# Organization

CS/IT 490 WD, Fall 2013

Last update 2013-09-24

Written by Rachel J. Morris

# Breakdown

- Organizing your server's directories
- Layers
- Frameworks

# Organizing your server's directories

- Your web server shouldn't just contain every file in one directory.
  - Have a *content* folder, with *graphics*, *audio*, *scripts*, *styles*, etc.
  - Have a *layout* folder for PHP layouts that will be repeated on various pages
  - Have a folder for the “behind-the-scenes” workings for a given page
  - Have structural files in another folder
  - Just keep pages themselves in the root directory.

# Organizing your server's directories

- Organization will be something to keep in mind as we talk about **frameworks** and setting up the framework for your website.

# Layers

- Usually, it is good to encapsulate different types of functionality within different layers.
- This is as opposed to directly accessing certain types of functionality from anywhere in your program.
- For example, anywhere in the website's back-end, building a query string and sending it to the DB – Bad!

# Layers

- Think of creating a layer as making a class, or set of classes, to be the gatekeeper to that functionality.
  - Databases
  - APIs
  - Sockets

# Layers

- The “email-signup.php” page should NOT be creating:

```
INSERT INTO email_list (name, email)
VALUES (“rjm”, “rjmfff@umkc.edu”)
```

- Instead, there should be a **layer** that knows how to properly sanitize, check for errors, send queries, receive the results, and format them properly for consumption elseware in the “program”.
  - e.x. Storing the results in a class object, sending that object elseware.
- `$success = $DBLayer->SignUpUser( “rjm”, “rjmfff@umkc.edu” );`

# Layers

- Layers are everywhere when it comes to computers.
  - The system: Hardware, firmware, assembler, kernel, OS, applications
  - Network: Physical, data link, network, transport, session, presentation, application

http://en.wikipedia.org/wiki/Abstraction_layer
http://www.washington.edu/lst/help/computing_fundamentals/networking/osi

# Layers

- Website:
  - Model, View, Controller
  - Model, View, View-Model
  - Keeping the styles (CSS) separate from the markup (HTML) separate from the scripts (JS) separate from the logic (PHP)
  - Keeping certain logic separate from others (form validation, database access, etc.)

# The Database Layer

- Since we're not working with MySQL *yet*, you might be wondering what to continue working on for your group's project.
- Layers!
- There should be a class in place that is the gatekeeper to the database – everything that wants to give or receive data will need to go through it!

