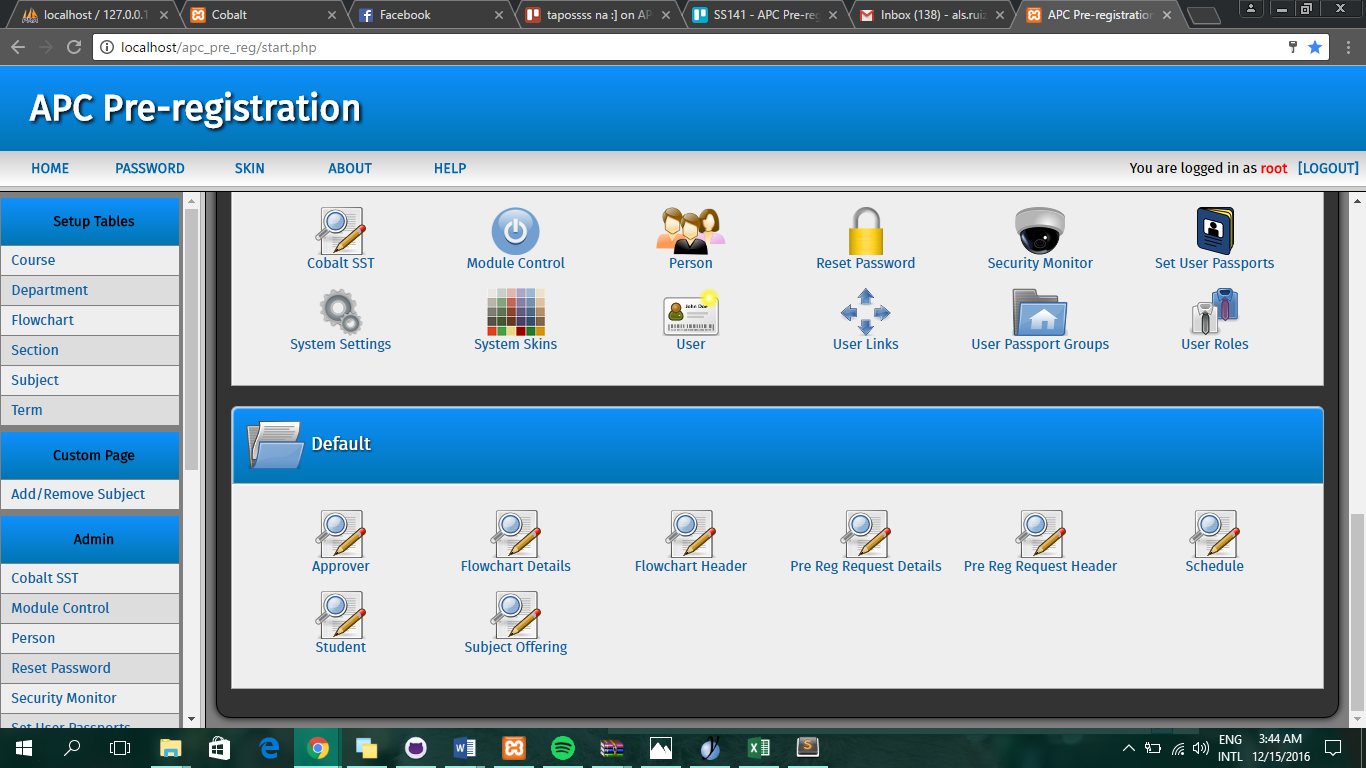
## Sample Input / Output Reports

**Administrator/Root**

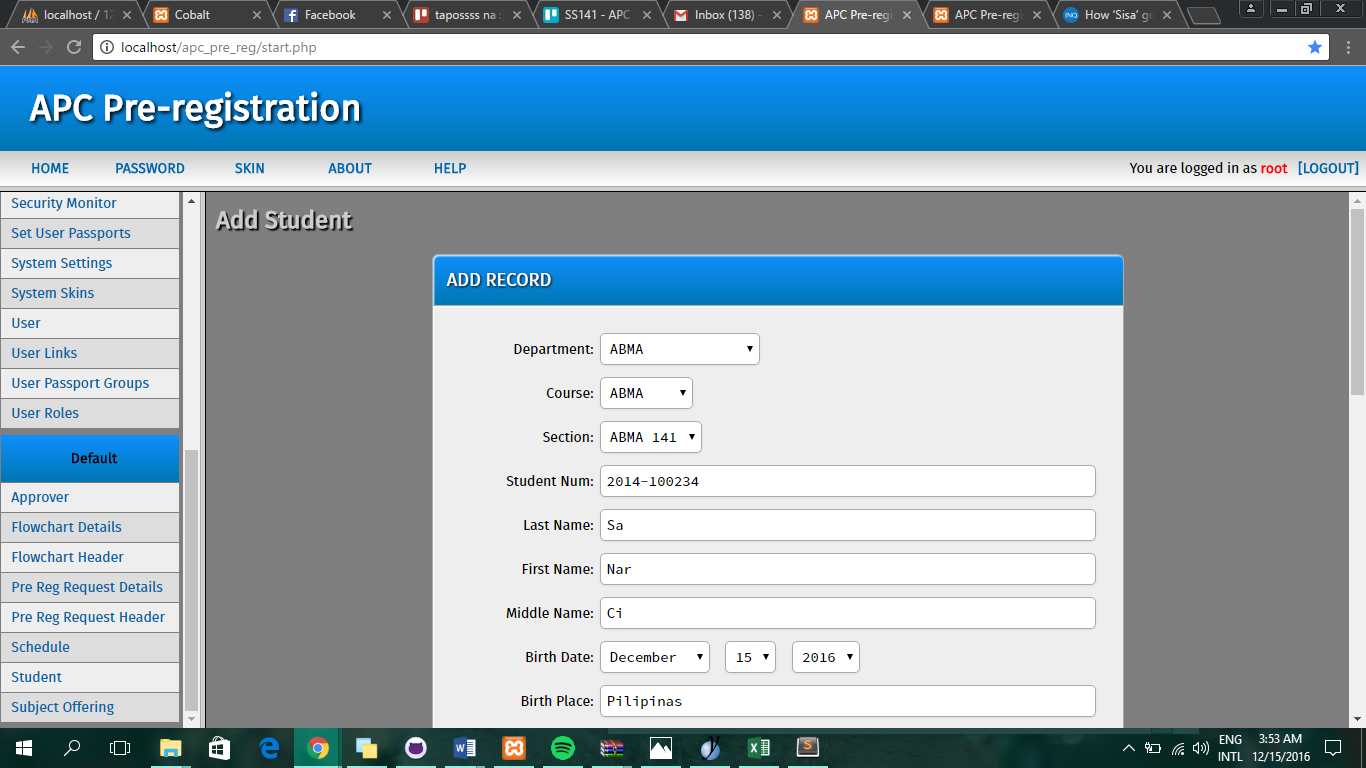
Login

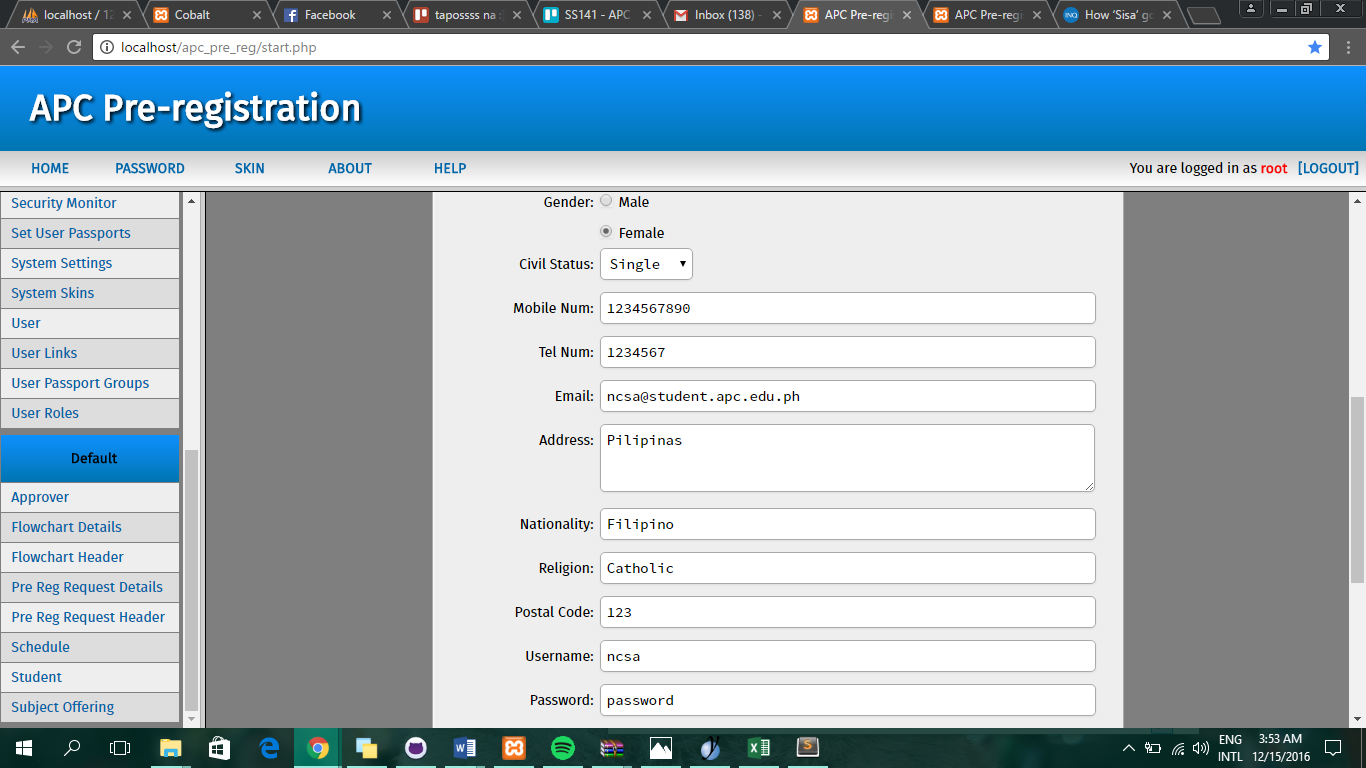


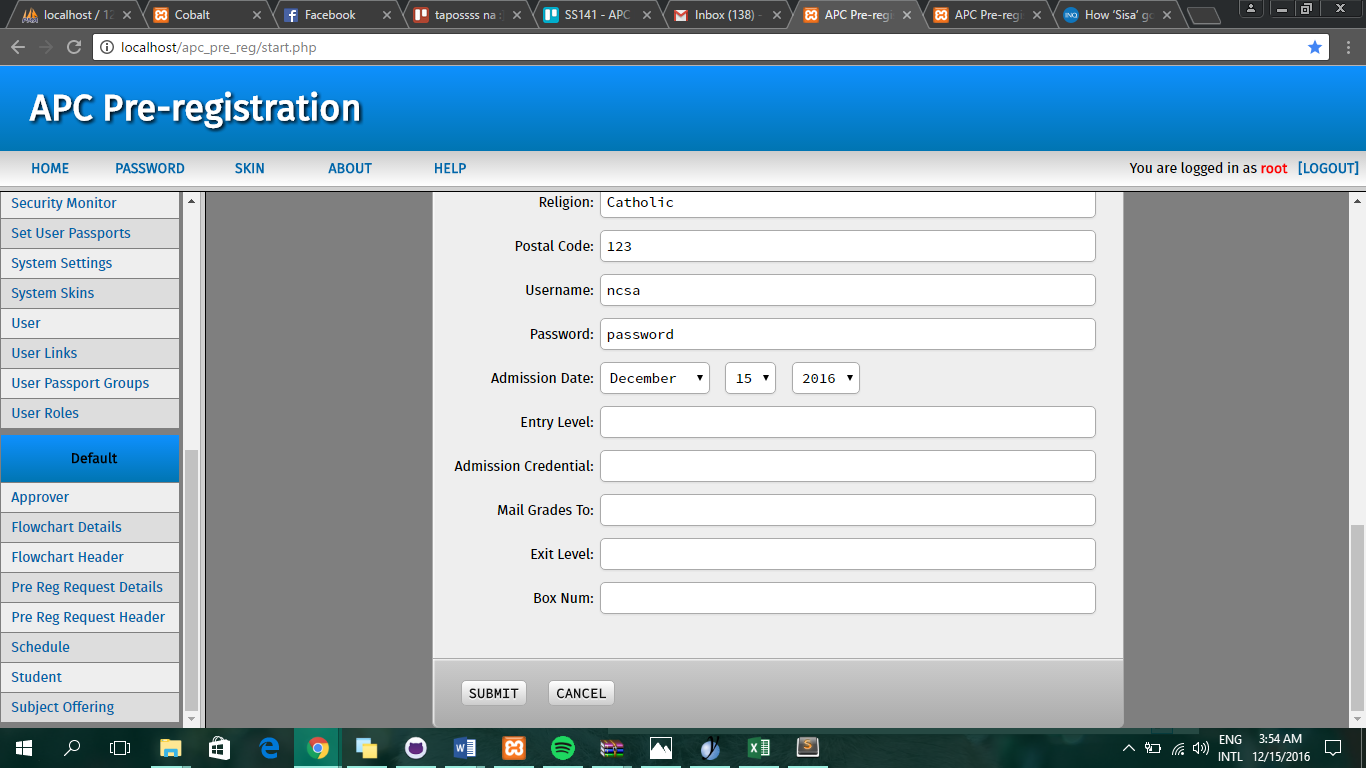
## Home



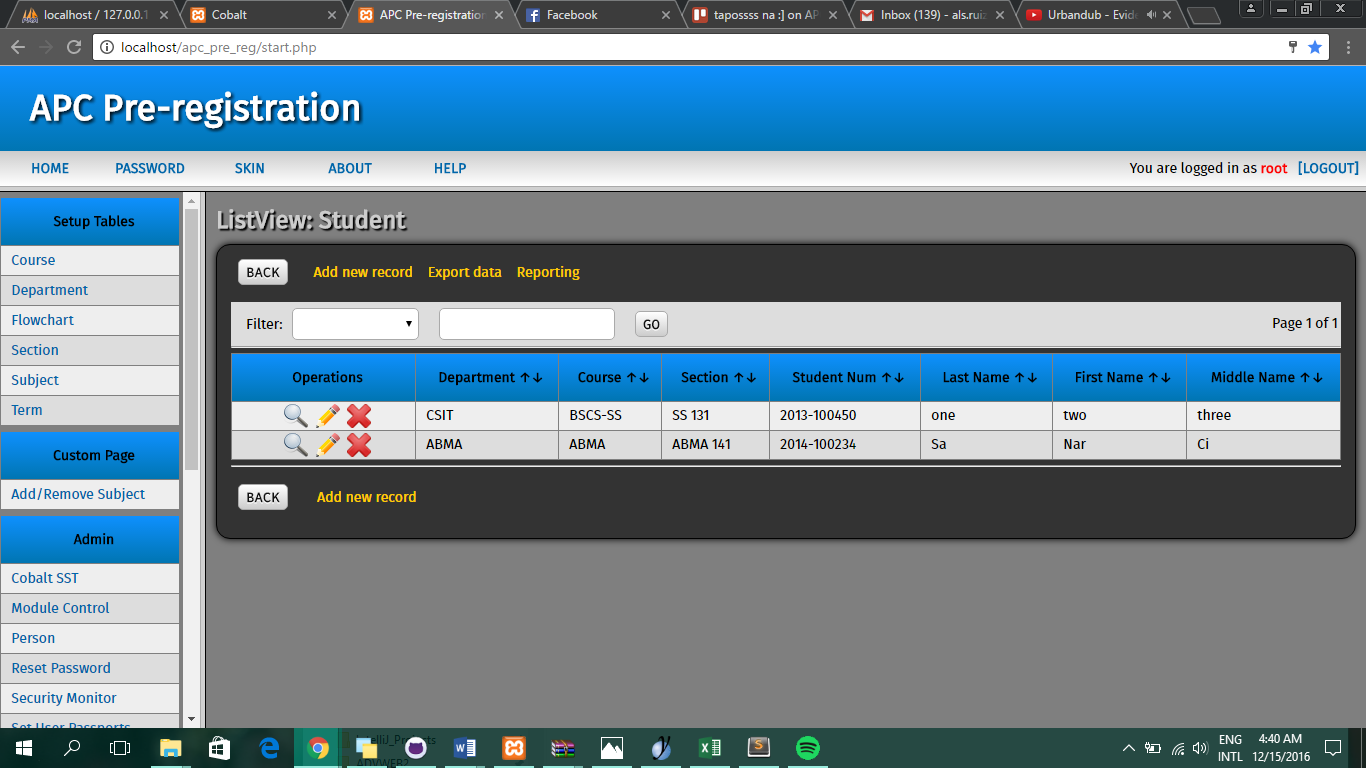
Input new record in Student table



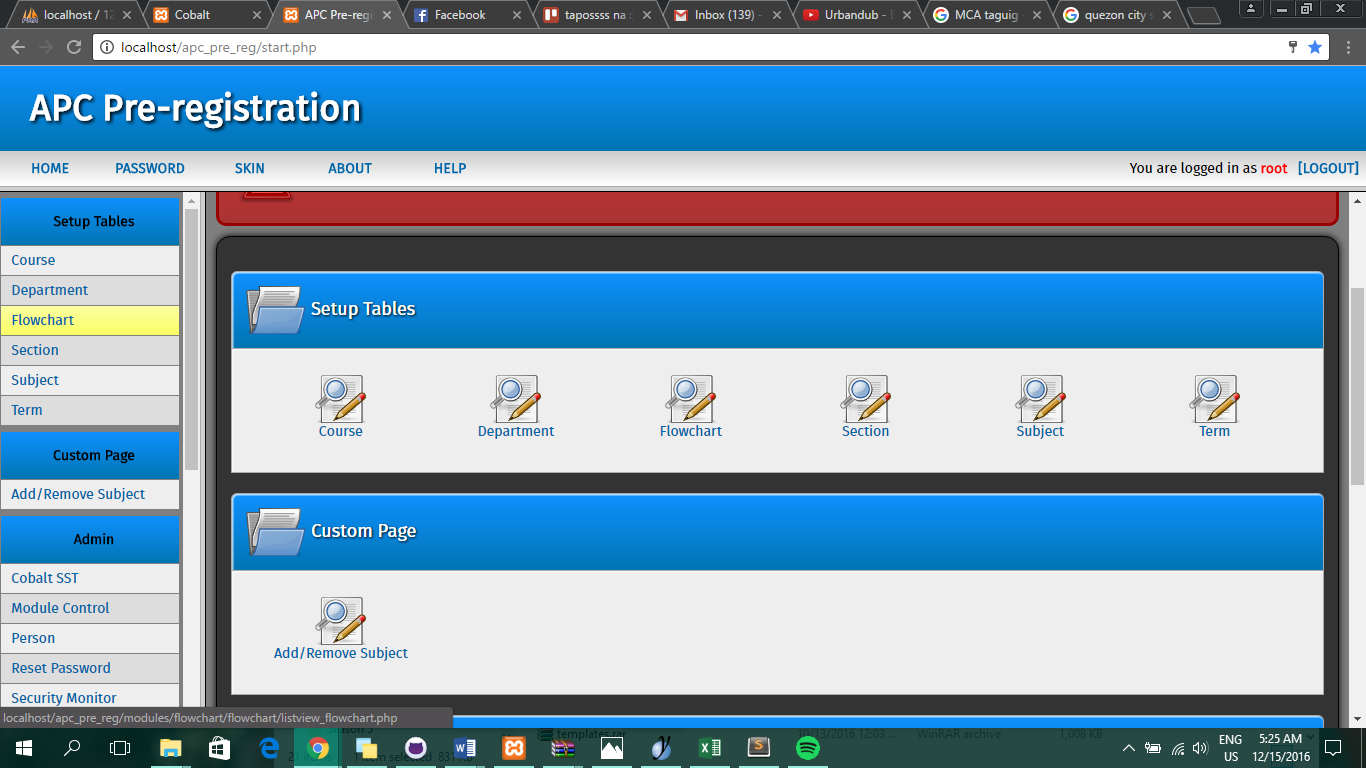


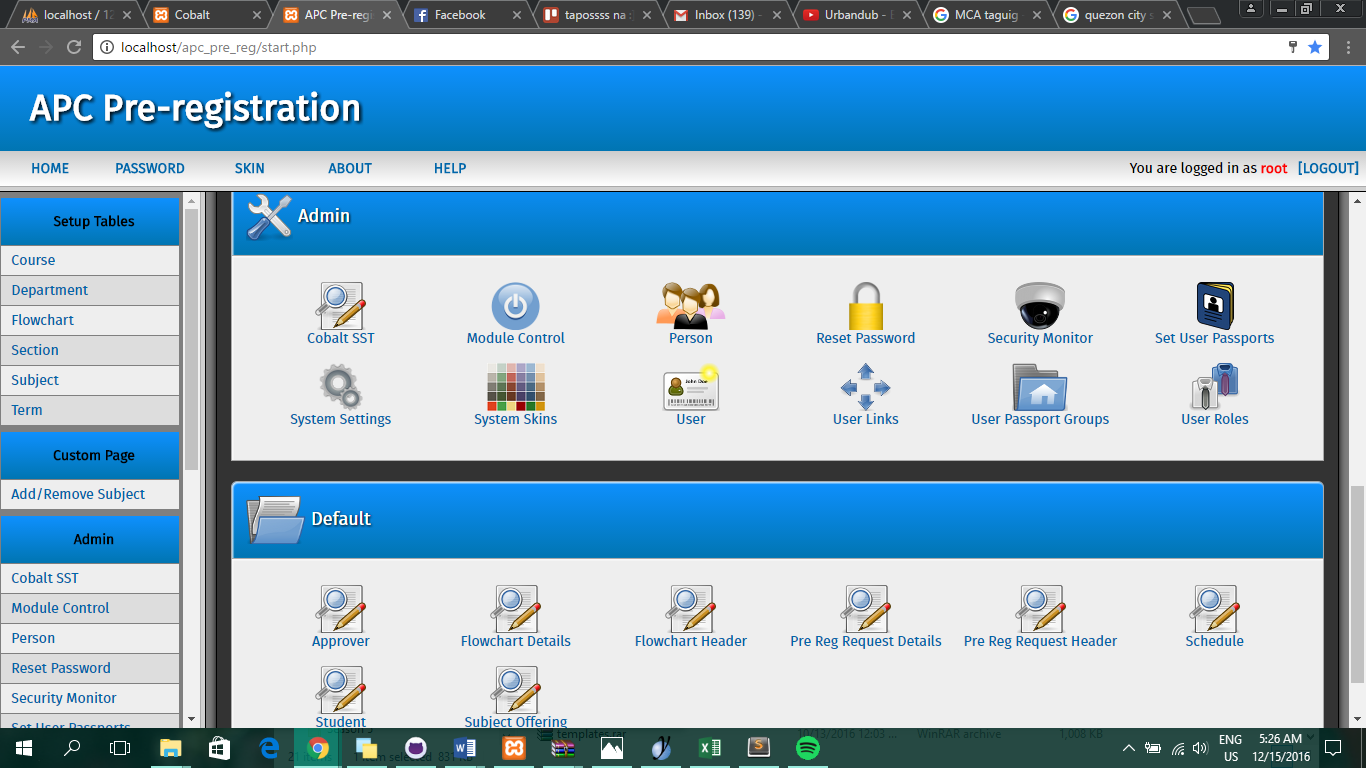


Output when records in Student table are submitted



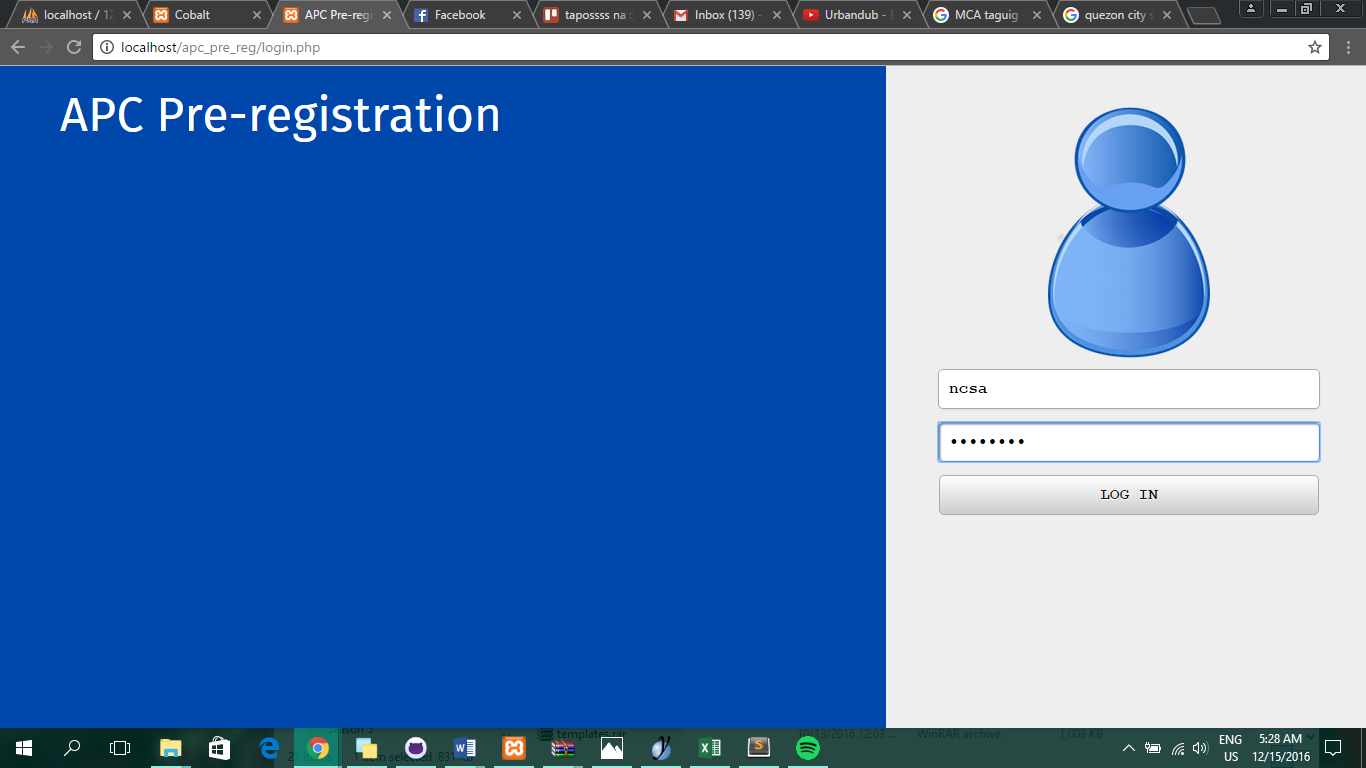
All tables





**Student**

Login



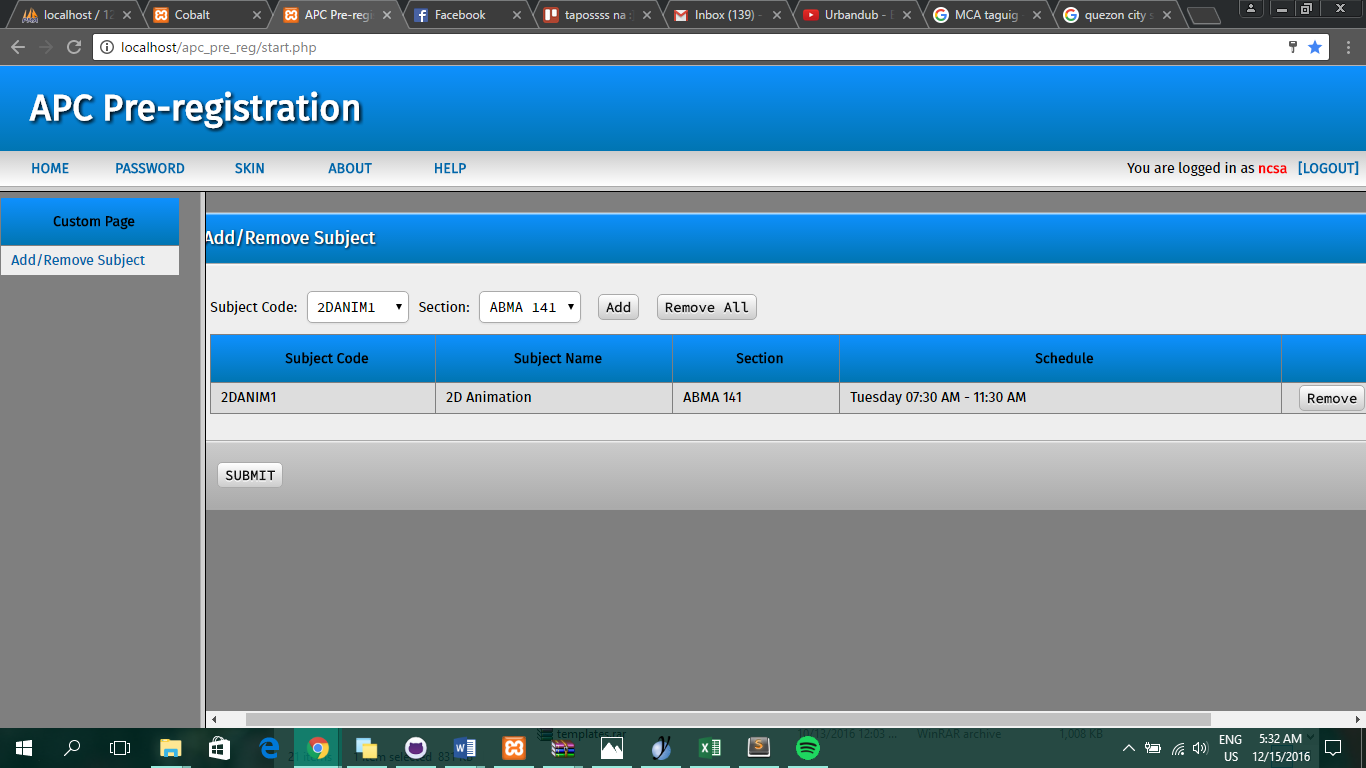
Home



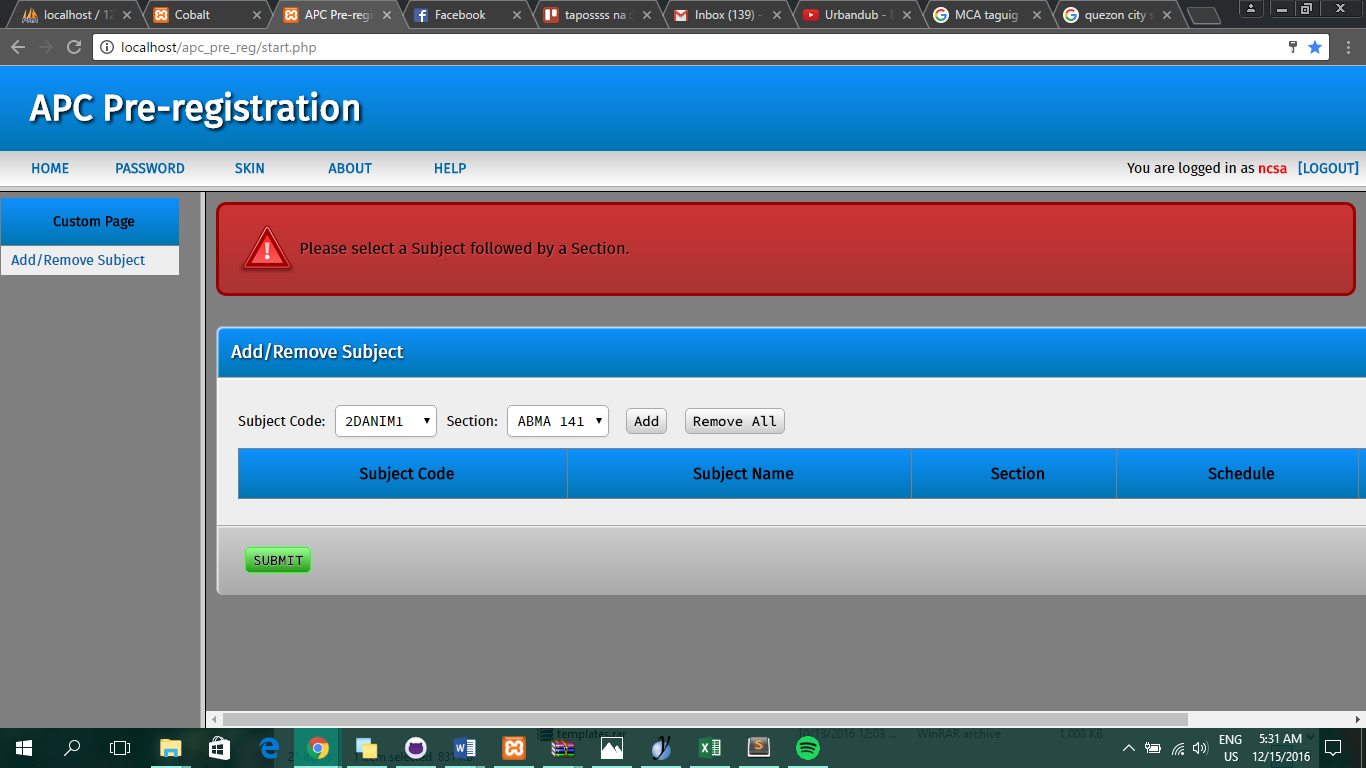
Add/Drop



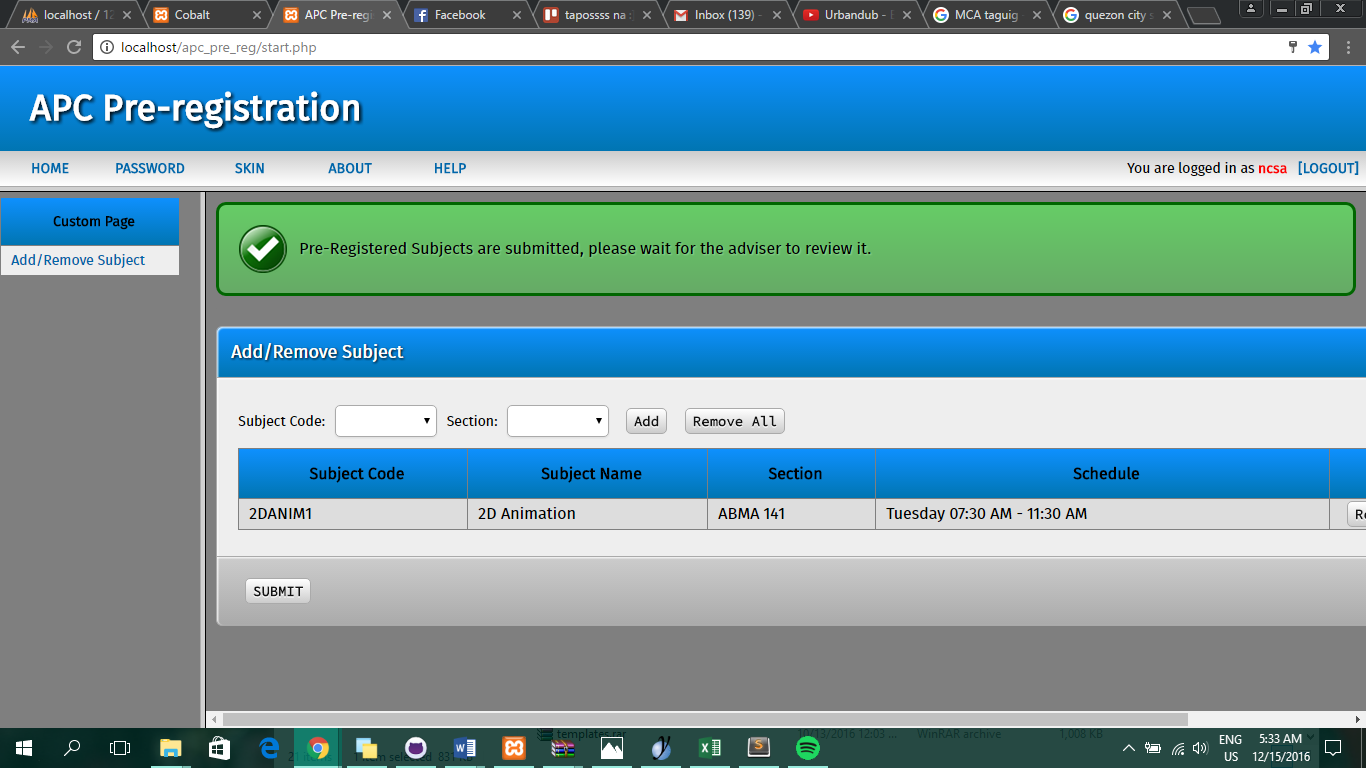
When Add button is clicked



When Add button is not clicked, only Submit

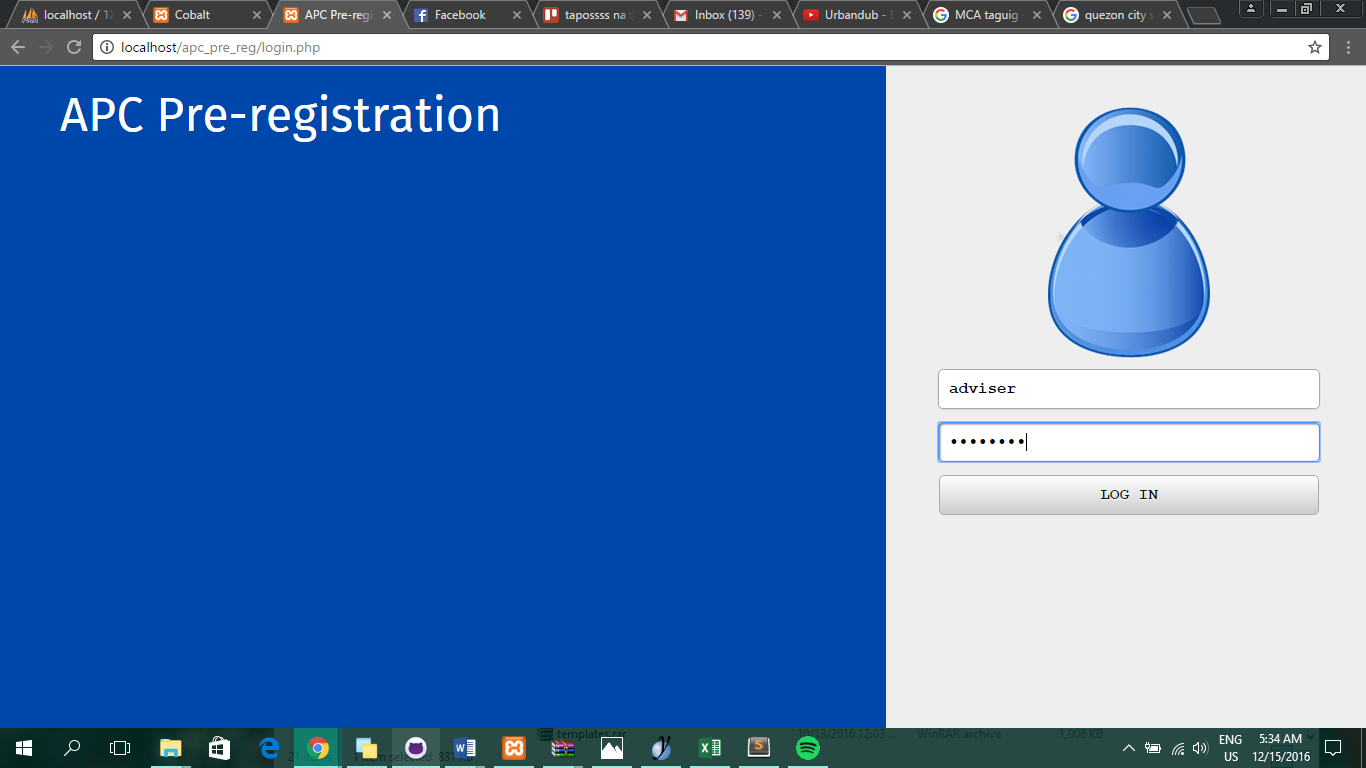


After adding a subject, Submit button is clicked

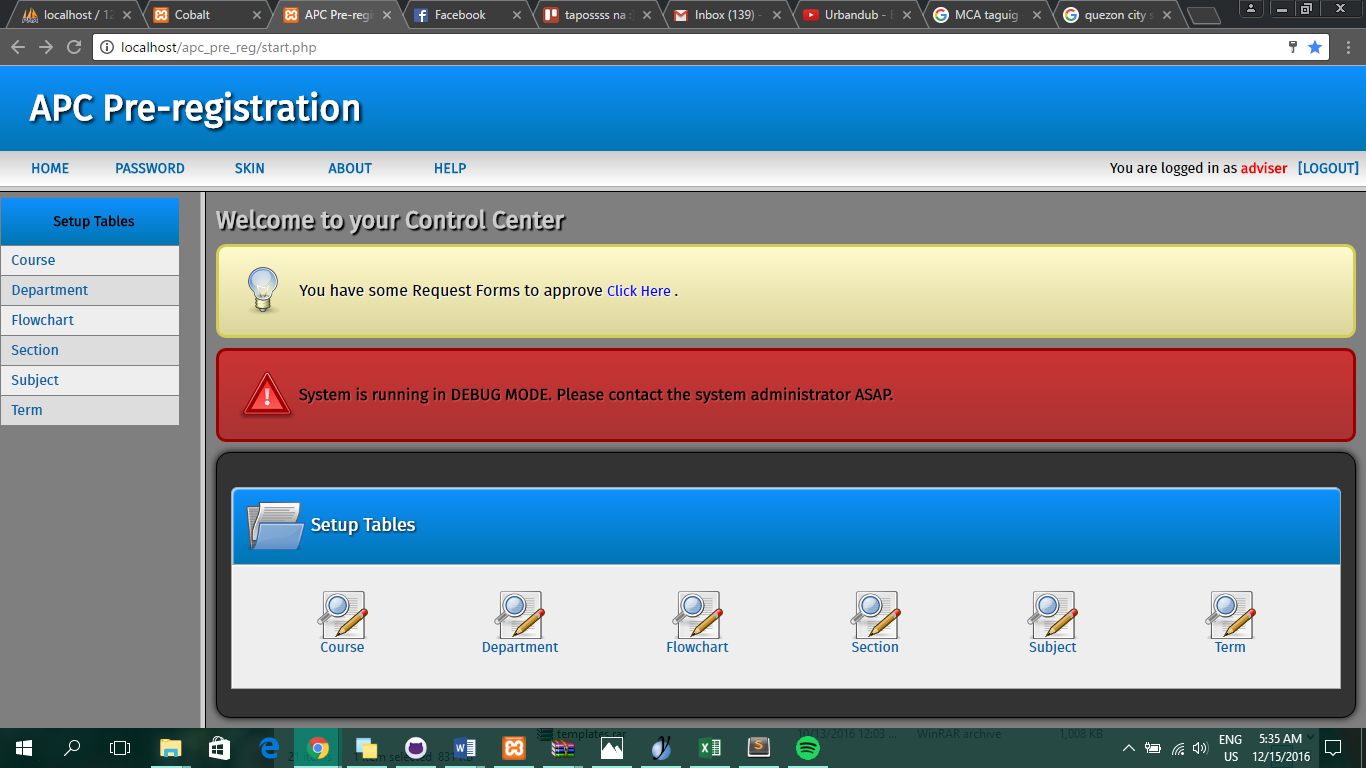


