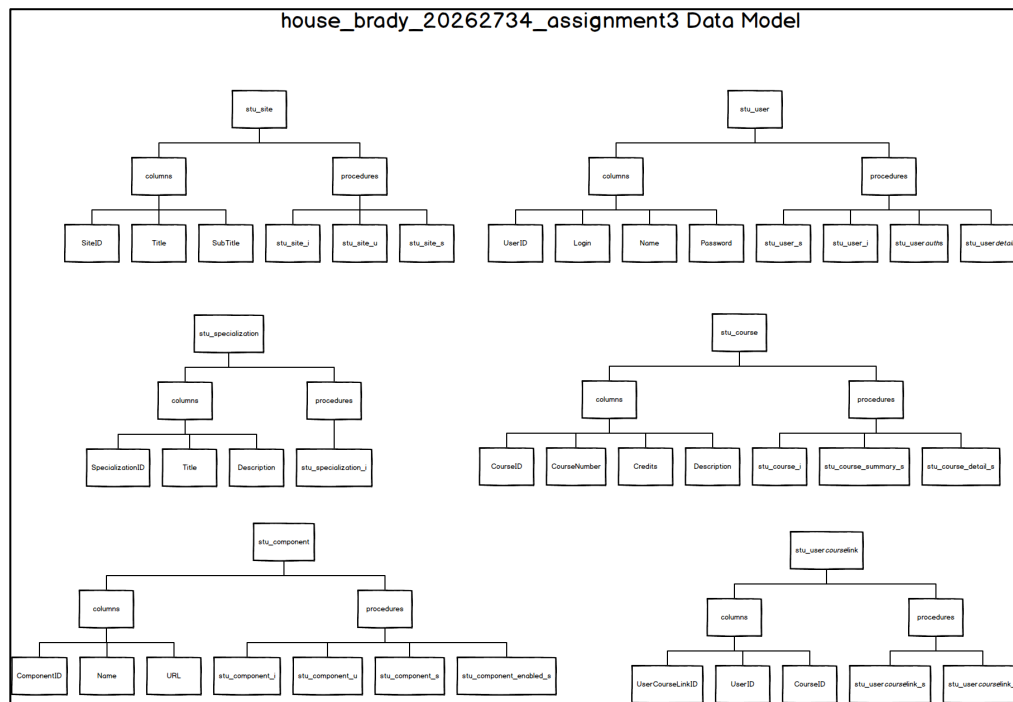


**Installation:**

1. Copy the entire assignment directory to the htdocs directory of your apache server
2. Start-up MySQL
3. Change the read/write settings on the content sub-directory. This can be done using chmod from the root assignment directory–

```
chmod -R u+rw,g+rw,o+r *
```

4. Open the root index.php using chrome. Since the application database does not exist, this will redirect you to the setup component –<http://index.php?option=setup&view=install>.
5. This view will invoke the /content/sql/install.sql script using the localhost/mysql database. Once it completes, your MySQL instance should now be configured with a "house\_brady\_20262734\_assignment3" database. After running the script, the setup component will display the message "house\_brady\_20262734\_assignment3 database created successfully."
6. You should now be able to re-open the root index.php, which should by default redirect you to the course list view, "index.php?option=store&view=courselist."

**Data Model:****Install Script**

The master install script is located in the content/sql sub-directory–

```
../content/sql/install.sql
```

For best results use the setup install utility--  
index.php?option=setup&view=install.

### Test Accounts

For testing purposes two accounts are packaged with the application. These include an admin and non-admin user.

#### Admin User Account

login: bhouse  
password: 33November

After login, this account can access the admin list view--  
index.php?option=admin&view=list.

#### Non-Admin User Account

login: bbaggins  
password: January44

After login, this account cannot access the Admin list view. If they try accessing the admin page, the application will display a "Not Authorized" error. This message is visible in the application's main content region.

### How To Add a User

Using the MySql command prompt (or phpMyAdmin interface), execute (call) the "stu\_user\_i" stored procedure compiled during install.

```
SET @p0='John Doe'; SET @p1='professor'; SET @p2='password'; SET  
@p3='johndoe@nomail.com'; SET @p4='word'; SET @p5='0'; CALL  
`stu_user_i`(@p0, @p1, @p2, @p3, @p4, @p5);
```

To give the user admin privileges, set the isAdmin bit to "1".

**NOTE - The current admin component does not provide any built-in support for adding users. It's assumed that the owner of the site can easily run the stu\_user\_i or stu\_user\_u stored procedures. It should also be noted there was no requirement for integrated user CRUD operations.**

### How to Update a User

Using the phpMyAdmin (or MySQL command prompt), execute (call) the stu\_user\_u stored procedure.

```
SET @p0='bhouse'; SET @p1='brady house'; SET @p2='33wonder'; SET
@p3='brady@houseknecht.com'; SET @p4='word'; SET @p5='0'; SET @p6='0';
CALL `stu_user_u`(@p0, @p1, @p2, @p3, @p4, @p5, @p6);
```

**Note - You cannot change the user's login since this value is used lookup the stu\_user record. Again, this is beyond the scope of the provided requirements.**

