**SHSU E-Center Design**

**Contents:**

**Design Details**

**1)Matrices**

* + Functional Design
  + Jackson Design
  + Dynamic Data Flow Design
  + Object Oriented Design

**2)Three-tier**

**3)Race Condition**

**4)Deadlocks**

**5)Network Security**

**6)Other Technology**

**7)Optimization**

**8)Reusability**

**9)Implementation Plan**

**10)Testing**

Black Box Testing

* + Welcome Page
  + New User page
  + Books
  + Tutors
  + Carpool
  + Roommates

**11) E-R Diagram**

**12)Design Diagrams:**

* + Functional Design
  + Jackson Design
  + Dynamic Data Flow Design
  + Object Oriented Design

**13)Prototype Images**

**Design Details**

**Metrics (1):**

Functional Design:

This design is the traditional technique which explains the basic procedures for the completion of the tasks but does not specify anything about optimization of time and space. It also doesn't explains the ways to increase the cohesion and decrease the coupling among the modules. This technique might be useful to explain the basics of the design but is not efficient and effective for the proper implementation of the design.

Jackson Design:

Static Data Structure Design, otherwise known as Jackson Structured Design (JSD), is a method for designing the structure of a program based on the Input/Output structure and operations needed to be carried out by the program. Best use case for Jacksons is when laying out control structure for a program. An advantage of Jacksons over the other design techniques is Jacksons breaks everything down to the simplest form, laying out all operations and iterations needed for the program. The same could also be a disadvantage as this design technique places a lot of emphasis on detailed control structures but leaves other aspects of software or a program left out.

Since we wanted our implementers to cover every single little aspects of our website we did not choose static data structure design.

Dynamic Data Flow Design:

Dynamic Data flow is easy to understand, helps in defining the boundaries of the system and explains the logic behind the data flow within the system but undergoes lot of alteration before going to users, so makes the process little slow. It makes the programmers little confusing towards the system.

.

Object Oriented Design:

Object oriented design breaks down the objects in to the simplest form laying out all operations and iterations needed for our design. We can use multiple features of object oriented designs like inheritance, encapsulation and information hiding to make our program to optimize more time and space and create a great reusability for the future users or programmers. So, due to these benefit of using object-oriented design we picked it for our implementation phase.

**Three-tier (2):**

SHSU E-center has three-tiers that are the user interface, a separate processing element and, a separate data storage element. The user interface consists of a login page. The users have to insert their username and password. The password is protected by MD-5. The username and password is compared with the main records. If the username and password is matched with the main record then it will navigate to the home page. If the username and password doesn't match with the main records then it will prompt an error message. On home page users can see four different selection i.e. Books and notes, Tutor and study groups, Carpool, Rooms and roommates. User can select any of the four options to view the features. After clicking any of the four options it navigates to the specific page. On each page users can see postings done by other users and they can post their own posts. Every posting done by the users are saved on the database. Any updates done or deleted posts are updated on the database. There is separate database for each option on home page. The user login information is stored on a separate database.

**Race Conditions (4):**

In order to prevent race conditions occurring, typically we would put a lock around the shared data to ensure that only one thread can access the data at a time.

Race condition occurs when two or more threads are able to access shared data and they try to change it at the same time. Because the thread scheduling algorithm can swap between threads at any point, we don't know the order at which the threads will attempt to access the shared data. Therefore, the result of the change in data is dependent on the thread scheduling algorithm, i.e. both threads are 'racing' to access and change the data.

But as for our design the database management system will handle the ‘ACID’ properties of it and we do not have any race condition occurring for our application. For example: if someone makes a post in our website, another user cannot edit the post at the same time because these posts are read-only for other users, which prevents race condition.

Also, two different users searching for a same query will not affect the application because it will be processed simultaneously because of the multi-threading processing we having in our design.

**Deadlocks (5):**

Our application will not have a deadlock problem because our database is ‘read-only’. Users can only read the data from the server’s database. The only time the users can ‘write’ on the database is when they want to edit their post. There would not be any problem even while the user ‘writes’, because the user can only edit his/her post or information. It means only one user will be editing one post. This will automatically avoid deadlock because deadlock only occurs when more than one application locks the data at the same time.