**Adviser**

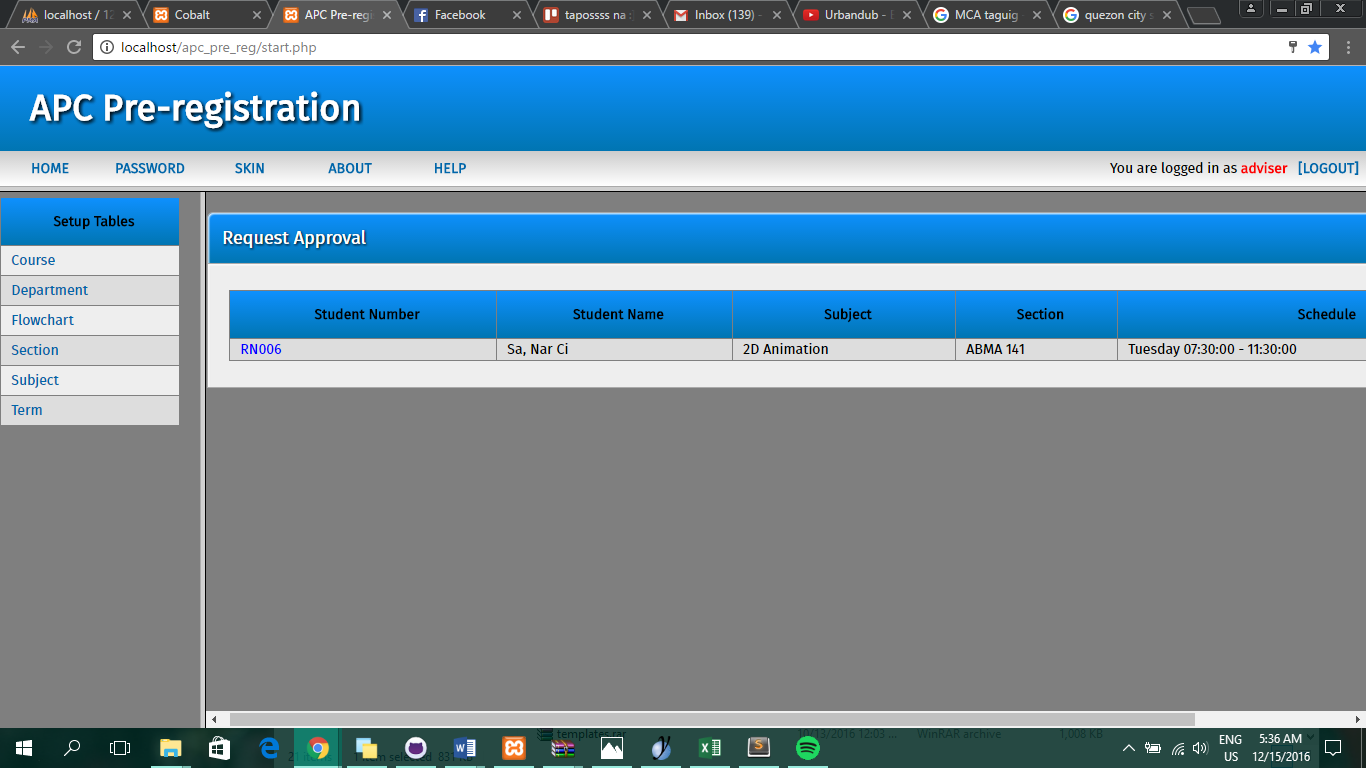
Login



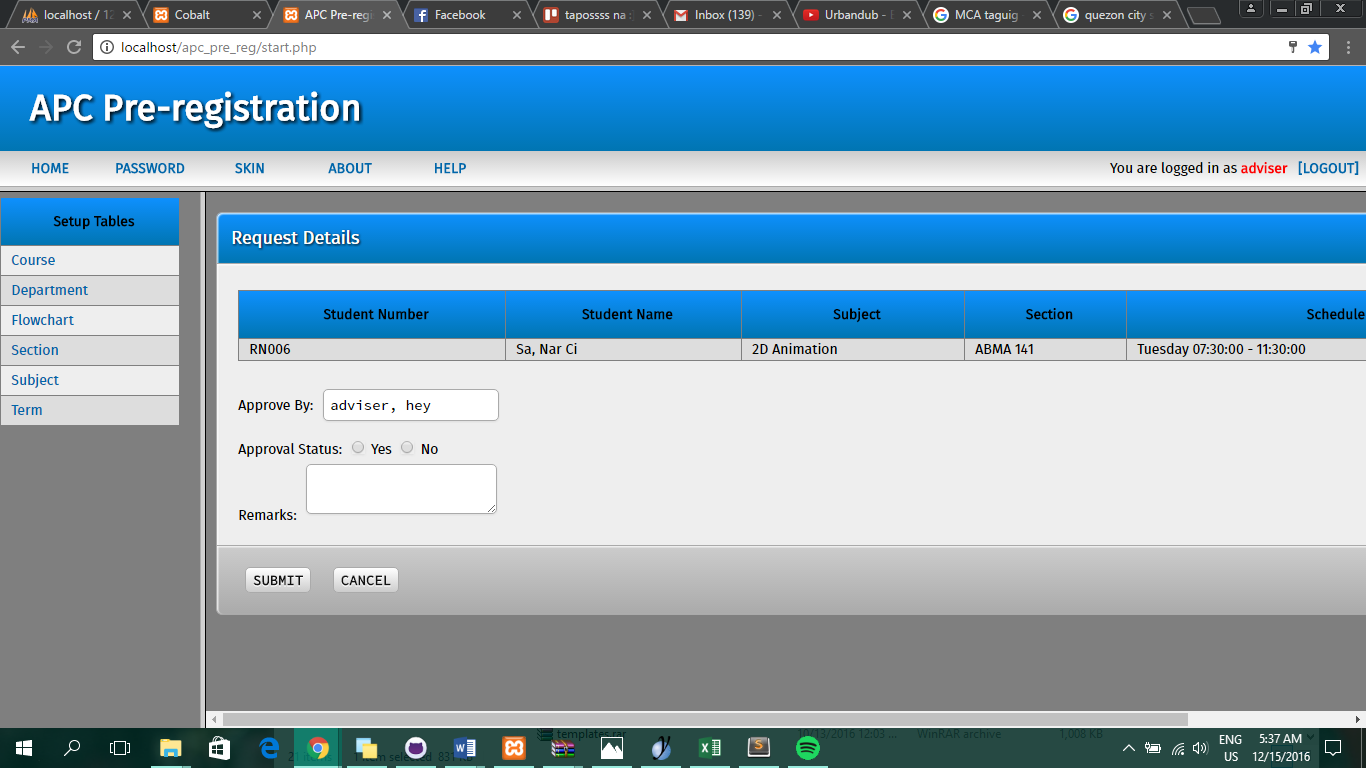
Home



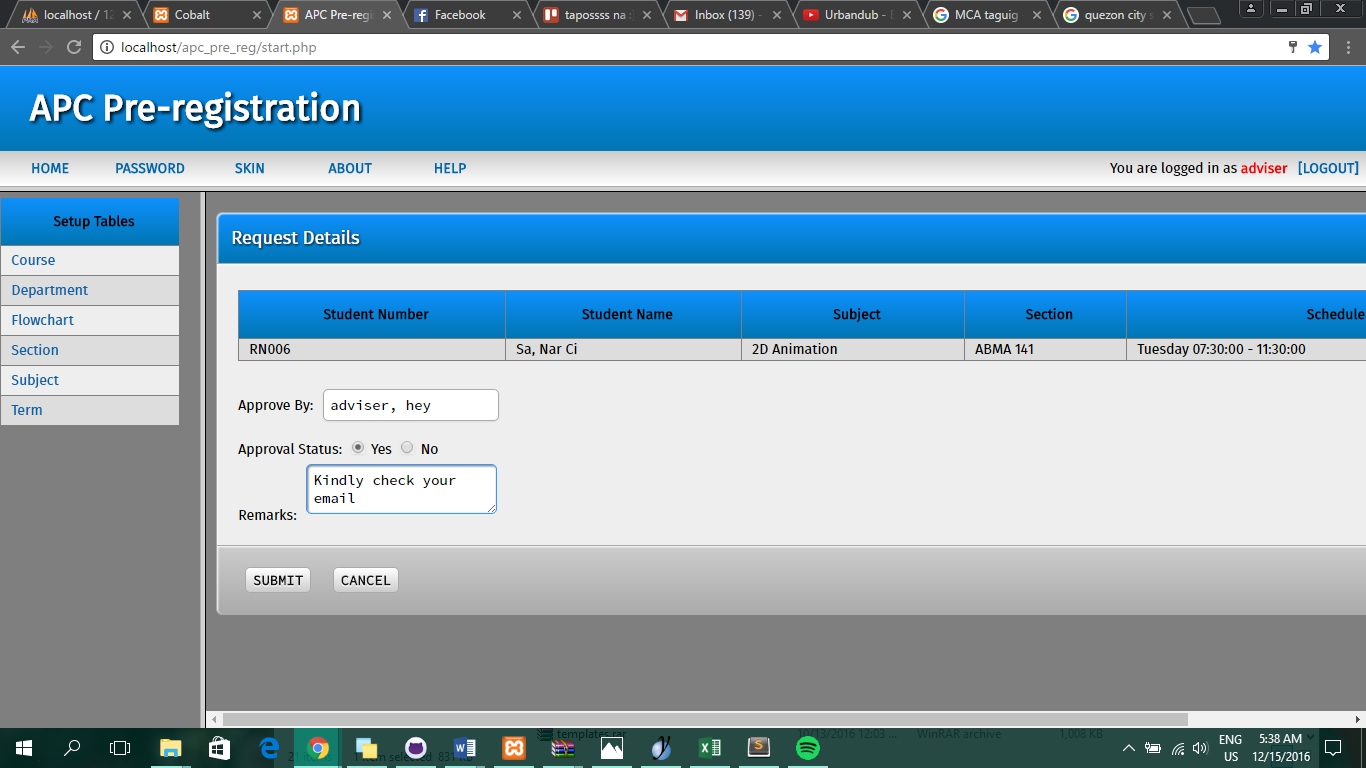
“Click here” Link is clicked



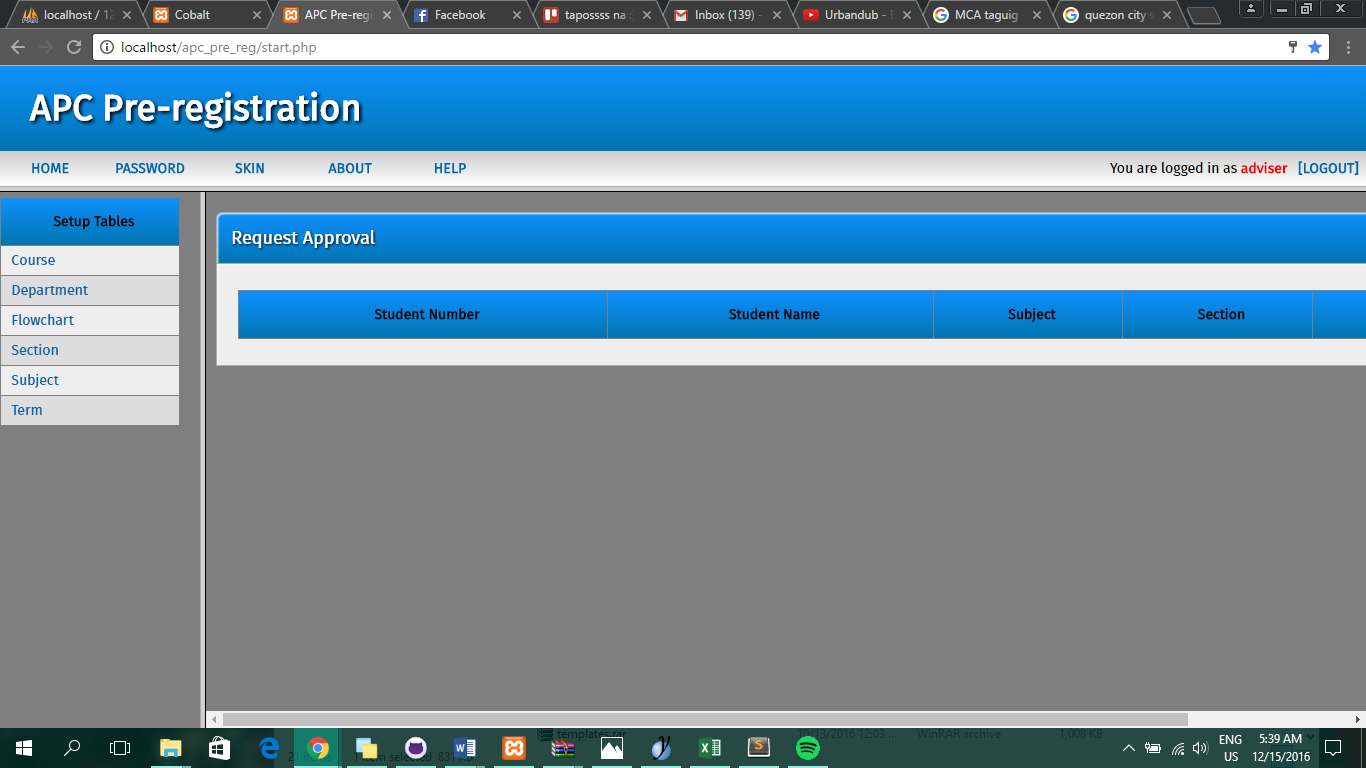
Registration Number of a request is clicked (Ex. RN006)



All fields are populated



Submit button is clicked



## User Guides

Upon the first enrollment of a student, the ITRO generates a FLAVIO account used for accessing both the main Flavio system and the Online Inquiry. Professors also have existing Flavio accounts. Therefore, there is no registration feature for the online pre-registration system.

* Allowed Users

APC students and professors

* Where to go

Users must access <http://apc.edu.ph/flavio>.

* Logging In

Users must input their user name and password in the log in page and click "Log In"

* Change Skins

Cobalt has various skins that can be used by a user