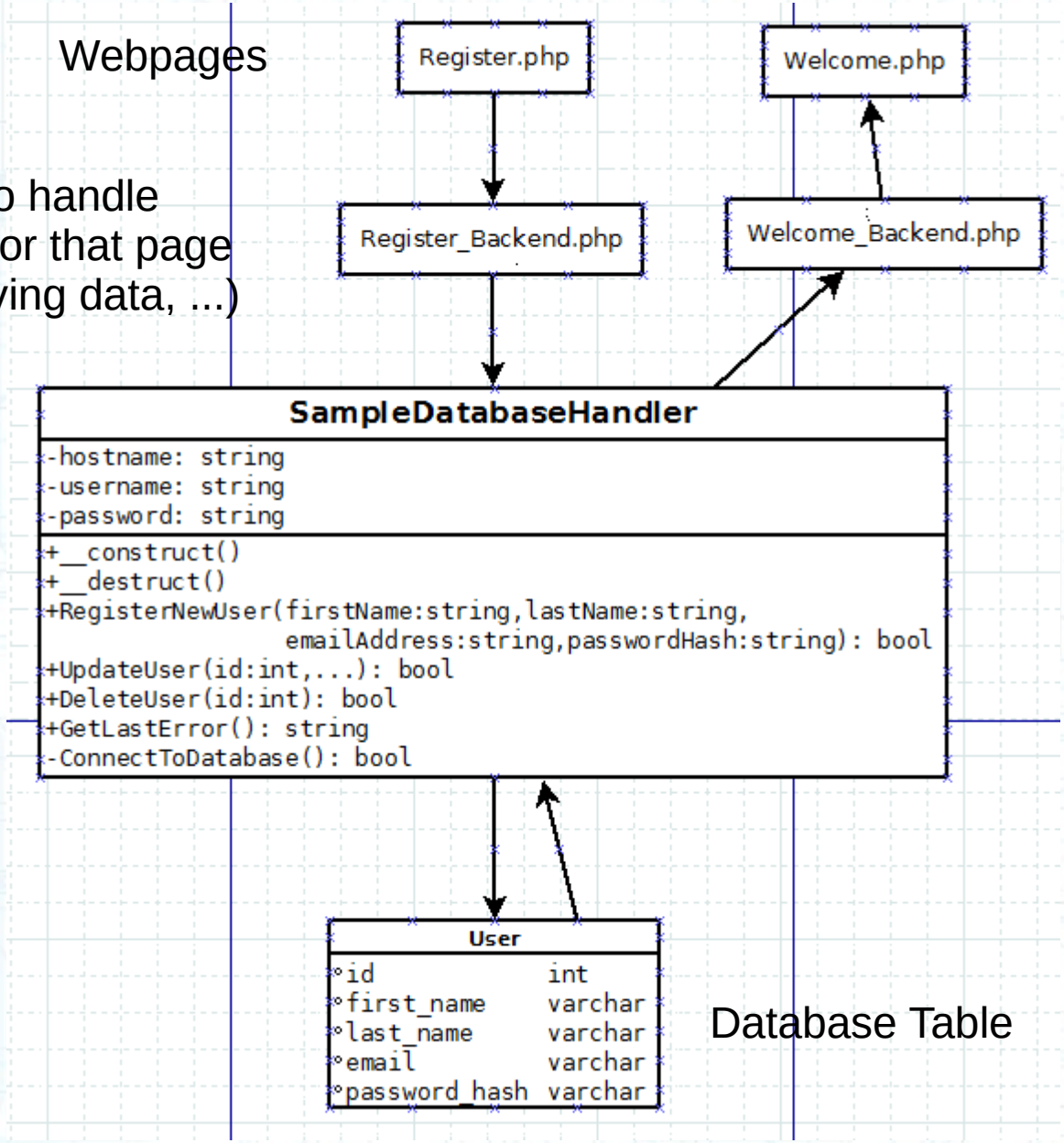
# The Database Layer

- Create a class that will have some of the functionality you may need.
- Since we're not connecting to the database *yet*, make any query functions return dummy-data.
- Then, you can swap out the dummy data for actual data later on!

Webpages

.php scripts to handle functionality for that page (forms, receiving data, ...)