**Network Security (7):**

1. Network Communications Security

For our security, we plan to use Apache, a common front-end, which utilizes its security and encryption modules (ie, mod\_ssl). Using Apache we would be able to provide both HTTP and HTTPS since it has the ability to handle any conflict of the two on the same server. Apache can be used to create both public and private keys for both one-way and two-way authentication. To provide the user with secure communication with the server, we plan to use two/way authentication using public/private key pairs. For public/private key exchange we plan to utilize the Diffie-Hellman key exchange algorithm with a RSA key.

Apache also has the ability to provide the system with a Certificate Signing Request or CSR that is used to purchase a SSL certificate from a certificate authority such as Verisign, Globalsign, or Webtrust. This CSR will be generated by Apache and stored as a .csr file that can be submitted to one of the CA’s listed or others.

1. User Account Passwords

User account passwords will not be stored within the server database; instead, the hash of the password will be stored in the system. Once the user inputs a password, the system will create a hash using MD5 hash function algorithm and store the hash in the database. Each time the user logs in, the hash of the password will be created and compared to the stored hash for authentication.

In the event that a user forgets or loses his/her password, the system cannot use the stored hash to provide the user with the password since the hash function is one-way. Rather, the system will generate another password and hash, and send the temporary password to the user via email. The user will be allowed to use the temporary password, but advised to create a new password as soon as possible.

**Other Technologies (8):**

There are many other technologies such as Remote procedure call (RPC), servlets and multi threaded sockets that are appropriate for other type applications but our application is web based and those are not appropriate for our use. Mainly we need a web server and a database application. We chose to use apache for server and SQLite for our database because of its simplicity while we could servlets for our server we chose php instead. Servlets don't provide a notable advantages over php for our application because php supports objects and is more geared towards html web pages than java. We chose not to use adobe flash because we want users to be able to access the website from more platforms than flash is available on. In addition to html we are utilizing CSS( cascading style sheets) and JavaScript for the client side presentation and scripting, respectively.

**Optimization (10):**

Optimization for time should not be done during the design phase. It should be done after the design is fully completed. If we optimize during we work on design it is local optimization. Local optimization may slow the overall system down so we waited until the design was completed to optimize for time. Waiting until the design has been completed allows for global optimization. We made sure that our design had high cohesion as possible and low coupling. We will also examine the execution time for each module and give different execution for each module. We went through each module and looked for the possible improvement considering the cost for it. We also established the priorities for the optimization process starting with the highest priority module until we met our specification goal.

To optimize for space we grouped the logically related modules together and also we designed it with the best way to have high cohesion and low coupling. We didn't need to choose the specific search algorithm because our database query search would do that for us which minimized that problem for space.

**Reusability (11):**

We have written the procedures that are object oriented that is , each procedures perform one task which makes it easier to reuse. We have added comments on the procedures and modules which help other developers to easily figure out what our code and design will do. We have grouped similar procedures in a module which really helps to organize the code functionally. We have also developed the library of procedures that could be reused using code library tool. Also, we have created custom objects by encapsulating related code in class modules so that we could add properties, methods, and events to custom objects and organize them in custom hierarchical object model. This custom object could be treated as black box so that we or other programmers can use the object without thinking about the code that it contains. Finally we have also developed the interfaces to extend custom objects.

**Implementation Plan (12):**

First thing we need to complete is writing the code and installing the program in our server computer. Then we have to create our database. Although most of the database can be pulled from the SHSU website, we might have to manually input some of the data. Then we will get a domain name from the hosting service. Patching our servers is the next thing we will focus to keep the server computer up to date. We will install and configure the web server and php. We will also install Sqlite and database server. We will run the tests mentioned in this design to make sure the system works as expected. We will modify the code if the test does not run as intended. We will also do some advertising of our website online as well as in the college billboards.

**Testing (9):**

**Black Box Testing:**

Black Box testing will be conducted to ensure the website runs the way it is intended to run. This test will focus on testing the functionality of the website, independent of the internal codes. In this testing, the correct and incorrect data are entered to ensure that system distinguishes between them.

