Social Network Solution

By

Steve Fedak

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

Spring 2008
Social Network Solution

By

Steve Fedak

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

© Copyright 2008 Steve Fedak

The author grants to the Information Technology Program permission
to reproduce and distribute copies of this document in whole or in part.

_________________________________________________ ________________
Steve Fedak         Date

_________________________________________________ ________________
Anny Prabhakar, Faculty Advisor      Date

_________________________________________________ ________________
Hazem Said, Ph.D. Department Head      Date
Acknowledgements

I would like to give special thanks to Annu Prabhakar, Associate Professor and Academic Advisor, for providing stimulating ideas and excellent project management support, while also motivating me to work hard. I would also like to give special thanks to all beta testers. They have assisted with developing this project by providing feedback and great ideas.
## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>i</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>ii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>iv</td>
</tr>
<tr>
<td>Abstract</td>
<td>v</td>
</tr>
<tr>
<td>1. Statement of the Problem</td>
<td>1</td>
</tr>
<tr>
<td>2. Description of the Solution</td>
<td>1</td>
</tr>
<tr>
<td>2.1 User Profile</td>
<td>2</td>
</tr>
<tr>
<td>2.1.1 Developer</td>
<td>2</td>
</tr>
<tr>
<td>2.1.2 User</td>
<td>3</td>
</tr>
<tr>
<td>3. Design Protocols</td>
<td>3</td>
</tr>
<tr>
<td>4. Deliverables</td>
<td>3</td>
</tr>
<tr>
<td>5. Design and Development</td>
<td>4</td>
</tr>
<tr>
<td>4.1 Timeline</td>
<td>4</td>
</tr>
<tr>
<td>4.1.1 Senior Design I Accomplishments</td>
<td>5</td>
</tr>
<tr>
<td>4.1.2 Senior Design II Accomplishments</td>
<td>5</td>
</tr>
<tr>
<td>4.1.3 Senior Design III Accomplishments</td>
<td>5</td>
</tr>
<tr>
<td>4.2 Budget</td>
<td>6</td>
</tr>
<tr>
<td>4.3 Software</td>
<td>6</td>
</tr>
<tr>
<td>4.4 Hardware</td>
<td>6</td>
</tr>
<tr>
<td>6. Proof of Design</td>
<td>7</td>
</tr>
<tr>
<td>6.1 Social Network Solution Installation</td>
<td>7</td>
</tr>
<tr>
<td>6.1.1 Download / Uncompress</td>
<td>7</td>
</tr>
<tr>
<td>6.1.2 Installation Wizard</td>
<td>8</td>
</tr>
<tr>
<td>6.2 User Registration &amp; Login</td>
<td>10</td>
</tr>
<tr>
<td>6.3 Update User Profile</td>
<td>11</td>
</tr>
<tr>
<td>6.4 Add Friends</td>
<td>12</td>
</tr>
<tr>
<td>6.4.1 Accept Friend Requests</td>
<td>14</td>
</tr>
<tr>
<td>6.5 Messaging</td>
<td>16</td>
</tr>
<tr>
<td>6.5.1 Send Messages</td>
<td>16</td>
</tr>
<tr>
<td>6.5.2 Reply to Messages</td>
<td>17</td>
</tr>
<tr>
<td>6.5.3 Delete Conversations</td>
<td>19</td>
</tr>
<tr>
<td>6.6 Photo Gallery</td>
<td>19</td>
</tr>
<tr>
<td>6.6.1 Create Album</td>
<td>19</td>
</tr>
<tr>
<td>6.1.1 Upload Images</td>
<td>20</td>
</tr>
<tr>
<td>6.1.2 View Album/Images</td>
<td>21</td>
</tr>
<tr>
<td>7. Testing Procedures</td>
<td>22</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1:</td>
<td>Project Timeline</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2:</td>
<td>Project Budget</td>
<td>6</td>
</tr>
<tr>
<td>Figure 3:</td>
<td>Download Social Network Solution</td>
<td>7</td>
</tr>
<tr>
<td>Figure 4:</td>
<td>Installation Wizard - Welcome</td>
<td>8</td>
</tr>
<tr>
<td>Figure 5:</td>
<td>Installation Wizard – Database setup</td>
<td>9</td>
</tr>
<tr>
<td>Figure 6:</td>
<td>Register / Login</td>
<td>10</td>
</tr>
<tr>
<td>Figure 7:</td>
<td>Update Profile</td>
<td>11</td>
</tr>
<tr>
<td>Figure 8:</td>
<td>User List</td>
<td>13</td>
</tr>
<tr>
<td>Figure 9:</td>
<td>Add Friend – Profile View</td>
<td>14</td>
</tr>
<tr>
<td>Figure 10:</td>
<td>Friend Request</td>
<td>15</td>
</tr>
<tr>
<td>Figure 11:</td>
<td>Send Message</td>
<td>16</td>
</tr>
<tr>
<td>Figure 12:</td>
<td>Inbox</td>
<td>17</td>
</tr>
<tr>
<td>Figure 13:</td>
<td>Conversation View</td>
<td>18</td>
</tr>
<tr>
<td>Figure 14:</td>
<td>Create Album</td>
<td>19</td>
</tr>
<tr>
<td>Figure 15:</td>
<td>Upload Image</td>
<td>20</td>
</tr>
<tr>
<td>Figure 16:</td>
<td>Album View</td>
<td>21</td>
</tr>
<tr>
<td>Figure 17:</td>
<td>Image View – Album</td>
<td>22</td>
</tr>
<tr>
<td>Figure 18:</td>
<td>User Registration Template</td>
<td>24</td>
</tr>
<tr>
<td>Figure 19:</td>
<td>User Registration PHP</td>
<td>25</td>
</tr>
<tr>
<td>Figure 20:</td>
<td>User Registration Business Objects</td>
<td>26</td>
</tr>
<tr>
<td>Figure 21:</td>
<td>User Registration Data Objects</td>
<td>27</td>
</tr>
</tbody>
</table>
Abstract

