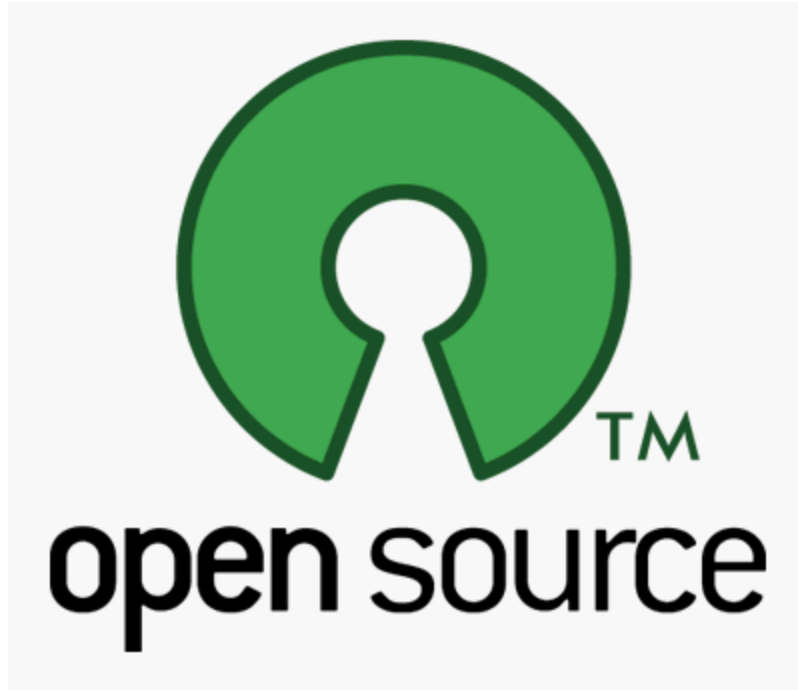


CS4354 Project Report

Brian Fairbanks - Derrick Franco - Noah Metzger



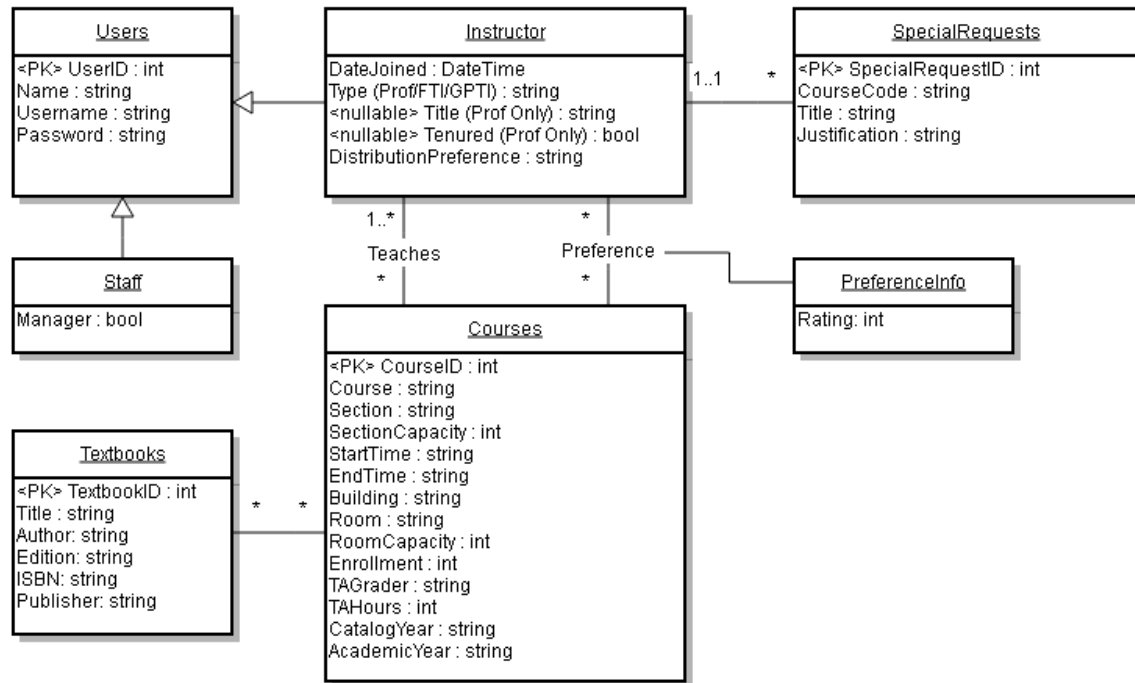
Open Sourced for all to use / see
<https://github.com/dfranc3373/CSProject>