**1. Welcome Page**

Welcome page has to main activity. First is to log in the valid user. Second one is to let a user create a new account. When a user clicks on New User tab, the system should allow a user to create a new user. If the system does not navigate to the "New User" page, there is a flaw in the system, and code should be revised.

When a user enters valid username and password, the user has to be authenticated by the server and be able to navigate different pages. For the testing of the Login Page, we will first enter valid credentials and then enter invalid credentials.

Entering valid credentials:

In this test, the correct username and the password associated with that username will be entered. If the server accepts this username and password and allow user to further navigate to other pages of this website, the login page works fine.

Entering invalid credentials:

In this test, first the correct username with a wrong password will be entered. If the server authenticates the user, there is a flaw in the software and the code has to be revised. Then, the correct username without a password will be entered. Similarly to the above situation, when the valid username will be entered with password field blank, server should not authenticate the user. Likewise, system should not authenticate the user when a blank or invalid username is entered with any correct password.

**2. New User Page**

When someone tries to create a new user, all the fields like username, password, retype password, first name, last name, SHSU email address, postal/zip code should be entered. The user also has to select the sex. All of these fields should be entered correctly for a user to create an account. If any of this field is not entered or selected, the server should not allow the user to create a new account. Also, if the password and retype password field do not have a same string, system should notify the user to retype the password. The server also has to check the database of SHSU to make sure the SHSU email address is valid. Besides that, the server also should not let a user to create a username if the username has already been chosen by another user.

If the system allows the user to create an account only after fulfilling all the conditions above, the system is working correctly. If the system allows someone to create an account without fulfilling above mentioned requirements, the codes should be revised.

**3. Tutors**

For this testing, we will first select the department we need to search from the drop down box which will be different subjects like Mathematics, Kinesiology, Biology, etcetera. Then we will select the course name like CALC 1401, BIOL 4301, of the respective department. After department is selected, the system should load the subjects pertaining to that department. For example when one selects the depart Computer Science; the system should load the courses like COSC 2325, COSC 2329, etcetera. After the correct department and courses are selected, the search tab will be clicked. After one clicks the search tab, the server should query the database on those two things: (i) Requesting/ Offering tutorial, and (ii) Course. Then the system should provide the posts related to the query mentioned by the user.

After verifying the above parts, the system also should make sure that courses listed should belong to the department selected. Besides that, after the user clicks the search tab, the messages shown in the page should relate to the fields the user have selected.

**4. Books**

First the use will select if he/she wants to buy or sell a book. For searching part, there are two searching criteria in this page. Search by course and search by book. In this page, first the department needed is selected from the drop down box. After that, the system should load the courses for that department. It means, after the department Computer Science is selected in the department field, there should be the courses of computer sciences in the dropdown list of the “Course”.

Another search criterion which is by books offers options of searching by Title and/or Author and/or ISBN. When the search field is entered, the query should look for the options selected in the entire pool of posts and display the posts that have those keywords.

**5. Carpool**

After selecting the option for offering/requesting carpool services, the user will select the location and the day(s) for the services and enter the search tab. The system should query the database and show the posts of the mentioned selections.

For test, first the possible selection of data will be entered to check if the system works in the normal situation. Then, the city will be misspelled to check how the system reacts to it. The test will also be conducted to see if leaving all fields empty will prompt the server to search anything. Then, the test will be conducted by un checking all the days. System is not supposed to display any result when no day is selected.

