PAL: Summative Report

Project Overview:

Title:

PAL

Members:

Daman Singh
Ju Hong Kim

Summary of accomplishments:

Created a fluid, modern and functional UI
Created a clean and functional progress report updater
Created search and advanced search functionalities
Created the ability to edit progress reports
Created a mobile site that increases productivity on the go
Created progress report editing capabilities for Admins
Created site elements editing for Admins
Created Export/Import capabilities for Admins
Created pdf view/ printing capabilities

Project Status:

The PAL website is officially fully functional as of 6/11/2014. During this project we have created a modern UI (figure 1) that allows for easier access of progress reports than ever before. This UI includes a universal menu bar on the left side of the screen that gives contextual links based on who is logged in. Part

Figure 1 - Teacher and Admin site elements, respectively

of the contextual menu includes a search feature that allows a teacher to search for a student throughout the school (figure 2) and lets an admin use the advanced search feature for more precise searches based on grade, teacher, level and more. Part of the project required us to provide the admin with the ability to change site elements and progress report content (figure 3) so, we added that functionality and then some by providing the admin with the ability to change the whole site theme. Since the sites main function is to allow teachers to update progress reports as fast and as efficiently as possible, we created a fluid and intuitive page that makes it extremely efficient for teachers to update their classes as fast as possible (figure 4). This layout literally flows, as the second a teacher updates one student, the page will automatically redirect to the next student so, the teachers can update the reports with the least amount of clicks as possible. Part of the convenience we wanted to provide the teachers with included a mobile version of this site (figure 5). The

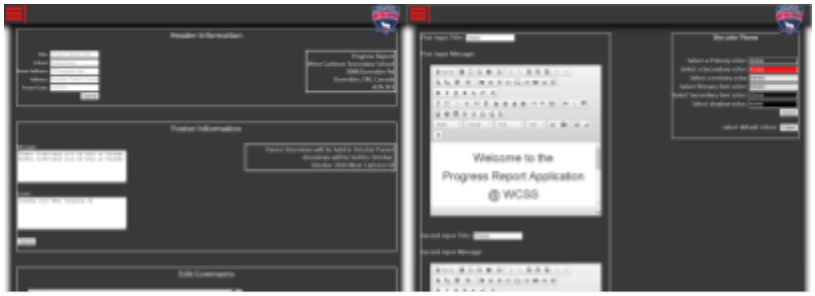
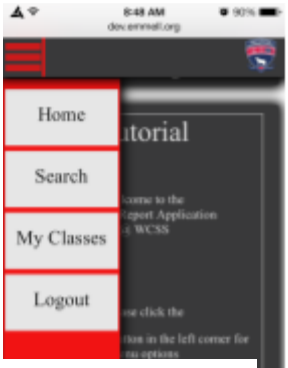
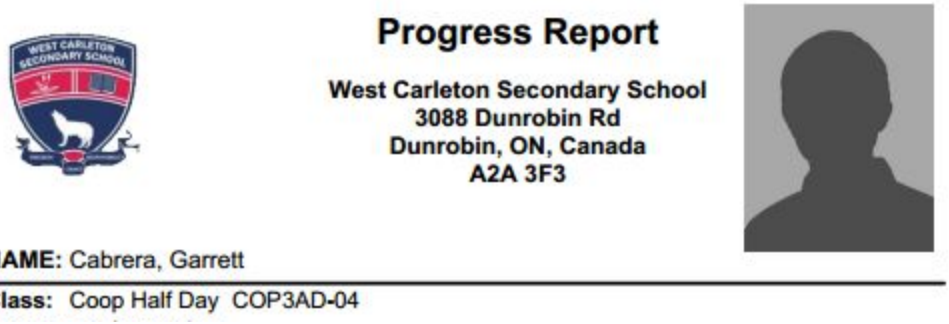


Figure 3 -Edit report and Edit site page, respectively

Figure 4 - Progress update page

mobile site can free the teachers from the shackles of desktop computers that the current programs rely on. Our site allows teachers and admins to access almost all the features of the site with the same ease as the desktop version of the site. A major feature that is mandatory to this site is the ability for the admin to export and import data sets into the website. Without this feature the site would be dataless and extremely useless. Another major feature is the ability to view and print the progress reports themselves as that is the whole purpose of the website. We have provided the admins to print out every single progress report for the whole school with one click. The teachers however get the ability to print out all reports for their own class(es) with one click or print out individual reports. The next page is what a generic report would look like.