### Course Component

- GET Request Format -- index.php?option=course&view=list
- No authentication is required to access this component
- After the setup process is complete, this is default view for the application
- Opening the index.php will redirect the browser to the list view provided by this component
- Per the requirements, it provides a list and detail view

### Course List View



- GET Request format - index.php?option=course&view=list
- The page displays the contents of the stu\_course table (as per the requirements)
- Clicking the detail link for a given course will open the Detail View

Course Detail View

- GET Request Format – `index.php?option=course&view=detail&courseId=#`
  - **Note – The courseId, is appended to the url string. This approach was simpler than creating forms necessary to use a POST request**
- If the user is authenticated and not enrolled in the class, then the page will display an `**enroll**` link
- If the user is not authenticated then the `*enroll*` link is not displayed
  - The column is simply not rendered (See Screenshot above)

Course Enroll View

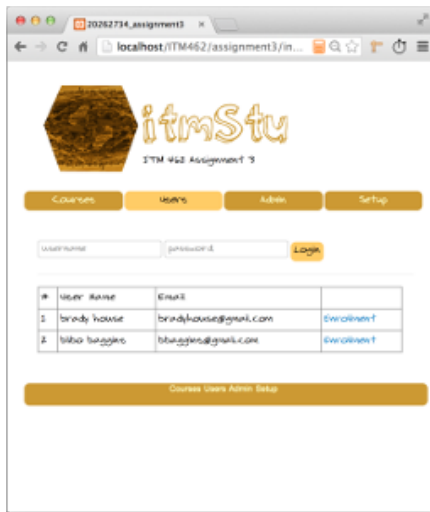
- GET Request Format – `index.php?option=course&view=enroll&courseid=#`
- Clicking the enroll link from the detail view invokes the `stu_user_course_link_i` stored procedure
- This procedure adds a record to the `stu_user_course_link` table

- **NOTE - once enrolled, a user cannot "un-enroll". Again, this represents a hole in the application. However, it does not violate the requirements for the assignment.**

### User Component

- GET Request Format – `index.php?option=user&view=list`
- No authentication is required to access this component
- Per the requirements, it provides a list and enrollment view

### User List View



- GET Request format – `index.page?option=user&view=list`
- Per the requirements, the page lists all users defined in the `stu_user` table
- The `stu_user_summary_s` stored procedure is services this view
- The enrolled link can be used to open user detail view

User Enrollment View

- GET Request format –  
index.php?option=user&view=enrollment&userid=#
- Page displays a list of courses that the user is enrolled in
- The stu\_user\_detail\_s stored procedure services (or is bound to) this view
- This procedure joins the stu\_user, stu\_course using the stu\_user\_course\_link table
- If the user is not enrolled in any classes, then the page simply displays the table header

**Admin Component**

- GET Request format index.php?option=admin&view=list
- Access to the component requires authentication and that isAdmin bit has been set to true for the user
- If the \*\*isAdmin\*\* bit on the returned record is true, then user is allowed to access the admin component
- Otherwise, they cannot access the component and will receive a Not Authorized (401) error message
- The Admin page broken up into two sections
  - Site Update
  - Component List

Admin Site Update Form

## Site Update

| Title                                 | Subtitle  |
|---------------------------------------|---|
| <input type="text" value="Index"/>    | <input type="text" value="ITM 462 Assignment 3"/> |
| <input type="button" value="update"/> |   |

- POST Request format –
  - URL – index.php?option=admin&view=siteupdate
  - \$\_POST['title']
  - \$\_POST['subtitle']
- This form allows the update of the site title and sub title (as per the requirements)
- The stu\_site\_u stored procedure services (or is bound to) this view
  - This procedure updates the stu\_site table
- The form can be disabled (hidden) via the component list (refer to the next section)
- **Note – all components are effected minus the setup component. The setup component uses the SETUP\_\* values defined in the configuration.php script. This represents an obvious hole. Again, however, patching this one up is beyond the time constraints for this assignment. In other words, additional time (or iterations) would be required.**

Admin List View

| Component List |        |            |         |                         |
|----------------|--------|------------|---------|-------------------------|
| #              | Option | View       | Enabled |                         |
| 6              | admin  | list       | Y       | <a href="#">Disable</a> |
| 7              | admin  | siteupdate | Y       | <a href="#">Disable</a> |
| 8              | setup  | report     | Y       | <a href="#">Disable</a> |
| 9              | setup  | list       | Y       | <a href="#">Disable</a> |
| 10             | course | list       | Y       | <a href="#">Disable</a> |
| 11             | user   | list       | Y       | <a href="#">Disable</a> |

- GET Request format – index.php?option=admin&view=list
- Components are listed by their option / view combination
- The user can turn components on or off by clicking the enable/disable link (as per the requirements)
- Component state is based on the stu\_component table's "isEnabled" flag
  - Clicking "enable" (or "disable") invokes the stu\_component\_u procedure which updates the stu\_component table
- **Note – its possible to turn off everything except for the course list. However, since this is the default view, the site will still open this view. In this state all buttons / links visible in the header and footer of the application are hidden.**