Table of Contents

Part A: UML data modeling	-	Page 3
Part B: Relation Schemas	-	Page 4
Part C: SQL Code and Design	-	Page 7
D. Program Design	-	Page 9
Part E: Design	-	Page 11
Part F: Contributions	-	Page 13
INSTALLATION	-	Page 14

Blog at: <http://brdeno.wordpress.com>

Part A: UML data modeling



Part B: Relation Schemas

Our Database Design implements both foreign key constraints and foreign key tables to allow for a wide variety of data manipulation in a safe way via queries.

The foreign key tables were used on most many to many relations as they give a connection between tables over multiple rows.

users - Grants login privileges to the system. Is extended by Staff and Instructor.

Attribute	Type	Description
user_id	int(255)	Unique identifier for the user
username	text	For login
password	text	For login
firstname	text	Personal info
lastname	text	Personal info
email	text	Personal info

staff - is extended from a user_id to allow staff on a user login

Attribute	Type	Description
staff_id	int(255)	Implementation Purposes
user_id	int(255)	Link to user
is_manager	int(1)	Modifies privileges

instructors - extended from a user_id to allow instructor on a user login

Attribute	Type	Description
instructor_id	int(11)	Unique id
user_id	int(11)	Foreign key
type	text	Type of instructor
title	text	Title
tenured	int(1)	Boolean if they are tenured
distribution_preference	int(11)	Distribution preference for work load
date_joined	text	Date they were entered into the system

teaches - uses foreign keys to allow multiple professors to teach multiple classes

Attribute	Type	Description
user_id	int(255)	Foreign id

course_id	int(255)	Foreign id
-----------	----------	------------

preferences - makes a list of instructors preferences for their course

Attribute	Type	Description
instructor_id	int(11)	Foreign id
course_id	int(11)	Foreign id
preference	int(1)	Preference for the course

special_request - list that is filled by professors

Attribute	Type	Description
special_request_id	int(11)	Primary key
user_id	int(11)	Foreign Key
course_code	text	Course Code
title	text	Course Title
justification	text	The justification of making this class for the department

courses - courses for the department

Attribute	Type	Description
course_id	int(255)	Primary Key
course	text	Course Title
section	text	Section
section_capacity	int(11)	Section Capacity
start_time	text	Start Time
end_time	text	End Time
building	text	Building
room	text	Room
room_capacity	int(11)	Capacity
enrollment	int(11)	Enrollment
ta_grader	text	Grader Name
ta_hour	int(11)	How many hours the TA will work for the class in a week
catalog_year	text	What catalog
academic_year	text	What year
course_semester	text	What semester

textbooks - textbooks that can be used by any course

Attribute	Type	Description
textbook_id	int(11)	Primary Key
title	text	Title of Book
author	text	Author
edition	text	Edition
isbn	text	ISBN
publisher	text	Publisher

course_textbooks - foreign keys to allow multiple textbooks for any course

Attribute	Type	Description
course_id	int(11)	Foreign Key
textbook_id	int(11)	Foreign Key

Part C: SQL Code and Design

Examples and ideas for queries that will be written to solve problems

1a / c. INSERT INTO `users` (`username`, `password`, `firstname`, `lastname`, `email`) VALUES ('username', 'password', 'firstname', 'lastname', 'email');

INSERT INTO `instructors` (`user_id`, `type`, `title`, `tenured`, `distribution_preference`, `date_joined`) VALUES ('2', 'professor', 'assistant', '0', '1', 'spring 2013');

1b. INSERT INTO `database`.`courses` (`course_id`, `course`, `section`, `section_capacity`, `start_time`, `end_time`, `building`, `room`, `room_capacity`, `enrollment`, `ta_grader`, `ta_hour`, `catalog_year`, `academic_year`) VALUES (NULL, 'test', '1', '10', '10:00', '11:00', 'Computer Science', '400', '10', '0', 'New Grader', '1', '2013', '2013');