Directory Structure:

Root Folder:

- **FPDF** -PDF Library
- **Images** -Pictures used throughout the site
    - **StudentPics** - Folder containing pics off all students
- **Includes**
    - **Toolbar.php -** HTML for static bar at the top of every page, includes school logo and links to optionsTab.php
    - **OptionsTab.php -** HTML for side menu that is accessible from the menu button in the toolbar
- **Javascript**
    - **ckeditor -** Text editor used in editSite.php, came pre-uilt
    - **jscolor -** Color picker used in editSite.php, came pre-built
    - **color.php -** Alternative CSS for site when colors are changed in editSite.php
    - **optionsTab.js -** Javascript for side menu
    - **popup.js - '**more info' popup on export/import page, came pre-built
- **PHP**
    - **file -** contains scripts and files for export and import
        - **export** - a subdirectory that contains related scripts and files for export
            - **export_csv_files.php -** retrieves data from 10 different tables and write them to 10 csv files and compresses the 10 csv files into one zip file called 'wcss_pal-tables.zip'
            - **pal-class.csv -** csv files that contains table `class` data
            - **pal-comments.csv -** csv files that contains table `comments` data
            - **pal-courses.csv -** csv files that contains table `courses` data
            - **pal-level.csv -** csv files that contains table `level` data
            - **pal-progress_report.csv -** csv files that contains table `progress_report` data
            - **pal-student_comments.csv -** csv files that contains table `student_comments` data
            - **pal-student_info.csv -** csv files that contains table `student_info` data
            - **pal-student_progress.csv -** csv files that contains table `student_progress` data
            - **pal-students.csv -** csv files that contains table `students` data
            - **pal-year.csv -** csv files that contains table `year` data

        - **wcss_pal-tables.php -** script to save zip file to computer
        - **wcss_pal-tables.zip -** zip file that contains 10 compressed csv files
    - **uploads** - uploaded files are stored in this subdirectory
    - **import_tmp_tables.php -** reads imported csv files and write them to the database into temp tables such as `temp_teachers`
    - **process_import.php -** organizes and retrieve necessary data into other tables such as first name to table `students` as first
    - **sql-import.php -** executes sql command without closing the wcss_pal connection if the sql command fails
    - **uploader.php -** script that moves uploaded tmp files to the folder **uploads**
- **pdf -** contains PDF compiler scripts and scripts needed to gather information for the PDF Progress Report Cards
    - **class_1_period.php -** Gathers all the class_id that is in period one
    - **class_students.php -** Gathers all the class_id in a specified class
    - **print_class_pdf.php -** Compiles every student's PDF Progress Report Card in a specified class
    - **report_card_maker.php -** Compiles PDF Progress Report Card for every student (should compile students classes in default year but not tested) and separates PDF Progress Report Card by 1st period
    - **student_pdf_report_card.php -** compiles PDF Progress Report Card for a specified student
    - **student_report_info.php -** gathers all related progress report information such as the progress level and student name
- **test -** test development and learning folder
- **check-teacher_class.php -** Checks if the teacher has any classes in the default year
- **check_classes.php -** Checks if the student has any registered class
- **check_progress-report.php -** Checks if the student has any existing progress report card for a specific class
- **check_student-comments.php -** Checks if student has any comments associated with their class
- **checkbox_comments.php -** Display all available comments from the database to a checkbox for teachers to select
- **class.php -** Displays all the classes that the teacher teaches
- **class_student_list.php -** To gather all student_id in a class
- **class_students_info.php -** Retrieves a list of student attending a specified class and their progress report level

- **colorPicker.php -** a script that updates the color theme of the site to the table `theme`
- **comments.php -** Retrieves all the comments and their id from database
- **connect.php -** connects to the wcss_pal database
- **default_year.php -** changes the current database default year (**year system**)
- **delete_comment.php -** deletes specified comment
- **editComments.php -** updates admin input of comment or inserts a new comment entree
- **footer.php -** Updates footer information to table `footer`
- **header.php -** Updates header information to table `header`
- **homepageContent.php -** Updates the admin input of the 2 boxes in homePage.php to table `homePage`
- **levels.php -** Retrieves all level from the table `level` and its id
- **pal-non_query-sql.php -** script to run non-query sql for wcss_pal
- **pal-sql.php -** script to run both non-query and query sql for wcss_pal
- **progress_update.php -** Updates progress report card changes by teachers
- **report_info.php -** Gather all related information for a student's progress report card of a specific class
- **sso-sql.php -** Connects to SSO Database and runs sql query + non query sql
- **student_report_cards.php -** retrieves all the student classes' class_id and the number of classes the student attends (counter)
- **student_valid.php -** checks if teacher has right to edit the student's report card
- **teacher-security.php -** checks if teacher has access a class to edit
- **update_year.php -** updates the default database year (year system)
- **user_session_check.php -** checks if the user is logged in by checking if there is a session

- **Search**
  - **advance-search-controller.php -** Retrieves a list of student that matches with the advance search options submitted
  - **advComments.php -** dropdown menu in advanced search for comments
  - **courses.php -** dropdown menu in advanced search for courses
  - **livesearch.php -** Live search bar on the search and advanced search page
  - **progress.php -** dropdown menu in advanced search for progress level
  - **search.php -** results from live search bar
  - **teachers.php -** dropdown menu in advanced search for teachers

