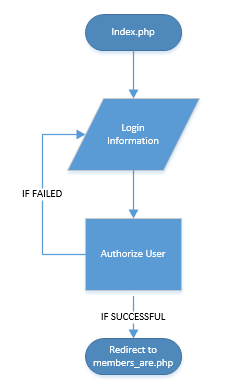
## *Index.php*

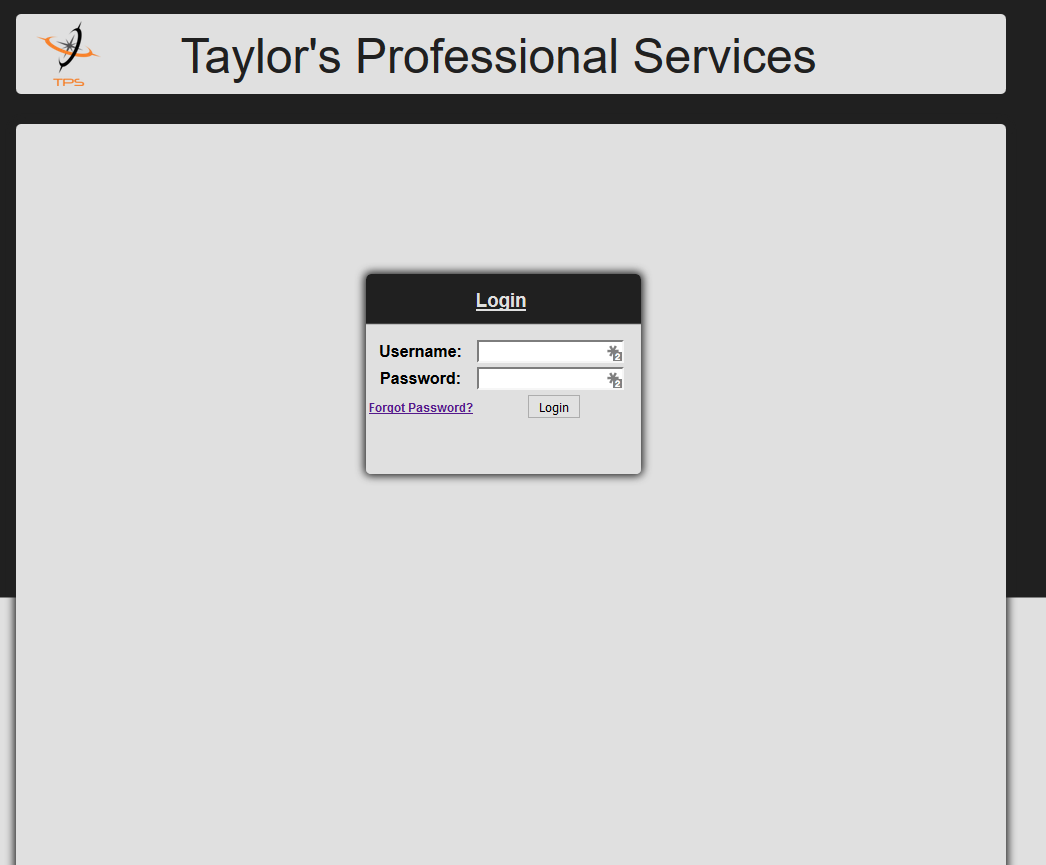
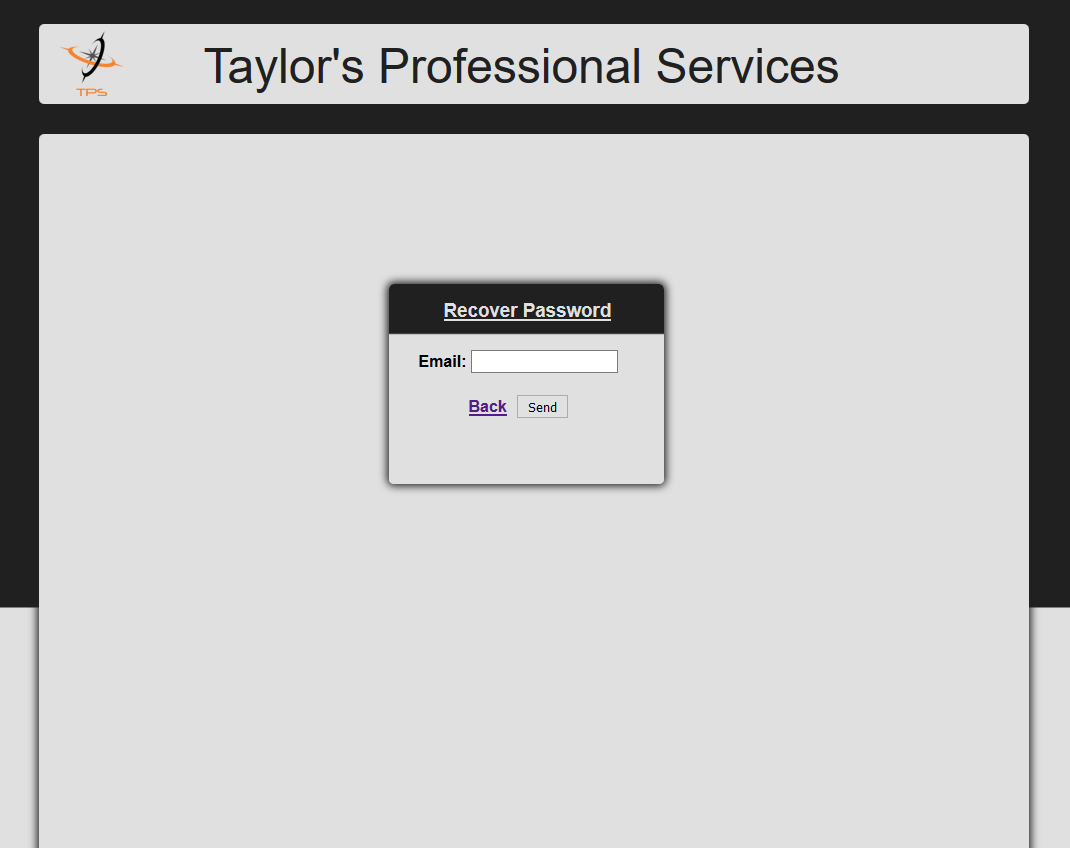
### Processing Description



