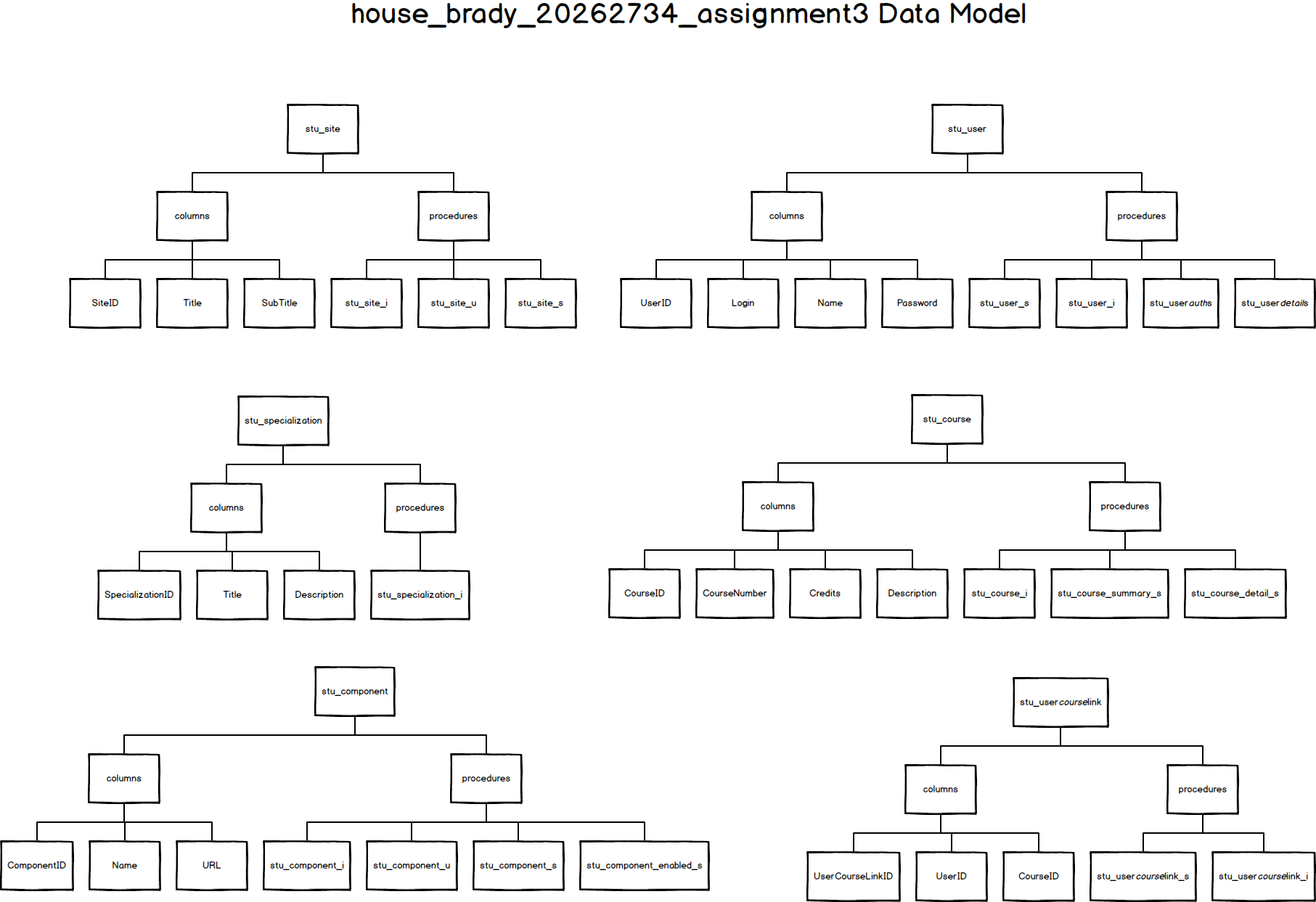
**Installation:**

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 \*

1. 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.
2. 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."
3. 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.

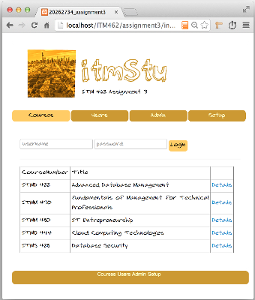
|  |
| --- |
| SET @p0='bhouse'; SET @p1='brady house'; SET @p2='33wonder'; SET @p3='brady@houseknecht.com'; SET @p4='word'; SET @p5='0'; SET @p6='0'; CALL `stu\_user\_u`(@p0, @p1, @p2, @p3, @p4, @p5, @p6); |

**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**

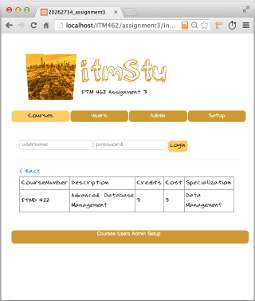
* GET Request Format -- index.php?option=course&view=list
* No authentication is required to access this component
* After the setup process is complete, this is default view for the application
* Opening the index.php will redirect the browser to the list view provided by this component
* Per the requirements,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

**Course List View**



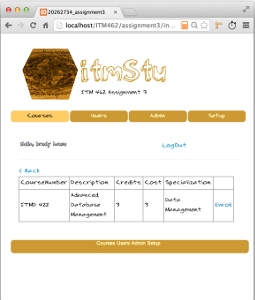
* GET Request format - index.php?option=course&view=list
* The page displays the contents of the stu\_course table (as per the requirements)
* Clicking the detail link for a given course will open the Detail View

**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 Detail 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**