- **sso -** Contains HTML and CSS for sso login specific to PAL
- **sso.bak -** (Connor added this folder to our project) redirect script?
- **advancedSearch.php** -HTML for the advanced search page, links to the search folder and advancedSearchResults.php
- **advancedSearchResults.php** -HTML for the results from advanced search, links to student.php
- **allClasses.php** -HTML for classes specific to teacher logged in, links to class_students.php
- **class_students.php** -HTML for students inside specific classes, links to student_report.php
- **editReport.php** -HTML for page that allows admin to change the progress report layout
- **editSite.php** -HTML for page that allows admin to change site elements including site theme
- **export.php** -HTML for page that allows admin to import and export databases and print out the whole schools progress report
- **HomePage.css** -Styling for the whole site, includes the mobile styling
- **index.php** -HTML for the homePage that is seen on login by everyone
- **searchPage.php** -HTML for the search page that includes all students, links to student.php
- **student.php** -HTML for the page that displays student progress reports
- **student_report.php** -HTML for page that allows teachers to edit the progress report of their students

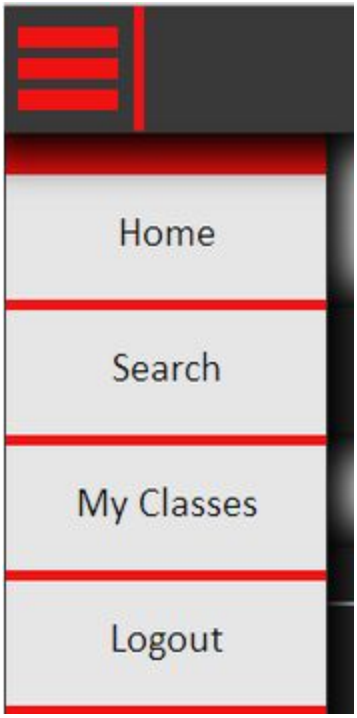
During testing, the username: daddy and password: daddy, was used for a teacher account and username: daddy_admin and password: daddy, for admin account

Project Documentation:

User guide

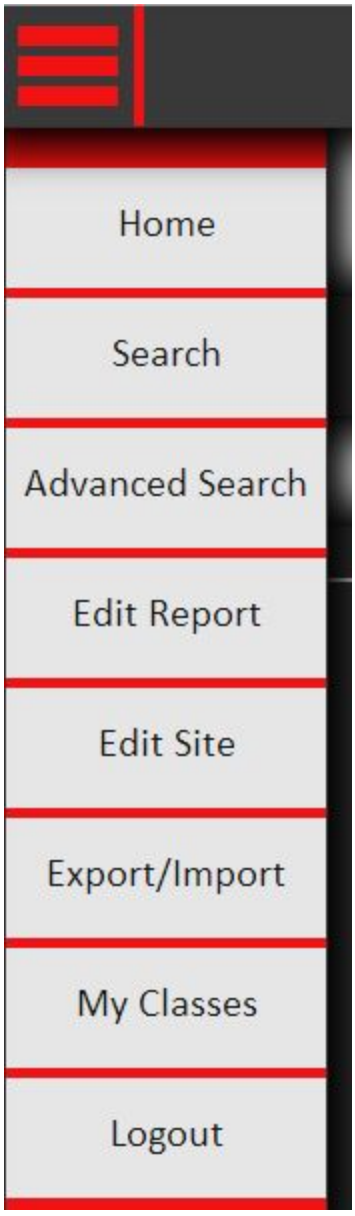
For teachers:

The site is created with minimalism in mind. In order to access the menu options, simply click the menu button (picture on right) in the top left corner. The menu that appears will allow you to go to the search page and search for any student or go to 'my classes' and see all the classes that you teach. From there, you can click on a class, then click on a student and then update their progress report. As you update the report, the site will automatically go to the next student. This allows you to be able to update all your report with least amount of clicks possible. There is also a preview button on every student's progress update page that allows you to view their report or even print it out manually. If you want to skip a student and advance to the next one, there are back and next buttons available and the ability to go to any specific student without leaving the page. And for added convenience, there are check marks next to the names of all the students you have edited a report for so you don't have to manually check each and every one.



For Admins:

The admin version of the site esthetically, looks identical to the teacher version of the site. The difference is in the menu options (picture on right) available to them. Admins get an advanced search that allows them to search for students based on their grade, progress level, comments associated, teachers and courses. The edit report and edit site pages allow the admin to edit what goes on the progress report itself and what goes on the site itself, respectively. If for some reason, admin does not like the color scheme (who doesn't like this scheme?!), they have the ability to change any aspect of the sites color through a color picker in the edit site tab. The export import tab allows the admin to export or import CSV files of the database. This tab also has a mass print option to print out a report of every student in the entire school.



Connecting to databases

There are two databases that are part of PAL: wcss_sso and wcss_pal. wcss_sso is the database that contains all the users' login information and permissions and the other database is wcss_pal which deals with most of the information necessary to PAL such as students' classes and their progress level.