The Social Network Solution is a free, open source social network application. It is designed to be easily installed with minimal server requirements and to provide a user friendly social network. The Social Network Solution was created using PHP, CSS, HTML and MySQL. The application is developed as a three tier application following object-oriented design principles, to maintain scalability for any size social network. The Social Network Solution was created to include many of the basic functions of typical social networks such as profiles, friends, public and private messaging, and profile specific image galleries. This project also includes a developer Web site that consists of documentation on how to install the application and create or modify modules. There is also a forum to provide a community for developers to exchange ideas about custom modules and templates they have developed. Overall, the Social Network Solution is an easy way to implement a social network, for anyone, even without Web development knowledge.
Social Network Solution

1. Statement of the Problem

Currently, there is only one free, open source, social network, ELGG, available for download and use. All other social network solutions can cost thousands of dollars to implement. This causes an issue for many people who wish to create their own social networking Web site. Since ELGG is the only free option available, users do not have a choice if they wish to create a social network. This can cause issues if users can not meet ELGG’s application requirements, or the user prefers a different type of application that is not so bulky and complicated. The Social Network Solution provides a competing solution for users looking for a straightforward, simple social network.

2. Description of the Solution

The Social Network Solution is a Web application that allows people to easily create their own social networking Web site. There are two major aspects to this project:

- Installing or creating a social network
- Using the social network that was created.

The Social Network Solution is available to download from the Social Network Solution Web site. This site will contain a developer’s area, which will consist of a forum for programmers to communicate with each other. The site will also provide installation guides, tips, and server requirements for clients to download free of cost. A person can download the application and upload the files to a Web server. Within minutes a person can create their own social network by following the installation wizard. The Social Network Solution allows for people to connect by creating profiles, having friends, and communicating with users by messaging and images.
There are three important aspects that the Social Network Solution will provide:

1. The application is free of charge to download.
2. The code will not be encrypted, open source for everyone to edit.
3. The system will be scalable, to handle large social networks.

Providing the code, free of charge, is a major advantage over competing social network products other than ELGG because, most other companies are charging large amounts for their application. Open source code will be a plus for developers because it will allow them to modify or update the features of the application to their liking. Some social networks have millions of users resulting in very high traffic. Social Network Solution is developed in Object Oriented Programming (OOP) which allows the program to handle a high number of users. The application runs off PHP programming to connect to a MySQL database and handle data. The default templates are constructed in HTML and styled with CSS.

2.1 User Profiles

There are two user groups that were considered while developing the Social Networking Solution.

2.1.1 Developer

The developer sets up and installs the system on a Web server. This person will have access to the whole Social Network Solution application and is able to modify the code. The developer will create the site name, description, and title. Also, the developer can create a site design if the administrator wishes to have a custom design. If not, the developer will be able to choose from two default templates or any custom templates that might be available from the Social Network Solution Web site.
2.1.2 Users

Users are general users that will be using the social network. Once registered these users will be able to create and update their profile, add friends, send message to their friends, upload images to their gallery, and view other user’s profiles.