### Interface Description

Main landing page which users are directed to upon navigating to the web domain.

* Login Form
  + Username Text Box – Users provide their username or employee id in this text box.
  + Password Text Box – Users provide their password in this text box.
  + Forgot Password Link – This link is used to send a new password to a valid email address in our system, in case users forget their password.
  + Login Button – This button submits the information to the server to have it verified.
* Forgot Password Form
  + Email Text Box – Users provide their email which is attached to an account in order to get a new password associated with their account.
  + Submit Button – This button submits the information to the server to have it verified.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

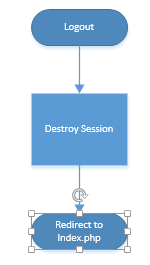
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

Used by all users to gain access to the members area of the website.

## *scripts/logout.php*

### Processing Description



### Interface Description

This component receives no data input, it provides no data output. When this component is called it simply destroys the session via the User Class.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

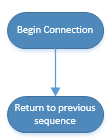
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This section does not apply to this component.

## scripts/database.php

### Processing Description



### Interface Description

This component receives no data. It creates a connection to the database for other components to use. This connection is a view only connection which does not allow the modification of database content.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

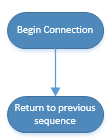
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This section does not apply to this component.

## *scripts/database\_admin.php*

### Processing Description



### Interface Description

This component receives no data. It creates a connection to the database for other components to use. This connection allows the modification of content in our database.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

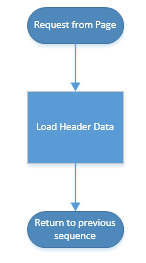
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This section does not apply to this component.