To connect to wcss_pal:

```
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/php/connect.php");
```

The pdo object for wcss_pal is just $pdo

To connect to wcss_sso:

```
include($_SERVER['DOCUMENT_ROOT'].'/sso/lib/database.php');
```

*NOTE: Refer to Connor Feeley's SSO documentation for more information regarding SSO

Executing SQL Commands

Running SQL QUERY for database wcss_pal

To run sql queries regarding wcss_pal, include the code below after the sql variable containing the SQL command has been created:

```
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/php/pal-sql.php");
```

```
Ex.1:
 //An example how to run an SQL query from the PAL database
$sql = "SELECT * FROM students"; //sql variable
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/php/pal-sql.php");
/*********************************************************
Do not worry about connecting to the database, since pal-sql.php connects to the PAL database
for you
*********************************************************/
```

Running SQL NON-QUERY Command for database wcss_pal

To run non-query SQL commands for wcss_pal, include the code below after sql variable containing the SQL command has been created:

```
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/php/pal-non_query-sql.php");
```

Running SQL non-query commands work exactly like running SQL query (look above) and in fact non-query SQL commands can be run with SQL query script also.

Running SQL Query for database wcss_sso

To run SQL commands either query or non_query, include the code below after sql variable containing the SQL command has been created:

```
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/php/sso-sql.php");
```

Check if user is logged onto the SSO

To check if user is logged onto SSO, a single signage system, include the code above every page that is access restricted to only users that are logged on. If user is not logged on then the user will redirected to the PAL login page

```
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/php/user_session_check.php");
```

For restricting a content or page for only a specific user, please refer to the SSO documentation. Note: There may be an existing SSO file that checks if user is logged on

Retrieving Information Regarding a Specific Student in a Specific Class

To get data regarding a student's progress report or their student number(oen), include the code below:

```
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/php/report_info.php");
```

Remember that a variable $class_id and $stud_id is REQUIRED for the script to retrieve information regarding the student's data in a specified class. Please look at the example below for more detailed information

Ex:

```
/*************************************************
$stud_id is the variable that specifies the student you wish to retrieve their information
$class_id is the variable that specifies the class of which you want the student information concerning the class specified
*************************************************/
$stud_id = 10;
$class_id = 2;
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/php/report_info.php");
echo $info['level']; //prints out the progress level of the student in class_id 2
/*********************************************************
Notice how all the variables are associative array
There are several associative arrays for different purposes
$info is an array that contains the course code, course name, OEN, first name, last name, level_id, level…
$comments is an array that contains the comment_id and comments of the student's course progress
Please look at the report_info to see the associative element array  names
*********************************************************/
```

Programming PDF Files

Anytime when creating a PHP file for the purpose of creating a PDF file, the FPDF library must be included at the start of the file:

```
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/FPDF/fpdf.php"); //connects to FPDF library
```

FPDF library is a free and useful library to program PDF files and is very simple to learn. Please refer to the official FPDF website: http://www.fpdf.org to learn how to learn FPDF syntaxes.

PAL Database

The PAL database, wcss_pal, has many tables, and each table contains data for a specific reason. The database is a relational database (Relational DBMS) so it relies heavily on other tables to retrieve related data which is connected together by a "Key". It is highly recommended for the person in charge of the database has great knowledge and experience of relational database and SQL Join commands. The PAL database relies on 3 big and major tables: **class**, **student_progress,** and **progress_report**. The big tables will be called the table hub since these tables contains many keys to connect to many related tables.

->class: a "table hub" which stores data of classes such as the key for the teacher information and the key for course information and the class period. Contains a primary key called class_id.

-The table class is consists of only numerical values since most of the columns stores the id number (key) of a class, teacher, period, and year.

Related Tables(3):

1. wcss_sso.users (Key: teach_id (the key for sso.users is id)) //identifies the teacher info
2. wcss_pal.courses(Key: course_id) //the class course code & course name can be found
3. wcss_pal.year(Key: year_id) //the year and semester of the class is stored in table year

->**comments:** contains data regarding all the comments available for the teacher to use when assigning comments to their student

As you may have known, comments can only be added,edited, and deleted by the admin. Teachers will not be able to change the comment tables if they have no admin permission.

-comment_id is unique (also auto incremented) and is referred to when the admin is editing or deleting the comments (key)

->**courses:** contains data regarding the course name and the course code

-Courses should be added automatically when importing files from Mama and the Ride database, if the course does not currently exist in the wcss_pal database already.

Related Tables:

1. **class** (Key: course_id) //table hub that connects teacher info, course info, and the class year, sem, and period

->**footer:** contains text for the PDF Progress Report Card message(i.e. Parent-Teacher Interview Date) and footer

->**header:** contains text information for the header of the Progress Report Card such as the street of the school, Postal Code, and etc

->**homePage:** stores data for the 2 customizable boxes that only admin can edit in the homePage that can be used as a notification, message, or a tutorial.

->**level:** contains all possible progress levels available in the database (i.e. Meeting Expectations)

-level_id is unique (key) and its related table is progress level

->**progress_report:** The table contains 3 keys that can be used to find the student's progress level, comments, and more.