INSERT INTO `database`.`teaches` (`user_id`, `course_id`) VALUES ('2', '1')

2a. UPDATE `preferences` SET preference = '1' WHERE `instructor_id` = '1' AND `course_id` = '1';

2b. UPDATE `instructors` SET `distribution_preference` = 'spring' WHERE `instructor_id` = '1';

2c. INSERT INTO `special_request` (`user_id`, `course_code`, `title`, `justification`) VALUES ('1', 'test', 'test');

2d. INSERT INTO `textbooks` (`textbook_id`, `title`, `author`, `edition`, `isbn`, `publisher`) VALUES (NULL, 'test', 'test', 'test', 'test', 'test');

2e. SELECT * FROM `courses` WHERE `course_id` in (SELECT `course_id` FROM `teaches` WHERE `user_id` = '1');

3a. //SELECT YEARS IN PHP

 //IN PHP foreach(YEARS as \$y)

 \$courses = SELECT * FROM `courses` WHERE `academic_year` = "\$y" AND
 `course_id` in (SELECT `course_id` FROM `teaches` WHERE `user_id` = '1');

3b. \$proffesors = SELECT * FROM `instructors` LEFT JOIN `users` on users.user_id = instructor.instructor_id;

 //IN PHP foreach(\$proffesors as \$p) {

```
$sum_ta_hours = SELECT SUM(ta_hours) FROM `courses` WHERE `course_id` in  
(SELECT T.course_id FROM `teaches` T WHERE T.`instructor_id` = "$p");
```

```
$sum_enrollment = SELECT SUM(enrollment) FROM `courses` WHERE `course_id` in  
(SELECT T.course_id FROM `teaches` T WHERE T.`instructor_id` = "$p");
```

```
echo $sum_enrollment / $sum_ta_hours;
```

```
}
```

3c. SELECT * FROM `courses` WHERE `instructor_id` = '1'

3d. //SELECT YEARS IN PHP

```
//IN PHP foreach(YEARS as $y)
```

```
    $courses = SELECT * FROM `courses` WHERE `academic_year` = "$y" AND  
    `course_id` in (SELECT `course_id` FROM `teaches` WHERE `user_id` = '1')  
    ORDER BY `academic_year`
```

3e. SELECT * FROM `preferences`;

3f. SELECT * FROM `textbooks`;

3g. SELECT * FROM `courses` WHERE `course_semester` = "SUMMER I" OR
`course_semester` = "SUMMER II"

3h. //SELECT YEARS IN PHP

```
//IN PHP foreach(YEARS as $y)
```

```
    SELECT SUM(enrollment) FROM `courses` WHERE `academic_year` = "$y"  
    AND `course_semester` = 'Fall' OR `course_semester` = 'Spring'
```


D. Program Design

The Database project was designed in PHP / MySQL using the smarty templating system to separate design and code logic.

Smarty gives developers:

- clean separation of presentation from application code
- PHP backend, Smarty template frontend
- compliment PHP, not replace it
- fast development/deployment for programmers and designers
- quick and easy to maintain
- syntax easy to understand, no PHP knowledge required
- flexibility for custom development
- security: insulation from PHP
- free, open source

The file structure and set up is listed below:

```
/
---www
    ---images
    ---home
---smarty
    ---templates
    ---templates_c
    ---configs
    ---cache
---classes
```

PHP / Logic Files

```
/www/
/www/home/
/classes/
```

Smarty Template / HTML Files

```
/smarty/templates/
```

The project is set up to work on any server running Apache, MySQL, and PHP and is being built and tested on Windows 7, Windows 8, and Linux.