* Administrator

Super Admin has 100% control over the system while System Admin only has control in the built-in tables of Cobalt

* Adding and Removing a Subject (for student only)

Once logged in, the students can have the option to add or remove a subject to their next term schedule. To do this, the users must click on the "Add/Remove Subject" button.

* Adding a Subject (for student only)

In the Add Subject section, students can search for their desired subject and section and click "Add Subject" to add the subject to their next term schedule.

* Removing a Subject (for student only)

In the Remove Subject section, students can search for their desired subject and section and click "Remove" to remove the subject to their next term schedule.

* Approving/Rejecting Requests (for adviser/Approver only)

Adivsers/Approvers, have the option to approve or reject the requests of students. Users must click on the "Requests" button. A list of student's registration requests would be displayed. The user can approve or reject a request. To approve a request, click "Approve" and to reject it click "Reject". If the user rejects a request, the system will require the user to give feedback or reasons for the rejection.

* Logging Out

In the upper right corner, click the "LOGOUT" button to log out of the system.

## Curriculum Vitae of team members

ROMULUS DIEGO PASCUAL GLORIA

61A Lot 14 Fort Bonifacio, AFPOVAI Phase 4,

Bonifacio Ave., Taguig City

09959682394

rpgloria@student.pac.edu.ph

**EDUCATION**

**Tertiary**

Asia Pacific College

Magallanes, Makati City

Bachelor of Science in Computer Science

(undergraduate)

June, 2014-Present

**Secondary**

MCA Montessori School

Taguig City, Metro Manila

2010-2014

**Elementary**

MCA Montessori School