Related Tables(3):

1. **student_progress** (key: progress_id)
2. **level** (key: level_id)
3. **student_comments** (Scomment_id)

->**students:** contains the student's basic info: their first name and last name

Related Table(2):

1. **student_info** (key: student_id) //used to find the OEN
2. **student_progress** (key: student_id)

->**student_comments:** contains the comments for the student in a class

Related Table(2):

1.**progress_report**(Key: Scomment_id)
2.**comments** (Key: comments)

->**student_info:** contain students's OEN

Related Table(2):

1. **students** (key: student_id) //used to find student name
2.**student_progress** (key: student_id) //a table hub that connects class information, student information, and progress information together

->**student_progress:** a table hub that connects the student progress report, class information and student information together

Related Tables(4):

1&2. **students** + **student_info** (key: student_id) //student name + OEN can be retrieved
2. **class** (key: class_id) //table hub that connects course info, teacher info, and class year and sem. The class period can be retrieved
3. **progress_report** (Key: progress_id) //table hub that connects progress level and student comments. Interview Request can be retrieved

->**temp_classes, temp_teachers, temp_teachers -** stores imported data from Mama and Ride files. The import_process.php requires the temp tables to process required information from the temp tables to its according table

->**theme:** stores the color theme of the site

->**year:** stores the year and semester of classes registered in PAL. Also contains a default column which is still a developing function in PAL (Default will be explain further in the documentation).

Related Tables(2):

1. **class** (Key: year_id)
2. **student_progress** (Key: year_id)

Note: In table student_progress, the column` year_id` is not needed since `class_id` can be used to find the year_id and connect the year table.

Look at the diagram below to have a more clearer and visual representation of the PAL's Relational Database.

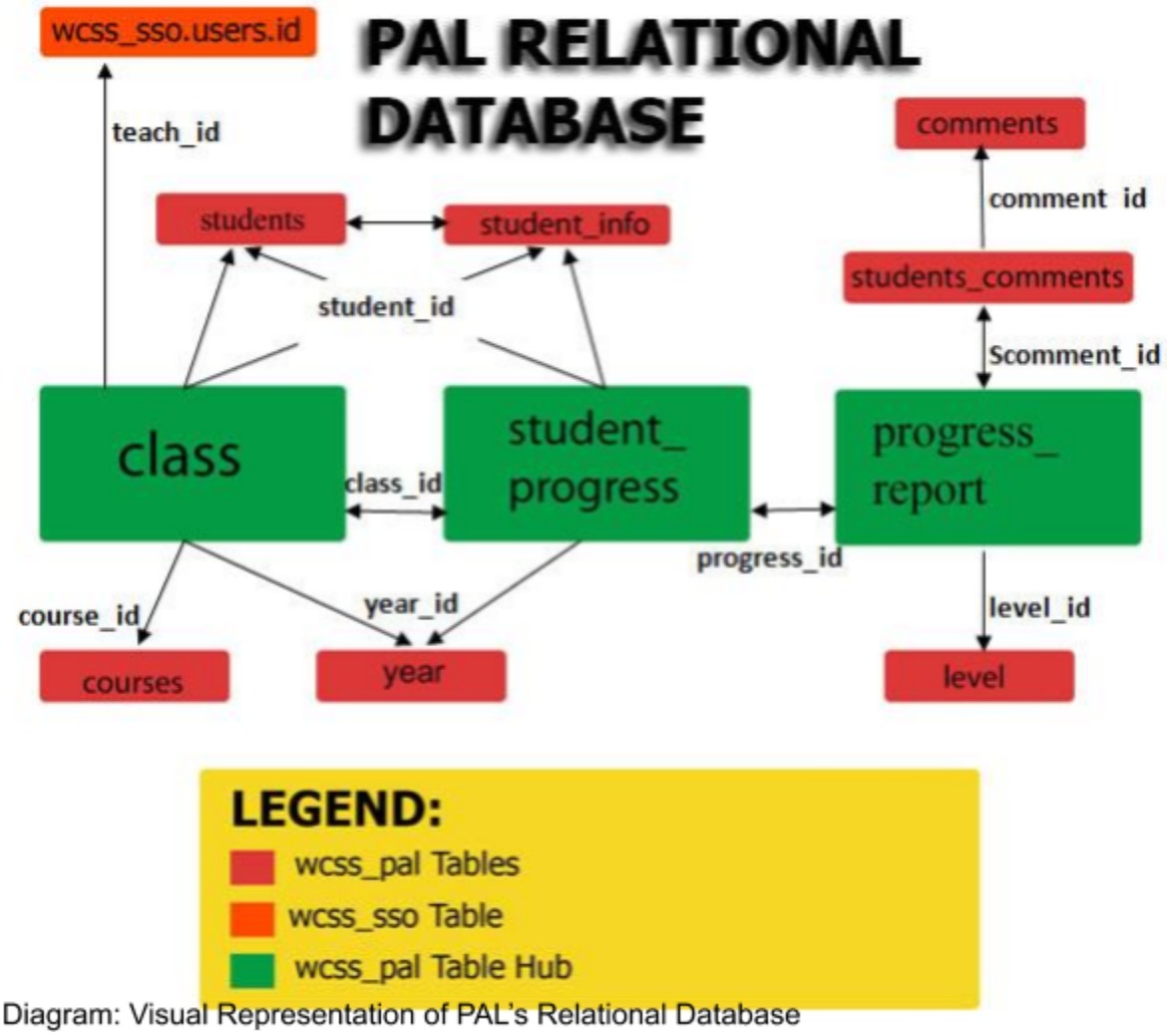


Diagram: Visual Representation of PAL's Relational Database

KEYS: A key is a field to sort data. Keys can be known as sort key, primary key, and index.