3. Design Protocol

This application was developed utilizing my skills in Web development and database. These are my two key areas of study while attending the University of Cincinnati, College of Applied Science. The Web site was developed using HTML, CSS, and PHP using the free text editor, EditPlus. The database was created using phpMyAdmin to create and edit a MySQL database. To implement this project, there is a cost apart from the application itself. To install and use this application a user needs a Web host with certain requirements, such as PHP and MySQL.

4. Deliverables

The Social Network Solution includes the follow deliverables in the final product.

1. Fully functional, installable social network
   a. User profile
      i. Personal information
      ii. Profile picture
      iii. Send/Receive public and private messages
      iv. Friends list
      v. Image gallery
   b. Administrative interface

2. SQL database schema for MySQL
3. Additional Web site that provides solution for download and documentation
   a. Solution package for download
   b. Installation guide and server requirements
   c. Developer forum
   d. FAQ
   e. Installed examples

5. Design and Development

5.1 Timeline

Development of the application and will continue to expand it. Along with the development, I will be researching PHP methods and the best practices to use. Figure 1 shows the timeline of this project.

<table>
<thead>
<tr>
<th>ID</th>
<th>Task</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Research</td>
<td>2007</td>
</tr>
<tr>
<td>2</td>
<td>Write Proposal</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Proposal Presentation</td>
<td>2008</td>
</tr>
<tr>
<td>4</td>
<td>Database Development</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Application Development</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Testing</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Develop Design</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Design Freeze</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>User Testing</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Application Updates</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Final Report</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Final Presentation</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Project Timeline

Organizing and understanding the concept of Tracking Issues Connection Software took place during May and June of 2005. I attended several meetings with my technical advisory Professor Annu Prabhakar to help develop ideas and a structure for this project. Research was performed on many aspects of this project.
5.1.1 Senior Design I Accomplishments

- Research available social network options
- Identify user profiles
- Install Xampp and test local development
- Write a proposal for developing Social Network Solution
- Present my project proposal

5.1.2 Senior Design Project Management I Accomplishments

- Attended meetings with Advisor to manage project development
- Designed the default template
- Created the database schema
- Developed user registration and login
- Developed user profile layout
- Developed update user profile
- Developed user messaging system
- Developed Installation wizard
- Tested the project on a Web server
- Design Freeze presentation and report

5.1.3 Senior Design Project Management II Accomplishments

- Started user beta testing
- Modified user messaging system
- Developed image gallery
- Final report and presentation
5.2 Budget

The total retail cost of the project is $768.40. Figure 2 shows the cost for each item and includes a short description about each item. Since the project was developed locally using Xampp, there is no cost for a Web server. The cost of Adobe Photoshop was not included because I used the University of Cincinnati’s Computer Labs to edit and create images.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Retail Cost</th>
<th>Actual Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Server</td>
<td>Web site hosting</td>
<td>119.40/yr (est.)</td>
<td>$0.00</td>
</tr>
<tr>
<td>PHP</td>
<td>Server-side Web language</td>
<td>$0.00</td>
<td>$0.00</td>
</tr>
<tr>
<td>MySQL</td>
<td>Server-side database</td>
<td>$0.00</td>
<td>$0.00</td>
</tr>
<tr>
<td>phpMyAdmin</td>
<td>Graphical user interface for administering MySQL database.</td>
<td>$0.00</td>
<td>$0.00</td>
</tr>
<tr>
<td>Smarty Templates</td>
<td>Template engine for PHP</td>
<td>$0.00</td>
<td>$0.00</td>
</tr>
<tr>
<td>EditPlus</td>
<td>Text, HTML, and programmer’s editor.</td>
<td>$0.00</td>
<td>$0.00</td>
</tr>
<tr>
<td>Adobe Photoshop</td>
<td>Graphics Editor</td>
<td>$649.00</td>
<td>$0.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$768.40</strong></td>
<td><strong>$0.00</strong></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Project Budget (1, 4, 9)

5.3 Hardware

*Web Server:* An Internet service to provide a Web site to host the application.

5.4 Software

*PHP 5:* Scripting language that is suited for Web development.

*MySQL:* Database server.

*phpMyAdmin:* Graphical user interface used to manage the database.