Taguig City, Metro Manila

2004-2009

**RELATED SKILLS**

* Research Writing Skills
* Technical Skills (MS Word, Excel, PowerPoint, Use of Internet search engines, Email)
* Programming Skills (HTML, CSS, Bootstrap, PHP)

**REFERENCES**

* Mr. Manuel Sanchez

College Professor, Project-Based Learning Professor, and Adviser

Asia Pacific College

JAIRUS ADRIAN GOMEZ ROGUEL

124 Kalantiaw Street, Project 4, Quezon City

09988662455

jgroguel@student.apc.edu.ph

**EDUCATION**

**Tertiary**

Asia Pacific College

Magallanes, Makati City

Bachelor of Science in Computer Science

(undergraduate)

June, 2014-Present

**Secondary**

Quezon City Science High School

Quezon City, Metro Manila

2010-2014

**Elementary**

Saint Anthony Academy

Quezon City, Metro Manila

2004-2009

**RELATED SKILLS**

* Research Writing Skills
* Technical Skills (MS Word, Excel, PowerPoint, Use of Internet search engines, Email)
* Programming Skills (HTML, CSS, Bootstrap, PHP, MySQL, Java)

**REFERENCES**

* Mr. Manuel Sanchez

College Professor, Project-Based Learning Professor, and Adviser

Asia Pacific College

****ANGELICA LAURENE SERRANO RUIZ

89 D Mariquita Street, Pasay City Barangay 82

09065913066

asruiz@student.apc.edu.ph

**EDUCATION**

**Tertiary**

Asia Pacific College

Magallanes, Makati City

Bachelor of Science in Computer Science

(undergraduate)

June, 2014-Present

**Secondary**

St. Scholastica’s College, Manila

Malate, Metro Manila

2010-2014

**Elementary**

St. Scholastica’s College, Manila

Malate, Metro Manila

2004-2009

**RELATED SKILLS**

* Research Writing Skills
* Technical Skills (MS Word, Excel, PowerPoint, Use of Internet search engines, Email)
* Programming Skills (HTML, CSS, Bootstrap, PHP, MySQL, Java)
* Fond in Web Designing