-Foreign Key is a key that identifies a row (record) in another table.

Ex: Foreign Key in class:

course_id, student_id, teach_id are all Foreign Keys because those keys identifies a row in another table such as course_id will identify the course code and course name of the id

YEAR SYSTEM

One of the detailed feature of PAL that was outlined at the beginning of the project was having PAL to have multiple sets of data from different years. Originally, we wanted PAL to be able to contain multiple years of progress report card so that admin can view previous progress report

cards and not have the database to be cleared each semester. The year system was suppose to track the year of the registered classes and have a default year where all classes in that year would be able to be edited. So far the current progress and status of the year system is unknown. There has been no data yet given to PAL and has not been officially tested. There is a page where admin can edit the default year and does work. There has been an unofficial testing of the year system, where the default year was changed. The results was what we would have expected, no classes was available for teachers to edit since there was no data for the class in that year and semester. However, it is not yet confirmed if the year system actually works.

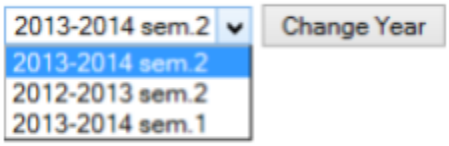


Figure 6- Changing the default year

The default_year.php is a page where admin can change the default year of the site. As stated before that it does update the default year but admin should stray off from accessing this page due to the feature being not completed and officially tested. The current features of the page is that it auto selects the default year and displays all year information from the database.

There are two methods to enter a new year. The 1st being the database administrator manually inputs a new year into the table `year` through SQL or GUI input feature that is available in phpMyAdmin. The 2nd method is the importing files. Any new year information that exists in the imported data will automatically be written to the `year` table.

For the year system to be completely functional, some scripts have to be modified to retrieve the default year class and progress report information such as in check_class.php:

```
//check_class.php
include ($_SERVER['DOCUMENT_ROOT'] . "/PAL/php/connect.php"); //connects to wcss_pal database

$class_valid = 1; //student has registered class

try {
     $class_result = $pdo -> query ("SELECT class_id FROM student_progress WHERE student_id =
$stud_id");
    $row_count = $class_result -> rowCount();
...
```

The SQL command query "`SELECT class_id FROM student_progress...`" needs to be changed to

```
"SELECT student_progress.class_id FROM student_progress
INNER JOIN year ON student_progress.year_id = year.year_id
WHERE student_id = $stud_id AND year.year_default = 1"
```

Many php scripts that contains SQL queries regarding class information and progress report card information will required to be changed to retrieve information for the default year only.

ADMINS please do clear out the data for the tables:

class, student_progress, student_comments, and progress_report

every time at the start of a new semester just to avoid bugs and possible errors in the program. If admins are concerned and want the data for the progress report cards of the year:

A. Save all the progress report card PDF files through the progress report card export function

B. There is an available feature in export.php where the admins can download a compressed csv zip files of student, progress report card, and class information. Admins can import those csv files into the PAL database. However all previous data will have to be cleared from the database.

EXPORT & IMPORT

Import Process-

The current PAL import process goes through 3 files to break down the memory being used for each page. At the beginning of importing, admin are to choose 3 files (2 csv files from Mama and 1 file from Ride). As the admin clicks the Import Button, the files are sent to the server into as a temporary file. This is when the programmer who is in charge of the import function comes in. The programmer's job is to move the temporary submitted file to it's permanent location and process the information to it's appropriate tables. Imported files must be files from Mama and Ride. The csv files for teacher and students have to be from Mama and the file for classes is form Ride. Any changes on the Mama or the Ride structure such as the table name or maximum lengths of each column must be adjusted to all the import files: uploader.php, import_tmp_tables, and process_import. In every file, the first line of the file will always be ignored, since the first line is usually the table names.

Process:

1. **uploader.php** - This is where the files are moved to a permanent location in the server directory PAL under /var/www/PAL/php/file/uploads.

    1. The script looks if the files have been sent to the temporary folder
    2. Checks if the file extension is '.csv'
    3. Checks if each file exceeds 1mb
    4. Moves the file into "/var/www/PAL/php/file/uploads"

2. **import_tmp_tables.php** - This is where the data from the imported files are sent to the temp tables in wcss_pal database.

1. The csv file for class data sequence is analyzed to check if the file is truly the csv class file. The csv data sequence check only analyzes the length of each column, discluding the first column since the first column is usually the column names.
2. Teacher csv data sequence is analyzed
3. Student csv data sequence is analyzed
4. After all 3 files have been analyzed and have passed the sequence security, all of the rows in each file with the exception of the first row will be inserted into the temp tables in wcss_pal.

3. **process_import** - This is where all the data are processed a sorted into its appropriate tables such as a student's name would be placed in the table `students`.