*XAMP:** Easy to install Apache distribution containing MySQL, and PHP (10)

*Adobe Photoshop:* Graphics design studio.
6. Proof of Design

The Social Network Solution has multiple functionalities. Below are all the details of the functionalities.

6.1 Social Network Solution Installation

The first steps to implement the Social Network Solution involve downloading and installing the application on a Web server.

6.1.1 Download / Uncompress

Social Network Solution is available to download at http://www.stevefedak.com. Figure 3 displays the Web site and the link to download Social Network Solution.

Once the file a completely finished downloading the file needs to be uncompressed either by Windows or another compression tool that can handle .zip files, such as WinZip.
6.1.2 Installation Wizard

The installation wizard must be completed for the Social Network Solution to operate properly. The application can determine if the Web site is installed or not. If the Web site is not installed, the application will automatically redirect to the installation page. Figure 4 shows the welcome page of the installation wizard.

![Installation Wizard - Welcome](image)

**Figure 4: Installation Wizard - Welcome**

The first step is a welcome page that provides the user with helpful information about the installation process. The second page determines if the configuration file, used for global variables and database connection, is writable. The permission on the file needs to allow users to write to the file because the installation script determines information within the file. If the file is not writeable the application will attempt to
change the permissions, if this fails the user will have to change the permissions via FTP. Figure 5 shows the form the user must complete to access the database, determine the site name, description, and title, as well as create an optional administrative user.

![Database settings form](image)

**Figure 5: Installation Wizard – Database setup**

There is one last page that notifies the user if the installation was successful and information relative to the application. This page also provides a link to the index of the application.
6.2 User Registration and Login

The index of the Web site consists of login and registration forms, this page can be seen in Figure 6.

![Register / Login](image)

**Figure 6: Register / Login**

Before any user is allowed to view pages of the Web site they must register and login. When a user registers an account successfully, they are automatically logged in. If a user is returning to the site they will need to enter their email and password to login. Several forms of validation occur on both the registration and login forms. Error messages are displayed on the appropriate forms depending on the type of error.
6.3 Update User Profile

After a user has logged in they now have access to use all of the site functionality. Under the Profile part of the navigation there is an Update link that takes the user to an update form. Figure 7 shows a snapshot of the general information form.

![Figure 7: Update Profile](image)

There are four tabs or additional links at the top of the pages that allows the user to update different sections of their profile. These sections contain the following information:

- **General** – Includes name, gender, and birthday.
- **Picture** – Allows users to browse their computer for an image they would like to be displayed as their profile picture. This will create two files, a regular resized image to be viewed on the users main profile page and a thumbnail that is displayed when writing messages and on the friends list.

- **Contact** – Includes contact email address, cell phone number, city, state, and Instant Message screen name.

- **Interests** – A small survey to display likes and dislikes of the user. This form includes favorite movies, favorite music, favorite TV shows, favorite books, five things I can’t live without, ideal match, my idea of a perfect first date, in my bedroom you will find, and other interests.

After a user has filled out the profile forms the information will be displayed in their profile page. The user can choose to view their profile page from the navigation and may view profiles of other users.

### 6.4 Friends

There are several different aspects to friends and what the user can do with them.

To start networking a user should request some friends.

#### 6.4.1 Add Friends

On initial login there is a list of all the users signed up on the Web site. Figure 8 shows the list of users one might see when logged in.
Figure 8: User List

This list provides an easy way for users to find friends. If a user knows of a specific friend they would like to add as a friend the user can type their name in the search field and a list of results of people with that name will be presented. The list contains a thumbnail image on the users profile picture, their names, a link to view their profile, and a link to send them a message. Figure 9 is a screenshot of the “Add Friend” link when viewing another user’s profile.
The user needs to click on the view profile link then click on the add friend link under the users profile image. After the link it clicked the user is asked if they really request to add this user as a friend. Yes will send a request to the specified user and return the user to the other users profile and no will just return the user to the other users profile.

### 6.4.2 Accept Friend Requests

Once a friend request has been sent, the other user receives notification in his/her navigation bar. Figure 10 displays a friend request page.
The number of friend request will be displayed next to the friend request link. Once the user accepts or declines all user requests the number next to the link will disappear. After the user clicks on the friend request link all of their friend requests will appear on the page. The user has the option to accept, deny, or ignore the friend request. If confirmed the user will now be able to see that user in the friends list.
6.5 Messaging

There are many ways to send messages to other users. The user can choose to select send message in the navigation to send a message to any friends. The user can also send a message to a specific user from the friend list or a friend profile.