**REFERENCES**

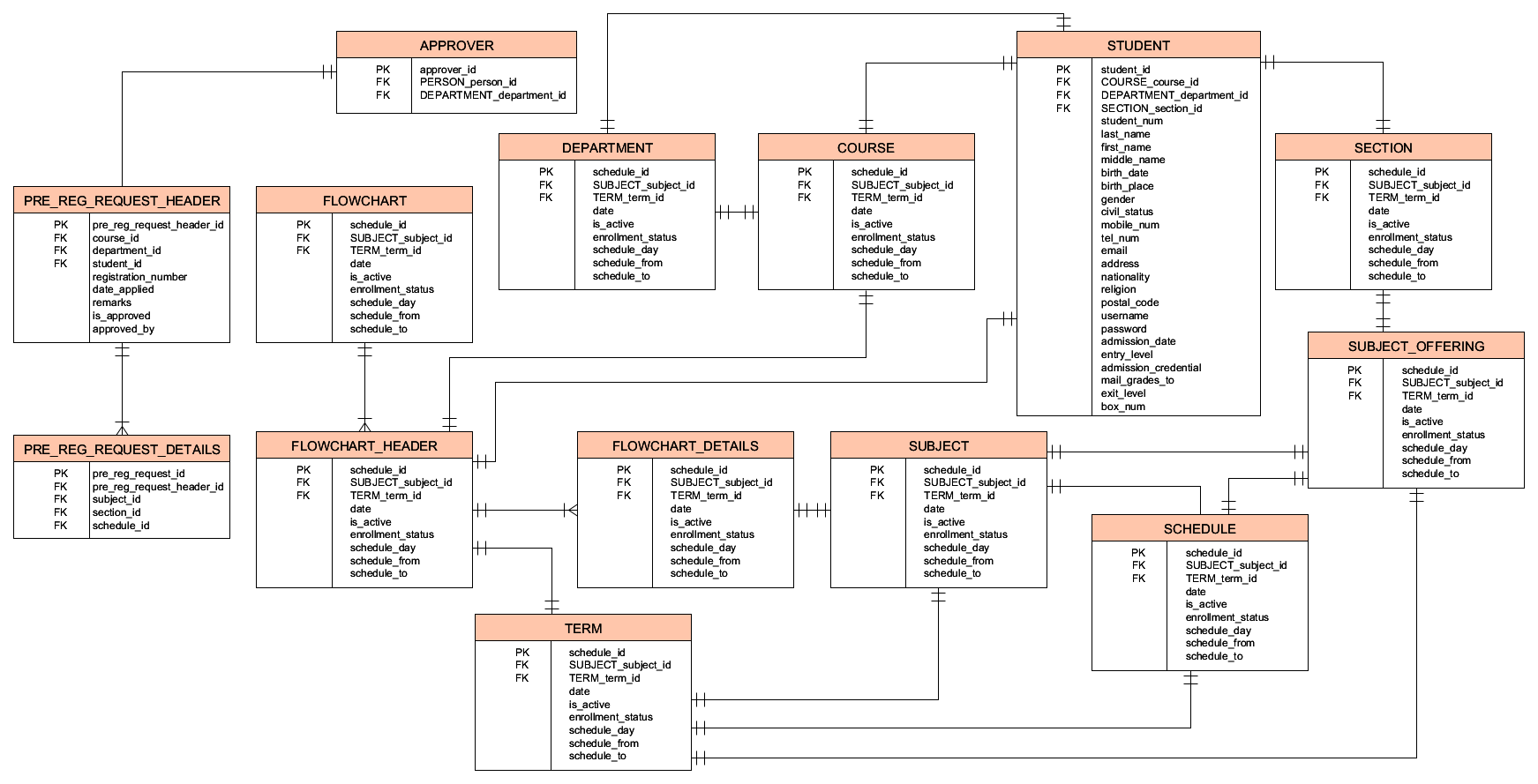
* Mr. Manuel Sanchez

College Professor, Project-Based Learning Professor and Adviser

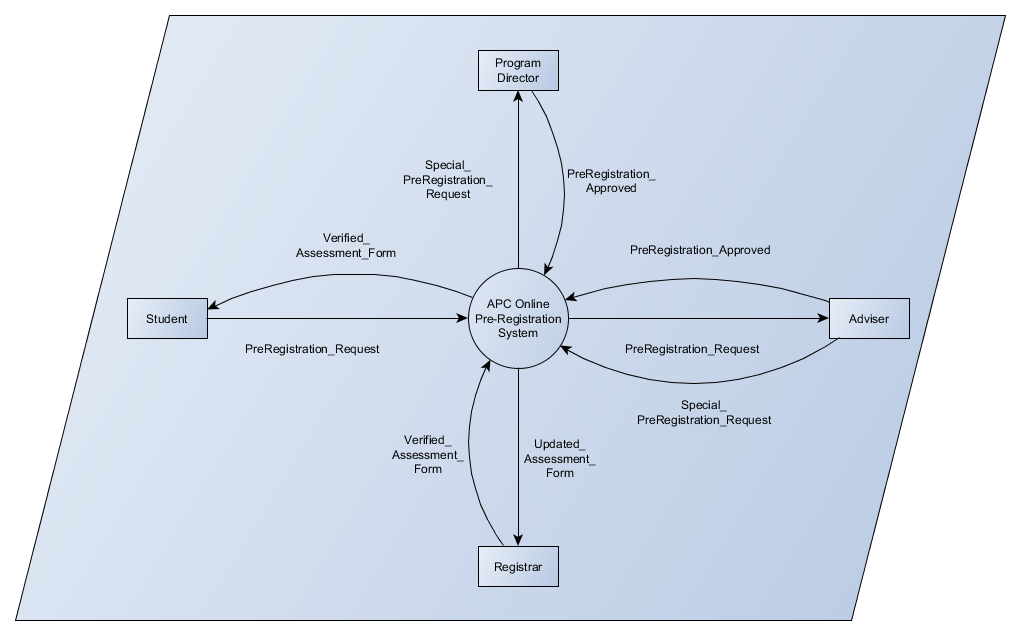
Asia Pacific College

## Diagrams

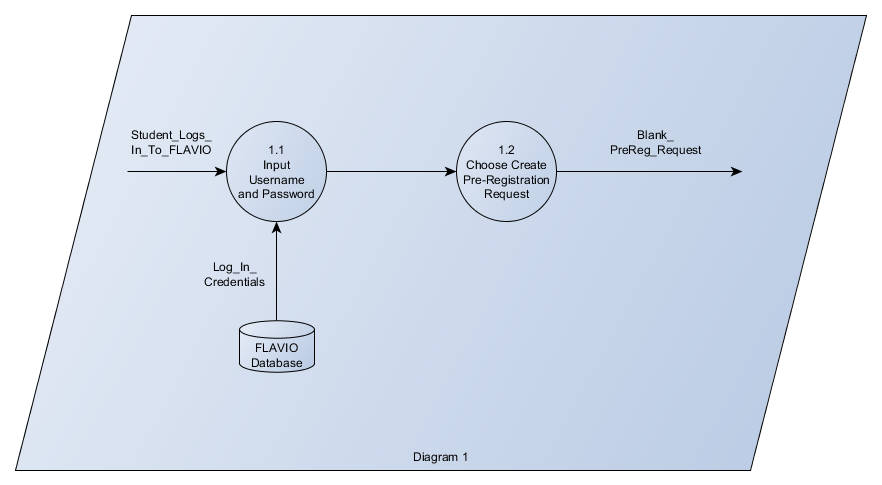
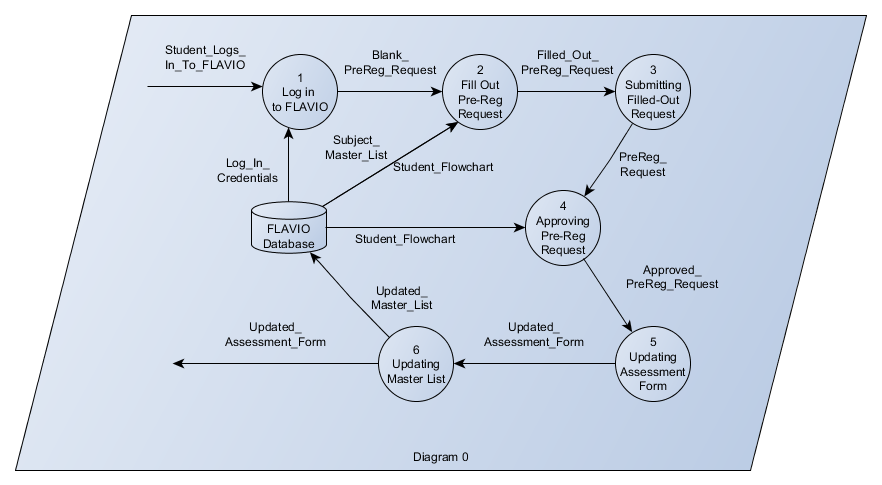
1. Entity Relationship Diagram