1. Checks if there are any courses not registered in the database. If there are any courses not registered in the database, the course will be added to the database in table `courses`.
2. Checks if there are any year entrees that does not exist in the database. Any non-existant year entrees will be added to the database in table `year`.
3. Adding classes into the database
    1. Retrieves the teacher's SSO id (should be their teacher number)
    2. Checks if the class exists using the teacher's SSO id to identify if the class is registered in the database
    3. If the class does not exist, then the class would be added to the database in table `class`
4. Adds students who are not registered already in the database table `students` and their OEN into table `student_info`
5. Adds the students into the class if they are not currently registered in the class

Export Process

Export is basically backing up the PAL database but since the year system is not yet complete, exporting files to csv files are useful to be referenced again. The export only exports a compressed zip folder containing csv files. The export does not export PDF Progress Report Card, since PDF extensions in browsers have the option to save the PDF file. The export process has basically 2 steps: writing to the file and compression of files into one zip folder. Please note that if a table structure does change, please **EDIT** the table names in export_csv_files.php.

In **export_csv_files.php:**

1. Retrieves records (rows) from 10 tables and write to 10 different csv files.

Tables that are being written to csv files:

students, student_info, course, comments, year, class, student_progress, progress_report, level, and student_comments

2. Compression of 10 csv files to one zip folder called wcss_pal-tables.zip located in "/var/www/PAL/php/file/export/wcss_pal-tables.zip"

Project Future

PAL is currently functional and tested for various situations. However, there are many areas where PAL can be improved and worked on.

1. **Year System:**

The year system was supposed to be one of the big highlights of PAL, where it was possible for admin and teachers to view student's previous Progress Report Cards from every semester since entering high school. I (Ju Hong Kim) really do wish to see the year system to be added into PAL and hope that the person who picks up the project will complete the year system. The year system as mentioned before is not completed and is not officially test either. In order to complete the year system, I believe that all PHP file related to a student Report Card or retrieving data regarding a class will need to be modified to retrieve data on the default year of the PAL year system. New files will have to be created to retrieve data from previous years.

2. **Security:**

Security is needed for any program, especially with PAL. PAL needs a good security to prevent outsiders and possibly users from accessing with private data such as a student Progress Report Cards. Although security is generally and should be dealt by the SSO project members, since SSO deals with the login/signage system, there are some security that can be added to PAL. Some security issues of PAL is Javascript and SQL injection. PAL is very vulnerable to injections as many sites did during the beginning of the creation of PHP. Javascript and SQL injection can lead to leakage of private data and damages to the database such as deletion of tables in the PAL database. We were not very familiar with injections but we did knew that PAL is vulnerable to Javascript and HTML Injection. PAL does has some sort of security but is still not secured and the security only covers the ability of other teachers from editing a student's report card (teacher-security.php and student_valid.php). The site hasn't been fully tested and should be in beta testing to receive feedbacks on the security. All pages that only admin should be able to use should have a restriction if statement to check if user has admin privileges. PAL has quite a fair number of query links which makes PAL more vulnerable to injections since anyone can change the values of the query. A new security should be added to prevent injection such as having a script to check for quotes or any suspicious values. PAL uses regular

PDO query to run SQL commands, but I have heard some programmers stating that prepared statements have better security. If prepared statements do have better security compared to regular PDO query, than PAL should change its method of running SQL commands to prepared statements. What PAL lacks is the ability to prevent users and guest from viewing the directories and subdirectories of PAL. For developing purposes, there was security made to prevent people from seeing directories.