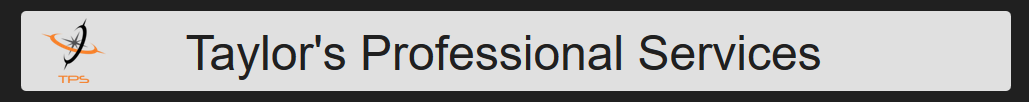
## *content/header.php*

### Processing Description



### Interface Description

This component receives no data input. This components output is the Logo, Company Name, and if there is an active session it will contain a logout link.



### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

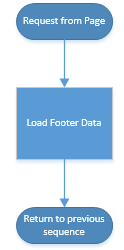
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used by all pages on our website. It provides the header for our website as content for the end user to view.

## content/footer.php

### Processing Description



### Interface Description

This component receives no input. This component outputs the copyright information for the website.



### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

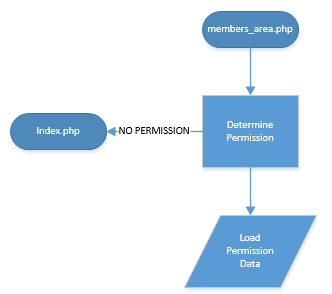
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used by all pages on our website. It provides the footer of our website as content for the end user to view.

## members\_area.php

### Processing Description



### Interface Description

This component receives no input. It obtains the session access if there is an active session and loads the appropriate files associated with the access level. If there is no active session it will redirect the user to the index.php page.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

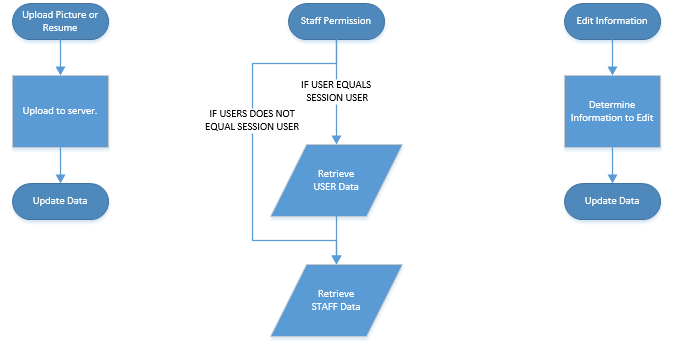
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is the core of our members area for our website. It acts as a content template which will load scripts and content based on permission and page requested by the end user.

## scripts/staff.php

### Processing Description



### Interface Description

This component contains no GUI information.

* Retrieves the profile ID from either the address or the session.
* Retrieves Staff Information via the Staff Class.
* Retrieves User Information via the User Class.
* Updates Staff Information via the Staff Class.
* Updates User Information via the User Class.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

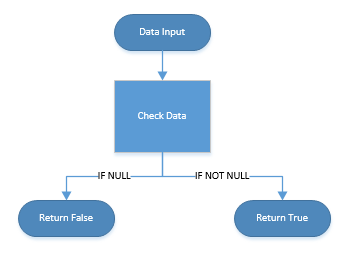
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used by our staff member’s area. When a staff member logs into our website we call these scripts from our members\_area.php page. Content pages for our staff member’s area sends data to this script file in order to retrieve or update their user information to/from the database.

## *scripts/javascript\_staff.php*

### Processing Description



### Interface Description

This component contains no GUI information.

* Prevents password to be sent to the server if password and confirm password do not match when staff members are editing their profile.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

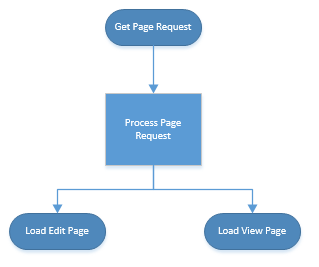
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used by our staff member’s area edit page. Its sole purpose is to provide a quick view for end users to denote when they have entered in an invalid password and confirmation password.

## content/staff.php

### Processing Description



### Interface Description

This component contains no GUI information.

* Loads content pages based on page requests by the end user.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

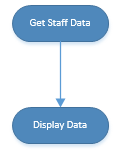
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This content component is used by our staff member’s area page. It determines if we should display the edit or the view page based on the page request.

## content/staff\_view.php

### Processing Description



### Interface Description

This component is strictly a GUI component.