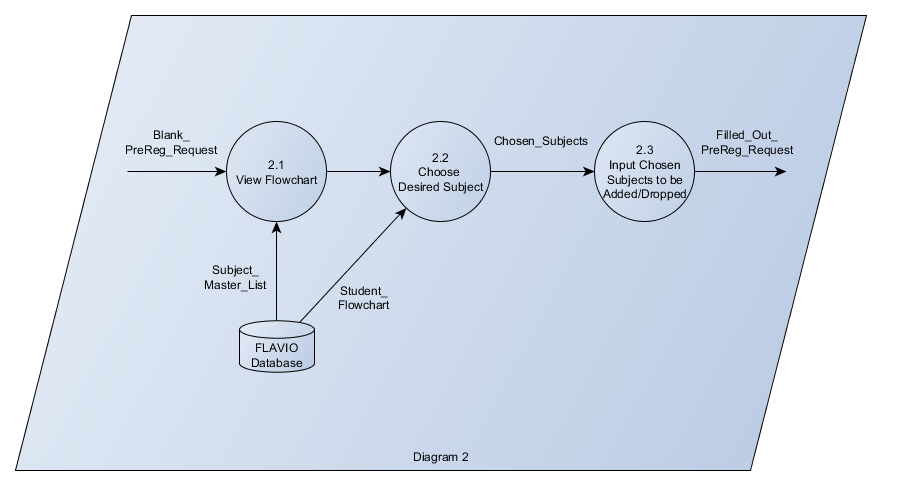


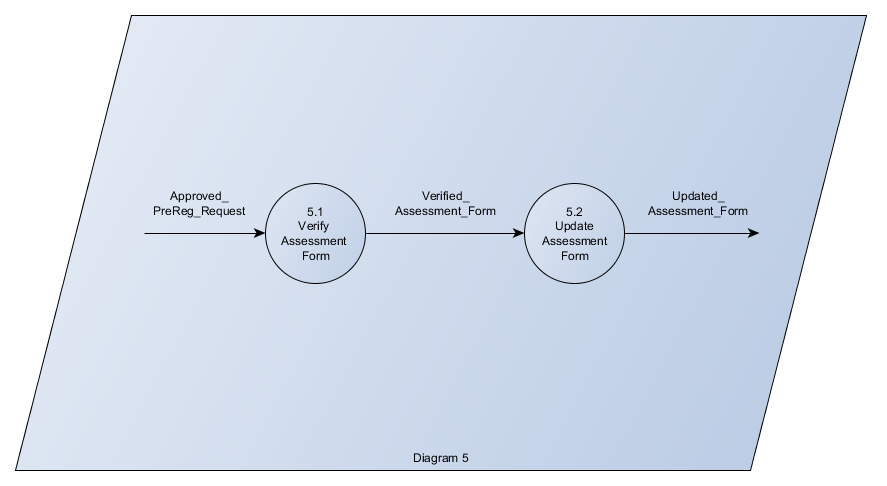
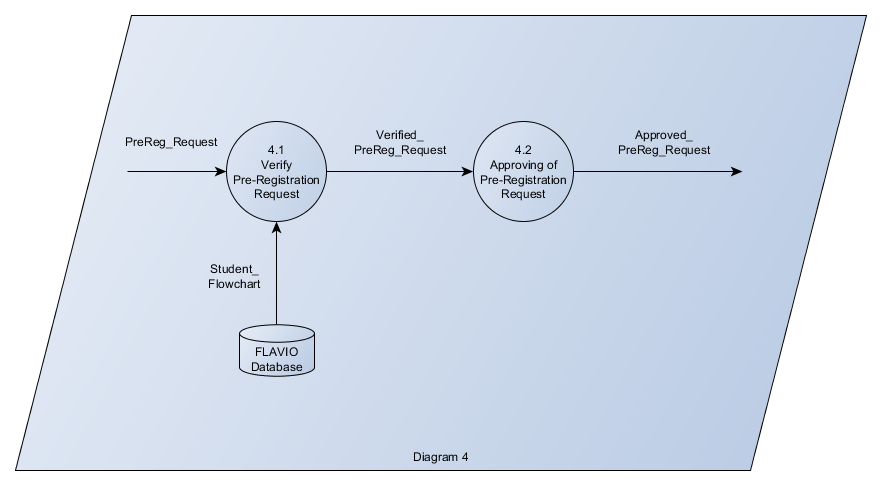
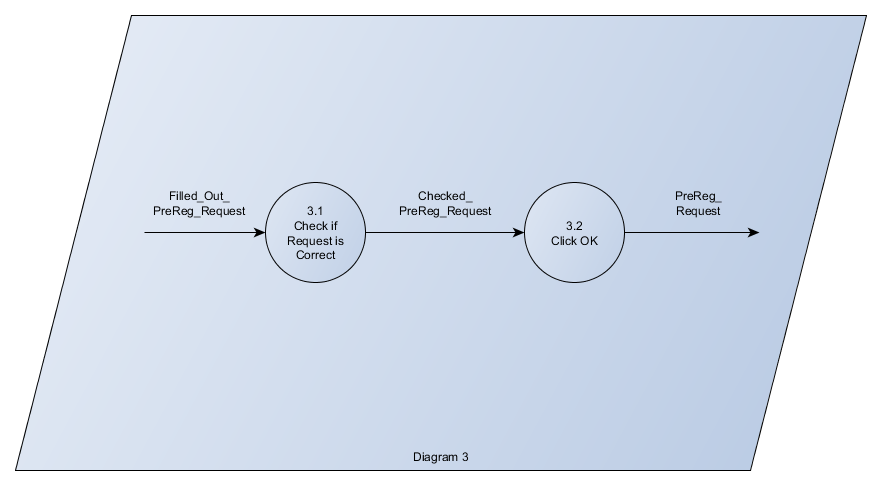
1. Context Diagram

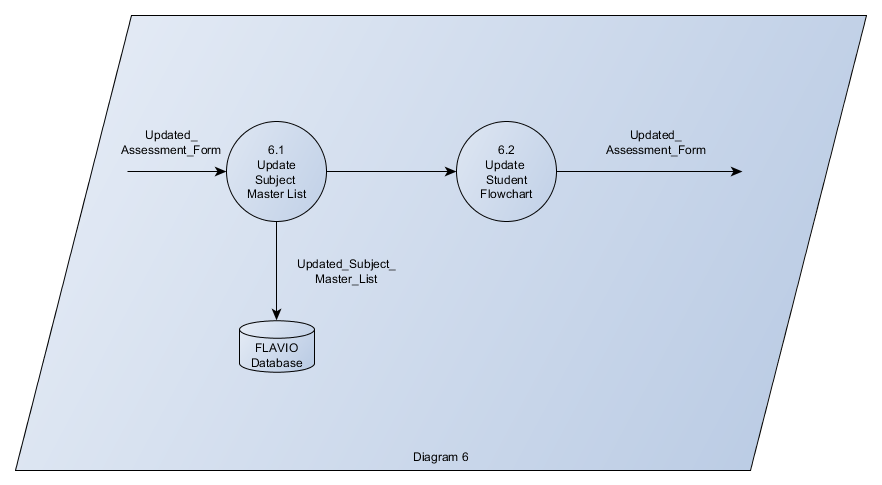


1. Data Flow Diagram

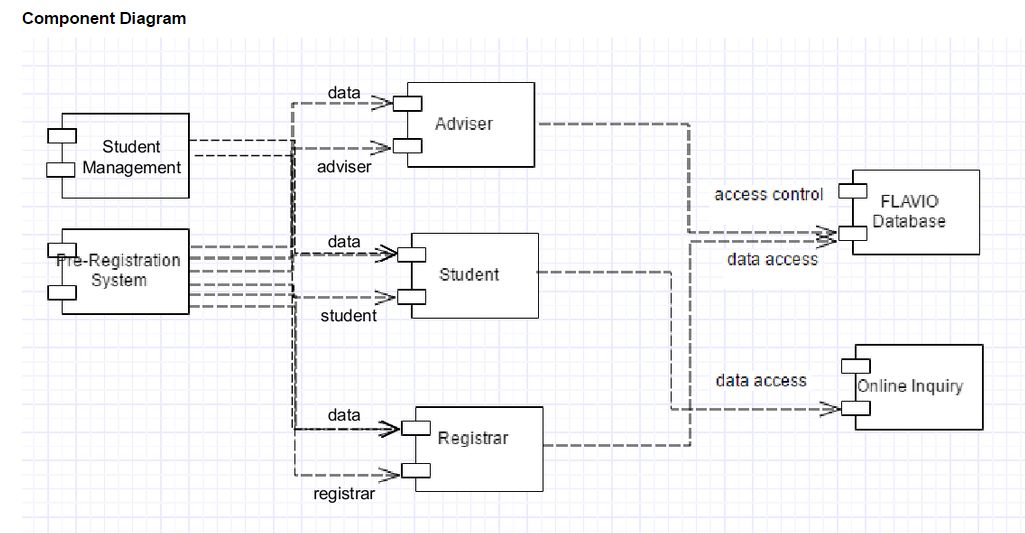




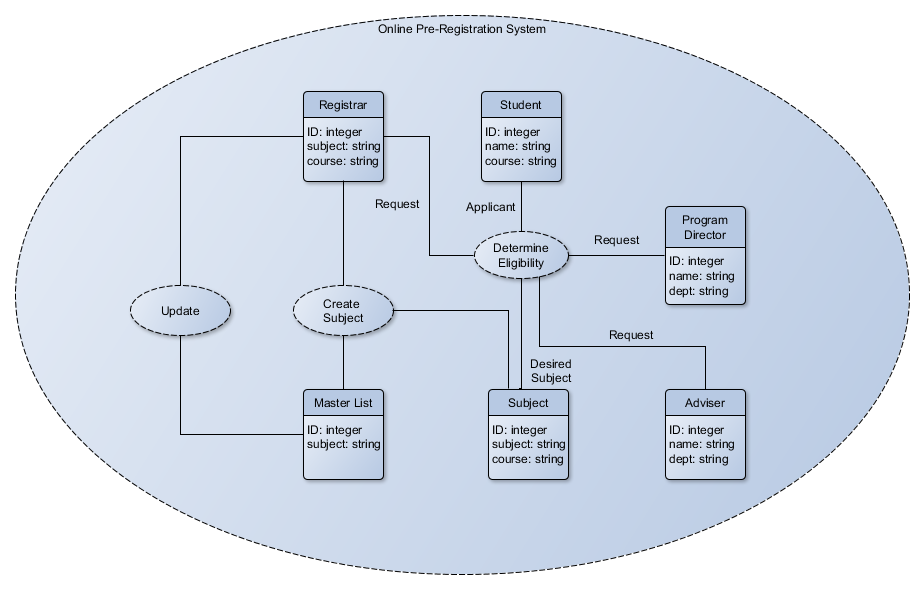




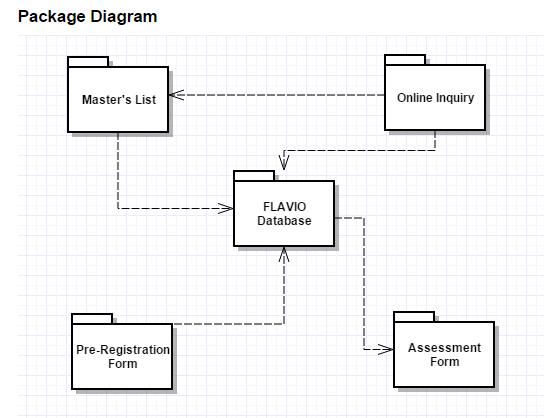
1. Component Diagram



1. Composite Diagram



1. Package Diagram



1. State Diagram