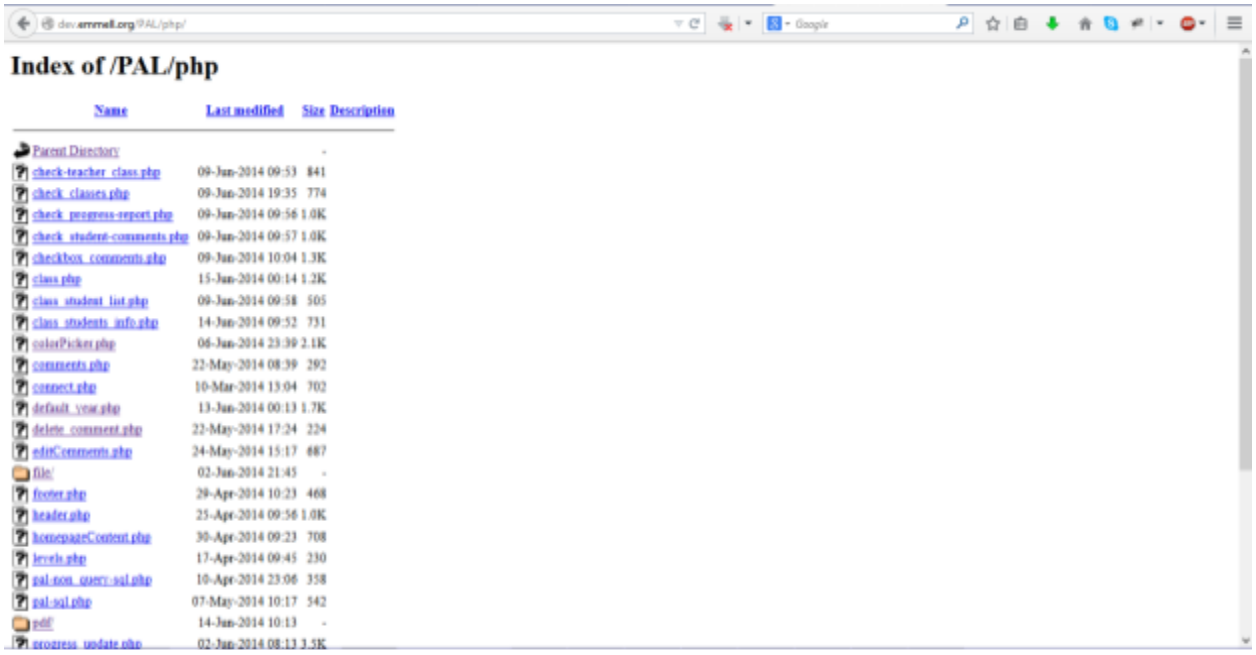


Figure: Anyone can view all files in the subdirectories and run any php files in PAL if he or she knows the file structure of PAL (Directories) and if the PHP files is not secured to prevent outsiders from accessing

Every file should check if the person is logged on (see documentation above to see how) to prevent from random people running PHP scripts. The current PAL does not have any security in those aspects other than the main pages of PAL (i.e. homePage.php) and every script in the year system (i.e. default_year.php).

3. **FINISH PAL**

PAL stands for Progress and Liabilities, meaning that the program was suppose to deals with Progress Report Cards and Liabilities. However, due to the amount of time given, the time was allocated for Progress Report Card part of the program. This means that PAL is incomplete because only P was completed and still missing the 'A' and the 'L', since the Liabilities have not been completed or yet even started. I am not keen on finishing on PAL if I was to ever take up the project ever again because there are some improvements needed for the Progress Report Card side but it would be ideal for PAL to be completed since liabilities is a very useful part of PAL.

4. **Import Functionality**

PAL import function has been tested and does work but there are some minor improvements that could happen. The import of PAL does not have a great UI in that the current PAL program will just print lines of SQL statements which is also a security risk if a random person was to see this by coincidence. The import does not tell users if the import has been successful and is not visually attractive. The import does take a lot of time to process the information and can be more efficient. There has been some attempts to make the import process more efficient but the process only sped up to 2-3 seconds and still is very inefficient. Some security risks for import is a person could write a upload script that connects to all the import process files and upload random data that matches with the data sequence, however it is unlikely since the person will require to know the directory and paths of the server and require certain permissions to be able to do so. Data sequence is also a possible security risk because any csv files that match with the data sequence can be imported to the database. This would mean there would be random data in the database that makes no sense at all. One functionality of import that I would highly recommend is updating data. It would be very useful if the import would update information along with adding data to the database. If a student switched out of a class or if there was an emergency and the teacher for the class changes, then update would be a very useful function.

5. **Efficient:**

Developers may have seen how messy the code was such as blobs of commented code and some of the inefficiency in various areas of the program. There are some files that may have extra functionality that are not being used and a lot of unnecessary codes. The database table structures can be cleaned a bit such as in the table `student_progress`, where the column 'year_id' is unnecessary since the 'class_id' can be used to connect to the table `class` which contains the key, 'year_id', in which connects to the table `year`. The column `grade` is very useful but the imported files does not give any information of the student's grade. It would be ideal if there was a way to receive the student's grade because it is an field in advance search. Having grade will help admin to narrow down results by grade. But if there is no possible method to retrieve student's grade, then the column 'grade' and the advance search field grade would be useless and inefficient.

6. **Mobile:**

During the end of the development process, we found that we had some extra time so we developed a mobile experience for the site. Currently, we are using media queries to check for a device width of 480px , which would contain majority of the smartphones available in the market today but, just recently there have been QHD phones being released with crazy high resolutions and phones with 5.5" screens are becoming more common. These new developments in technology are not being accessed in our code currently. So, as part of

possible future development, a better mobile experience that takes into account the changing technologies would be a place to start.

Note to Potential Developers or Administrators:

Dear potential developers or administrators,

PAL was a big and fun project. There was a lot of hard work done by the PAL team and a lot of changes were made throughout the semester. As mentioned in the documentation and project future sections of the program, there are some bugs and security risks in PAL. If the project was ever to be used, I would like the program to go through various testings to minimize bugs and security risks before the program is runned in school servers. It would be terrible if the program was runned without having being tested for bugs and security vulnerabilities. It would be great if the year system was to be completed and PAL undergoing through major security updates and improvements to make PAL a better program. The key features of PAL from my perspective is the capability to be used for mobile devices, great flexibility in customization of the site, and the site stylization (my partner did a excellent job in making it happen). Although these features give off a fresh and mobility of the program compared to the Ride and is important for the users, the key of the program is its functionality. The program being my partner and my first big web program we ever did, we did a great job. The program does work and is functional as stated many times in the whole document, there are some concerns about the program such as the security and bugs in program. This is due to the lack of time given and my lack of planning and knowledge. I hope that PAL will undergo improvements to be used by schools soon.

Ju Hong Kim
A member of PAL