* Edit Profile Button – Provides a way for the staff member to edit their profile.
* Picture – If the staff member has uploaded a picture it will be displayed.
* Availability Label – Displays whether the staff member is available for work or not.
* First Name Label – Displays the staff member’s first name.
* Last Name Label – Displays the staff member’s last name.
* City Label – Displays the city in which the staff member is located.
* State Label – Displays the state which the staff member is located.
* Zip Code Label – Displays the state which the staff member is located.
* Type of Work Label – Displays the type of work which the staff member has chosen.
* Experience Label – Displays the experience in years the staff members has in their field.
* Education Label – Displays the education level of the staff member.
* Salary Label – Displays the amount of money the staff member wants as their salary.
* Resume – Displays a link to the staff member’s resume, if one has been provided.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

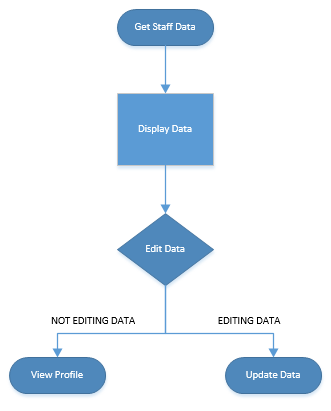
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

Its sole purpose is to provide a user interface to the end user for when they view staff member profiles.

## content/staff\_edit.php

### Processing Description



### Interface Description

This component is strictly a GUI component.

* View Profile Button – Provides a way for staff members to view their profile.
* Picture File Upload – Provides a way for staff members to upload their picture.
* E-Mail Text Box – Provides a way for staff members to change their email associated with the account.
* Current Password Text Box – Required to change either the email or password on the account.
* Password Text Box – Provides a way for staff members to change their password associates with the account.
* Confirm Password Text Box – Required to be the same as the password if the password is being changed.
* Submit Button – Submits the information to the server to be changed.
* Availability Selection – Provides a way for staff members to change their availability.
* First Name Text Box – Provides a way for staff members to change their first name.
* Last Name Text Box – Provides a way for staff members to change their last name.
* City Text Box – Provides a way for staff members to change the city in which they are located.
* State Selection – Provides a way for staff members to change the state in which they are located.
* Zip Code Text box – Provides a way for staff members to change the zip code in which they are located.
* Type of Work Selection – Provides a way for staff members to change the type of work they’re interested in.
* Experience Number Input – Provides a way for staff members to change the amount of years’ experience they have in their chosen field.
* Education Selection – Provides a way for staff members to change their level of education.
* Salary Input Box – Provides a way for staff members to change their desired salary.
* Submit Button – Submits the information to the server to be changed.
* Resume File Upload – Provides a way for staff members to upload their resume.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

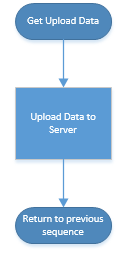
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

Its sole purpose is to provide a user interface to the end user for when they edit their profiles.

## scripts/upload.php

### Processing Description



### Interface Description

This component contains no GUI information.

* Retrieves information for the uploaded picture or resume by the staff member and edits their profile with the new information.
* Deletes old pictures or resumes from the server, which are no longer in use.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

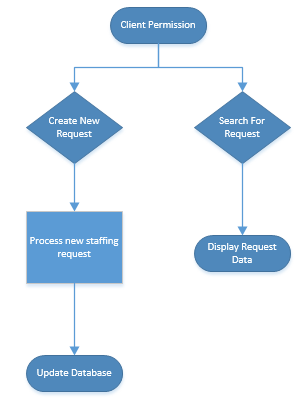
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used when a staff member uploads either a picture or a resume to the server. It accesses the database retrieves the old filename, deletes it from the server, and updates the profile with the new filename.

## *scripts/client.php*

### Processing Description



### Interface Description

This component contains no GUI information.

* Retrieves the staffing request id from the address bar.
* Retrieves Staffing Requests via the Client Class.
* Submits Staffing Requests via the Client Class.
* Searches for potential candidates in the database.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

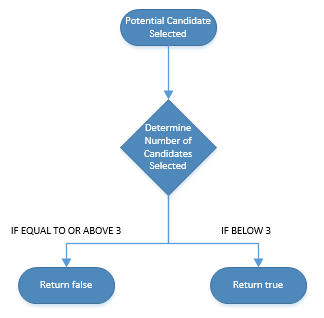
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used by our client member’s area. When a client member logs into our website we call these scripts from our members\_area.php page. Content pages for our client member’s area sends data to this script file in order to retrieve old or create new staffing requests.