The SQL queries are made using PDO (PHP Data Object) for both security and simplicity on the server side code. The PDO objects are used as followed:

```
<?php
```

```
    //*****FILE DEFINED OUTSIDE OF ROOT IN /classes *****
```

```
    $mysql = new PDO("mysql:host=localhost;dbname=database", 'root', "");
```

```
    $sth = $mysql->prepare("SELECT * FROM `users` WHERE `username` = :username");
```

```
    $sth->execute(array(":username" => $_POST['username']));
```

```
    if($sth->rowCount() != 1) {
```

```
        header("Location:loginpage.php?error=User_Not_Registered&username=".$_POST['username']);
```

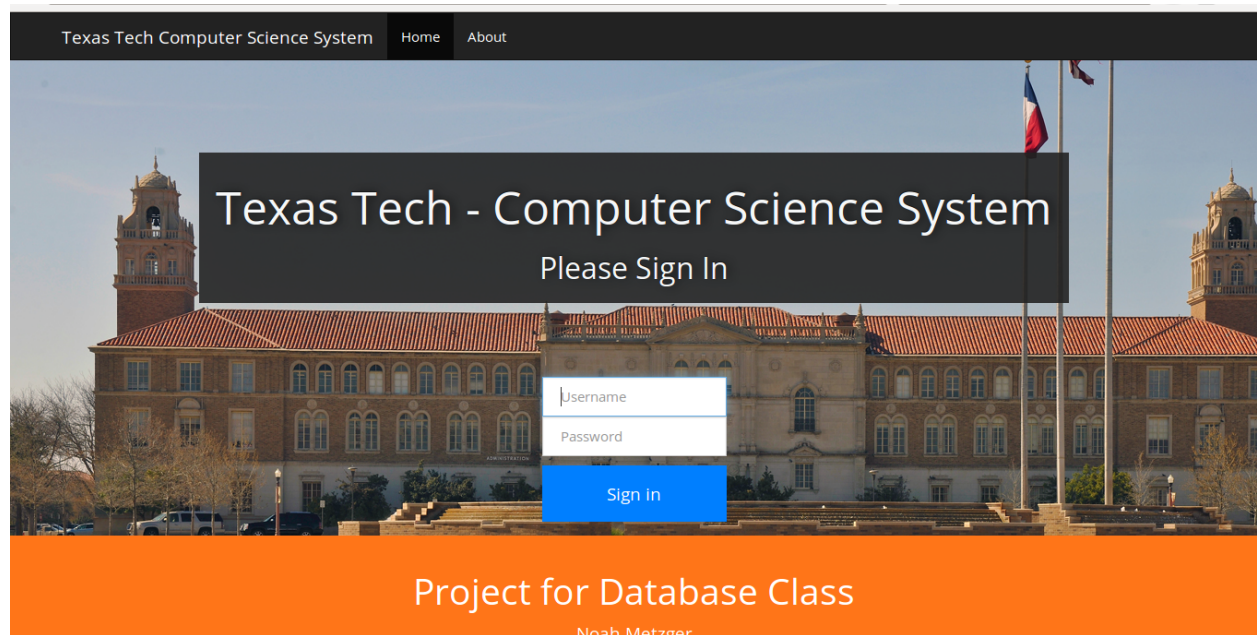
```
        exit();
```

```
    }
```

```
?>
```

The code was open sourced and hosted on GitHub at <https://github.com/dfranc3373/CSProject> to manage our code across all our machines consistantly.

Part E: Design



The design of the web application is consistent across the entire application and uses simplicity and pictures to attract the users to the content posted. The design uses Twitter Bootstrap to allow for a faster and more consistent look in the main pages. Bootstrap also allows for amazing colors and buttons with very little error.

The design focuses on the background images which are fixed to create a design that is both modern and clean. It also allows us to focus the users attention to the colors in each photo.

Texas Tech - Computer Science System

Welcome 'test'

New Instructor

New Class

Project for Database Class

Noah Metzger
Brian Fairbanks
Derrick Franco

Part F: Contributions

Noah Metzger:

Designed and Created original UML and final UML design, responsible for set up of Database design. Wrote blog post for project update. Also added parts A and B to report.

Brian Fairbanks:

Helped finalize UML and set up team Blog. Wrote first blog post

Derrick Franco

Created server setup, git account, coded preliminary PHP and Smarty files for project. Added parts C, D, E, F, and installation to report.

INSTALLATION

To install the program follow the following steps:

1. Checkout the latest repository from the Git account
// git checkout [git@github.com:dfranc3373/CSProject](https://github.com/dfranc3373/CSProject).git
2. Upload the database.sql file to PHPmyadmin
3. Download Smarty from smarty.net onto the computer
4. Go to /smarty folder in the project and edit the Smarty.php to match the location of smarty on your computer
5. Go to the localhost address