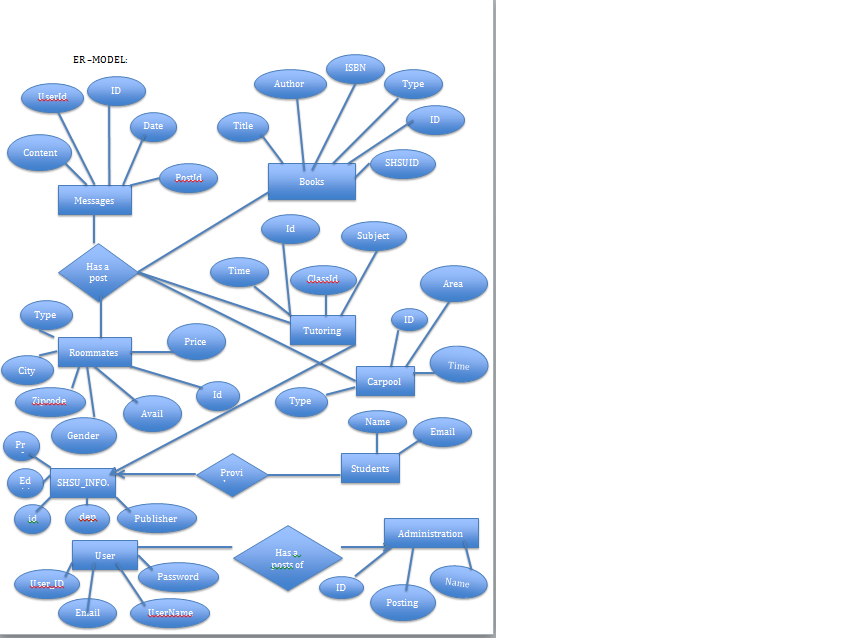
**6. Roommates**

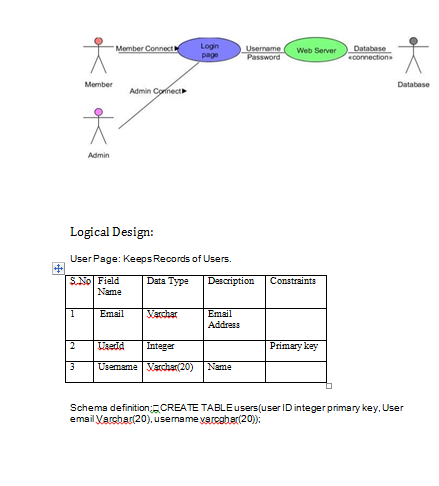
After selecting the option of requesting a roommate or offering a roommate and entering a location, the system should query the database and display the posts based on the user’s selection.

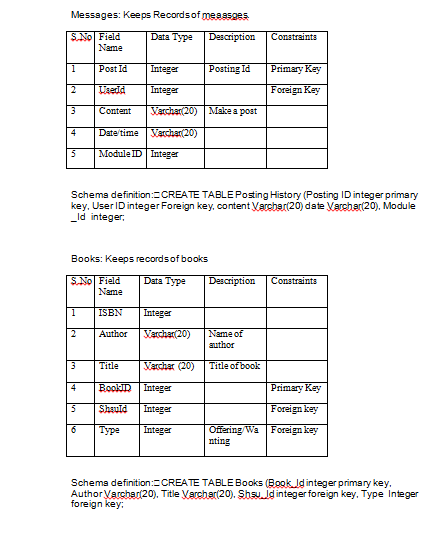
First we will test by leaving all fields empty and enter the submit button. In this case, nothing should display. Entering the misspelled words also should display the error message. Simply leaving the field of requesting or offering a roommate should show all the posts that are associated to the certain location. Specifically:

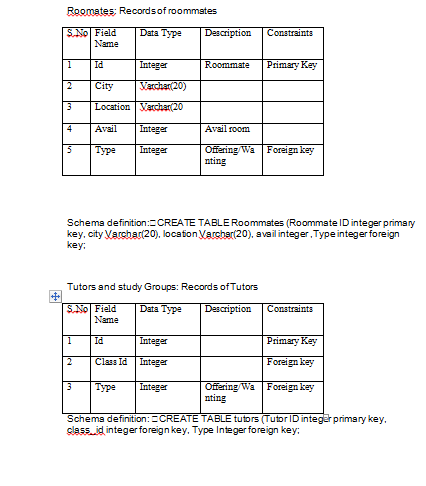
* Only local zip codes can be entered for the location
* There is only one selection for Location Type.
* The message fits requirements for posting (character limits, etc.)
* The new message shows up when the Post button is selected.
* The new message does not show up when Cancel is selected.
* All of the options are visible including the ability to reply to the message.
* Selecting the thread displays the message properly.