## Scripts/javascript\_client.php

### Processing Description



### Interface Description

This component has not GUI.

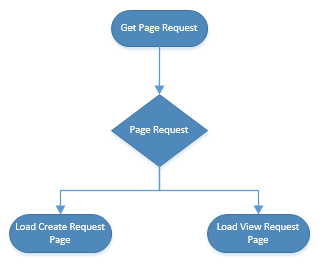
### Pseudocode

### Modules Uses

This component provides a way to monitor the amount of potential candidates selected.

## content/client.php

### Processing Description



### Interface Description

This component contains no GUI information.

* Loads content pages based on page requests by the end user.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

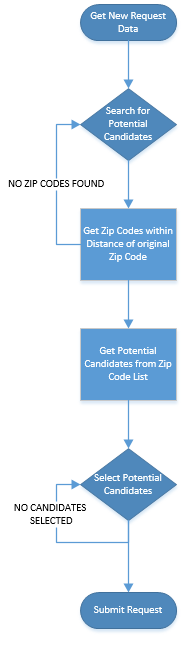
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This content component is used by our staff member’s area page. It determines if we should display the edit or the view page based on the page request.

## content/client\_request.php

### Processing Description



### Interface Description

This component is strictly a GUI component. It contains code which will be used to search for potential candidates and to submit new staffing requests.

* Zip Code Text Box – This text box will be used to contain a zip code for the new staffing request project.
* Distance Selection – This select component will be used to contain the amount of miles to search for new candidates.
* Type Of Work Selection – This select component will be used to contain the different types of work which clients are allowed to submit new staffing requests.
* Experience Number Input – This number input component will be used to contain the amount of years’ experience clients are looking for in the particular staffing request.
* Education Selection – This select component will be used to select the education level of potential candidates the clients are looking for in the particular staffing request.
* Salary Input Box – This input box will be used to contain the salary in which potential candidates must either meet or be below to be potential candidates.
* Search Button – This button will search for potential candidates.
* Potential Candidate Selection – This selection will contain the potential candidates for the staffing request. Potential Candidates will not be shown if they are not available for work.
* Submit Button – This button will submit the potential candidate and staffing request information to the database.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

Its sole purpose is to provide a user interface to the end user for when they are submitting a new staffing request.

## content/client\_viewrequest.php

### Processing Description



### Interface Description

This component is strictly a GUI component. It contains code which will be used when viewing previous staffing requests.

* Date Submitted Label – Contains the date which the staffing request was created.
* Zip Code Label – Contains the zip code of the staffing request.
* Type of Work Label – Contains the type of work for the staffing request.
* Experience Label – Contains the amount of experience required for the staffing request.
* Education Label – Contains the education level for the staffing request.
* Salary Label – Contains the max salary for the staffing request.
* Potential Candidate Component – Contains the selected potential candidates with names, pictures and a link to their profile (Contract Manager Only).
* Status Label – Contains the current status of the staffing request. VALID, INVALID, UNABLE TO FILL or FILLED. Will be a dropdown selection for Contract Managers instead of a label until it’s been set in stone.
* Submit Button (Contract Manager Only) – Provides a method of updating the staffing request with the correct status. Only shows if the status is VALID.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

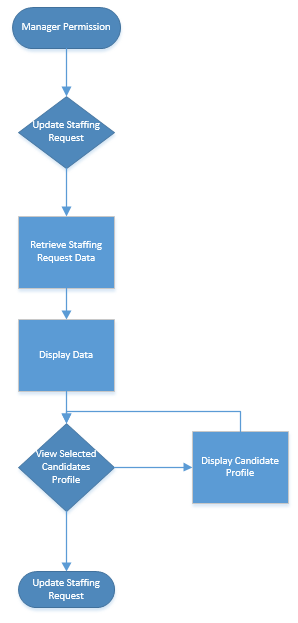
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

Its sole purpose is to provide a user interface to the end user for when they are viewing previous staffing requests.