6.5.1 Send Messages

By selecting the send message link in the navigation a user is able to select any of his/her friends to message. Figure 11 shows the send message page.

![Send Message](image)

Figure 11: Send Message

The dropdown box is populated with all the users friends. The user may then enter a subject and message to send to the specified friend.
After the message has been sent the user will see a message that confirms the message has been successfully sent. If the user does not add a subject [No Subject] will appear as the subject.

6.5.2 Reply to Messages

When a user receives a message they receive notification in the navigation. Figure 12 displays the user’s inbox with one new message.

![Figure 12: Inbox](image)

The number of messages will be displayed next to the inbox link. There is an additional “(1)” after Inbox to signify the user has one new message.
When a user clicks to read the message in the inbox the number will not be displayed anymore.

The user may also reply to the conversation by clicking the reply link. This will take the user to the same form when sending the message but the username and subject will already be completed. From here the user can change the subject line but not change the user that will receive the message. The inbox page will display all conversations but will not display specific messages. To view these messages the user can click on the subject or message snippet to read the full conversation. Figure 13 is a screenshot of a conversation after the user clicked the message in their inbox.

Figure 13: Conversation View
6.5.3 Delete Conversations

Clicking the X in the top right hand corner of any conversation in a user’s inbox allows the user to delete a conversation. If a conversation is deleted by one user and the other user replies to the conversation, the user that deleted the conversation will be able to see the entire conversation again.

6.6 Photo Gallery

The photo gallery allows users to create albums and upload pictures into the created albums.

6.6.1 Create Album

Allows the user to create an album the user selects on the upload images link in the navigation. This link will display a hidden form that allows the user to create an album. Figure 14 displays the create album form after the create album link has been clicked.
Figure 14: Create Album

After the form is submitted the form will disappear and the album will show up in the dropdown box labeled album.

6.6.2 Upload Images

To upload images the user can select what album they would like the picture to be stored. Figure 15 is a screenshot of the upload image form after an album has been created.

Figure 15: Upload Image
The albums will be displayed in alphabetical order in the dropdown menu. The user can then select the browse button that allows the user to browse for an image they would like to upload.

6.6.3 View Album/Images

To view an album and images the user clicks on view on the navigation. Figure 16 is the page that shows the user’s albums they have created.

![Figure 16: Album View](image-url)
When a user clicks on an album all the images in that album will appear on the page. The user can then click on each thumbnail to see the full size image. Figure 17 is the page that displays the pictures in the album the user selected.

![Image View - Album](image.png)

Figure 17: Image View - Album

7. Testing

This application was tested by various user groups. While developing this project I conducted redundant tests to verify what I was developing. During the whole process Professor Annu Prabhakar looked at the progress of the project and made comments and suggestions. There was also a small user group of beta testers that used the social
network. These users created profiles, added friends, and sent messages between each other. These tests were done to get feedback from the users of the social networking experience and the functionality of the Web site as a whole. Finally, there was another small user group that conducted tests by installing the Social Network Solution. These testers gave me feedback on how to improve the installation process and new ideas to think about.

8. Conclusion and Recommendations

8.1 Conclusion

The Social Network Solution application makes it easy for Web developers to implement a working social network. The system provides developers an open source application that will allow them to create a social network with ease. They have the ability to edit the layout as well as to add or delete modules that will best suit the network they are trying to create. The application also allows novice Web programmers to easily make a Web site by using the installation wizard. Social Network Solution provides developers with a great tool to implement a social networking Web site.
Appendix A