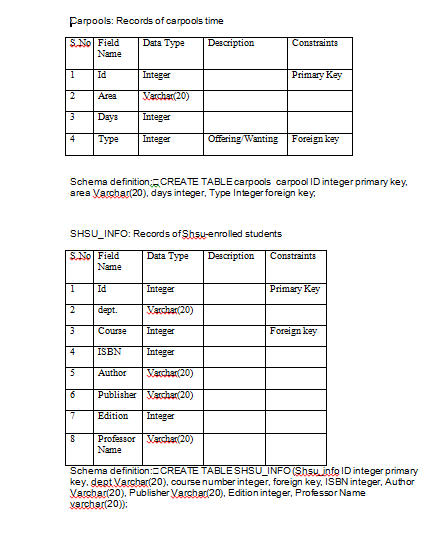
**E-R Diagram (6):**





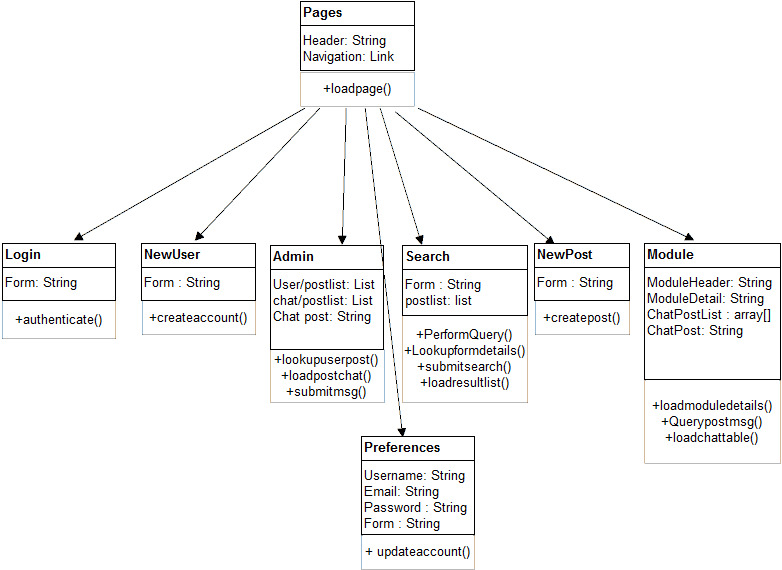


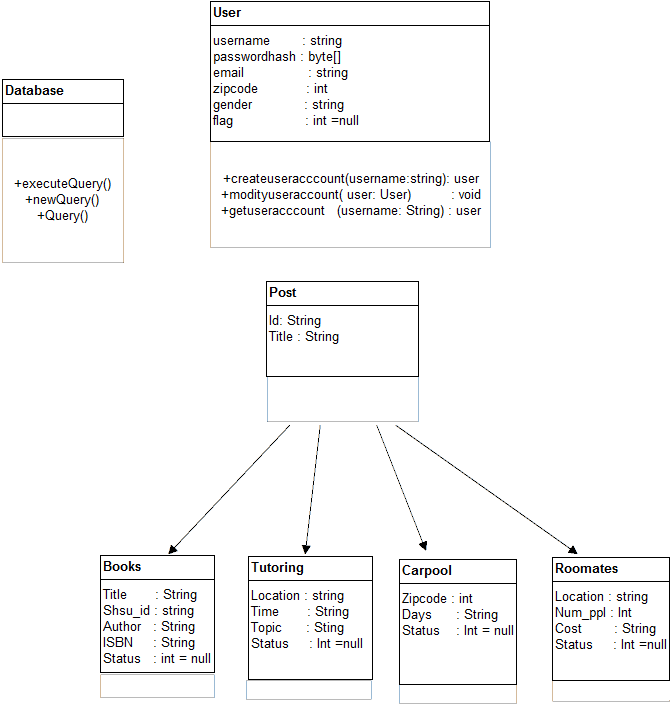




**Design Diagrams:**

**Object Oriented Design->**





**References**

**Burris, David. "Course Documents." Sam Houston   
 State University, n.d. Web. 25 Mar. 2013.**

***wikipedia*. N.p.. Web. 25 Mar 2013. <http://www.wikipedia.org/>.**