.php file with Database-Handler class

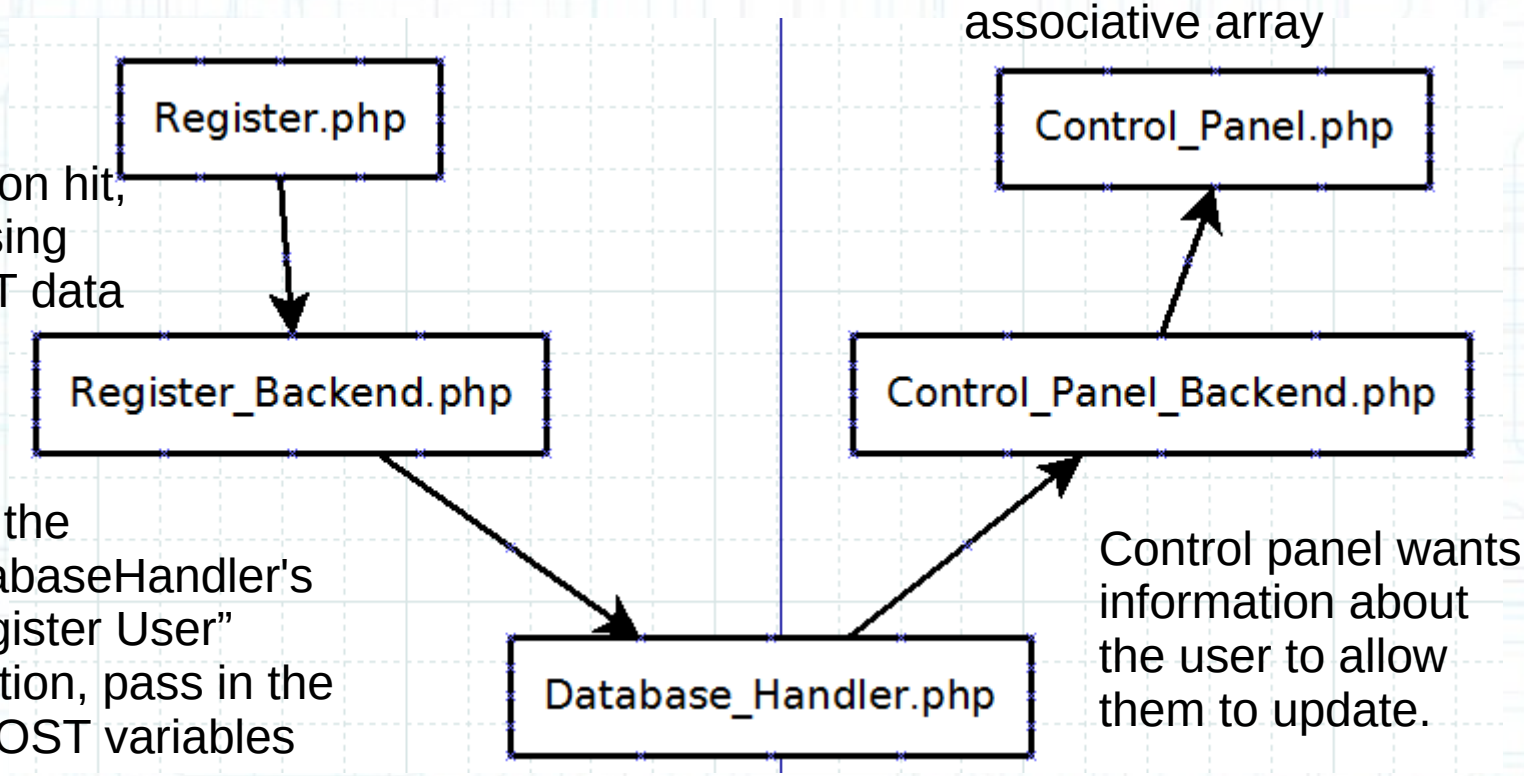



Database Table

# Layers

- Generally have two different types of .php files (both are just .php):
  - Webpage – The markup, with a any variables inserted into the page, passed in from the back-end.
    NO LOGIC, just variables!
  - Script – PHP script ONLY, no webpage content. Classes, back-end functionality, ...

# Layers




Submit button hit, handle parsing the $_POST data

Call the DatabaseHandler's "Register User" function, pass in the $_POST variables

Actual webpage plugs in the variables from the associative array

Control panel wants information about the user to allow them to update.

Call "GetUserInfo" function

Return data is associative array

# Gallery.php

# Database Layer

- When writing the database layer, you may also create **model** objects corresponding to some tables...
  - Author
  - Photo
  - Product
  - Etc.

# Database Layer

- Also keep in mind that you can **join** tables together in a query (we will discuss this more later).
- Basically, keep in mind that a query could return data from multiple tables, so you might need to return multiple objects.
  - Each photo will also have author information.

# Layers

- Understanding Model-View-Controller
- http://www.codinghorror.com/blog/2008/05/understanding-model-view-controller.html

# Frameworks

- You can build your own frameworks, and there are PHP frameworks you can install.
  - CakePHP
  - Symfony
  - CodeIgniter
- They help you create a clean, organized, MVC (model-view-controller)-based website.

# Frameworks

- Build a PHP MVC Framework in One Hour
- http://johnsquibb.com/tutorials/mvc-framework-in-1-hour-part-one

## Learning Curve (from the website)

- We are going to assume the following:
- You know PHP, preferably version 5 looking forward to 6.
- You at least understand the tenets of object oriented design.
- You know how to set up PHP include paths on your server, have the ability to change permissions settings, configure apache, etc.
- We are developers in a LAMP world, and will be writing from that perspective.

# Frameworks

- Build a PHP MVC Framework in One Hour
- http://johnsquibb.com/tutorials/mvc-framework-in-1-hour-part-one
- It is not required that you set up an MVC framework
- It is heavily suggested that you at least skim through the article and learn a little bit about what goes into creating a framework.

# Frameworks

- If you feel like doing the configuring yourself, you can use CakePHP, CodeIgniter, or Symfony.
  - Though probably not on the host that I'm providing. I haven't had much luck with 1&1.
  - There are communities based around these solutions you can get help from.

# Frameworks

- There are other frameworks out there for other languages...
  - ASP.NET MVC
  - Django (uses Python)
    - Django was developed in Lawrence, KS
    - Cerner hires Django developers
  - Ruby on Rails



Written by Rachel J. Morris

# References

The OSI Model, University of Washington
http://www.washington.edu/lst/help/computing_fundamentals/networking/osi

Abstraction layer, Wikipedia
http://en.wikipedia.org/wiki/Abstraction_layer