The following code contains the structure of the registration functionality of the site.

```html
/* Smarty */
{load_user_registration assign="user_registration"}

<form action="{$user_registration->mActionPage}" method="post"
name="register" class="register">
<h2>Register</h2>
<label>E-mail Address</label><input type="text" name="txtEmail"
value="{$user_registration->mEmail}"/>
{if $user_registration->mTxtEmailError}
    <span class="error">You must enter an e-mail address.</span>
{/if}
<label>Name</label><input type="text" name="txtName"
value="{$user_registration->mName}"/>
{if $user_registration->mTxtNameError}
    <span class="error">You must enter your name.</span>
{/if}
<label>Password</label><input type="password"
name="txtPassword"/>
{if $user_registration->mTxtPasswordError}
    <span class="error">You must enter a password.</span>
{/if}
<label>Confirm Password</label><input type="password"
name="txtPasswordConfirm"/>
{if $user_registration->mTxtPasswordConfirmError}
    <span class="error">You must re-enter your password.</span>
{elseif $user_registration->mPasswordMatchError}
    <span class="error">You must re-enter the same password.</span>
{/if}
<input type="submit" name="sent" value="Confirm" class="submit"/>
<input type="button" value="Cancel" class="cancel"
onclick="window.location = '{$user_registration->mReturnPage}';"/>
</form>

Figure 18: User Registration Template
<?php
function smarty_function_load_user_registration($params, $smarty)
{
    $user_registration = new UserRegistration();
    $user_registration->init();
    $smarty->assign($params['assign'], $user_registration);
}

class UserRegistration
{
public $mEditMode=0;
public $mReturnPage;
public $mName;
public $mPhone;
public $mEmail;
public $mTxtNameError=0;
public $mTxtEmailError=0;
public $mTxtPasswordError=0;
public $mTxtPasswordConfirmError=0;
public $mPasswordMatchError=0;
public $mEmailAlreadyTaken=0;
private $mErrors=0;
private $mBoUser;
private $mHaveData=0;

function __construct()
{
    $this->mBoUser = new BOUser();
    $this->mActionPage = $_SERVER['REQUEST_URI'];
    $this->mReturnPage=
        substr($_SERVER['REQUEST_URI'],0,strlen($_SERVER['REQUEST_URI']));
    if (isset($_POST['sent'])) $this->mHaveData=1;
    if ($this->mHaveData==1)
    {
        if (empty($_POST['txtName']))
        {
            $this->mTxtNameError=1;
            $this->mErrors++;
        }
        else $this->mName=$_POST['txtName'];
        if ($this->mEditMode ==0 && empty($_POST['txtEmail']))
        {
            $this->mTxtEmailError=1;
            $this->mErrors++;
        }
        else $this->mEmail=$_POST['txtEmail'];
        if (empty($_POST['txtPassword']))
        {
            $this->mTxtPasswordError=1;
            $this->mErrors++;
        }
        else $this->mPassword=$_POST['txtPassword'];
        if (empty($_POST['txtPasswordConfirm']))
        {
            $this->mTxtPasswordConfirmError=1;
            $this->mErrors++;
        }
    }
}
```php
else $password_confirm=$_POST['txtPasswordConfirm'];
    if (!isset($password_confirm) || $this->mPassword != $password_confirm)
    {
        $this->mPasswordMatchError=1;
        $this->mErrors++;
    }
}
public function init()
{
    if (($this->mHaveData==1) && ($this->mErrors==0))
    {
        $user_read=$this->mBoUser->GetUserIdPassword($this->mEmail);
        if (!empty($user_read))
        {
            $this->mEmailAlreadyTaken=1;
            return;
        }
        $this->mBoUser->AddUserAndLogin($this->mName,$this->mEmail,$this->mPassword);
        if (isset($this->mReturnPage))
        {
            header("Location:".$this->mReturnPage);
            exit;
        }
    }
}<
```
public function AddUserAndLogin($name, $email, $password)
{
    $user_id = $this->AddUser($name, $email, $password);
    $_SESSION['id'] = $user_id;
}

public function AddUser($name, $email, $password)
{
    $hashed_password = PasswordHasher::Hash($password);
    $user_id = $this->mDoUser->AddUser($name, $email,
        $hashed_password);
    return $user_id;
}

<?php
class DoUser
{
    private $dbManager;
    function __construct()
    {
        $this->dbManager = $GLOBALS['gDbManager'];
    }
    public function GetUser($userId)
    {
        $query_string = "SELECT * FROM user WHERE id = $userId";
        $result = $this->dbManager->DbGetRow($query_string);
        return $result;
    }
    public function AddUser($name, $email, $password)
    {
        $query_string = "INSERT INTO user (name,email,password)
VALUES ('$name','$email','$password')";
        $this->dbManager->DbQuery($query_string);
        $query_string = "SELECT LAST_INSERT_ID()";
        $result = $this->dbManager->DbGetOne($query_string);
        return $result;
    }
    public function UpdateGeneralProfile($userId, $name, $gender, $birthmonth, $birthday, $birthyear)
    {
        $query_string = "UPDATE user SET name = '$name', gender = '$gender', month = '$birthmonth', day = '$birthday', year = '$birthyear' WHERE id = '$userId'";
        $result = $this->dbManager->DbQuery($query_string);
        return $result;
    }
}

Figure 20: User Registration Business Objects

Figure 21: User Registration Data Objects
References