## scripts/manager.php

### Processing Description



### Interface Description

This component contains no GUI information.

* Retrieves Staff Information via the Staff Class.
* Retrieves User Information via the User Class.
* Updates Staff Information via the Staff Class.
* Updates User Information via the User Class.
* Retrieves Staffing Requests via the Client Class.
* Submits Staffing Requests via the Client Class.
* Retrieves new Staffing Requests from the database.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used by our contract manager’s area. When a contract manager logs into our website we call these scripts from our members\_area.php page. Content pages for our contract manager area sends data to this script file in order to retrieve old or create new staffing requests.

## scripts/javascript\_manager.php

### Processing Description

This component has no use yet.

### Interface Description

This component has no use yet.

### Pseudocode

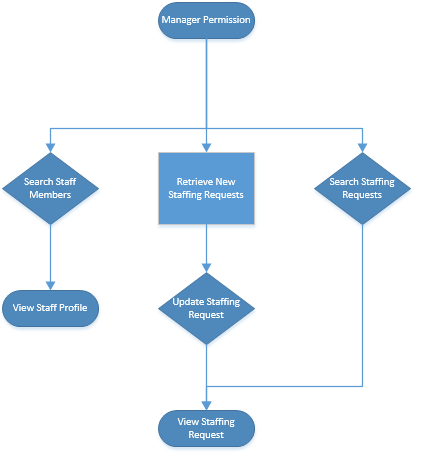
This component has no use yet.

### Modules Uses

This component has no use yet. The only reason it is included is to prevent PHP from throwing an error stating that it’s missing a file; this error is due to the generic way we are creating the content page for each member’s area.

## content/manager.php

### Processing Description



### Interface Description

This component is strictly a GUI component. It contains code which will be used when viewing previous staffing requests.

* Displays new staffing requests.
* View Staffing Request Link – Provides a link to the staffing request information.
* Allows Contract Managers to search for staffing requests.
* Staff Request ID Text Box – This text box will be used to contain the staffing request ID to search for in the database.
* Search Button – This button sends the staff request ID to the server to retrieve the information based on it.
* Allows Contract Managers to search for staff members.
* Staff Search Text Box – This text box will be used to contain the staff members information which we wish to search for in the database.
* Search Button – This button will send the search information to the server to retrieve the information based on it.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

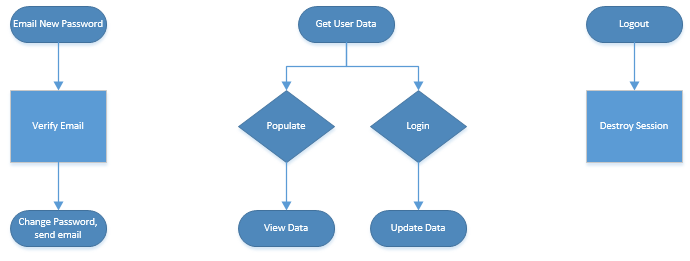
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used by all contract managers. Its purpose is to provide a user interface to the end user which will allow them to perform their daily tasks.

## User Class

### Processing Description



### Interface Description

This component has no GUI.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

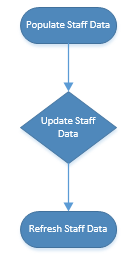
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used to retrieve and update user data from the database. It is also used to email new passwords for user accounts to a valid email address.

## Staff Class

### Processing Description



### Interface Description

This component has no GUI.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

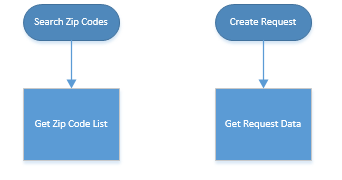
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used to retrieve and update staff data from the database. It contains three functions, one private called populate and two public, update and refresh. The populate function will populate data from the provided user ID. The update function will update the staff information in the database with the provided data in the class. The refresh function calls the populate function so we can access the most recent information in the database.

## Client Class

### Processing Description



### Interface Description

This component has no GUI.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

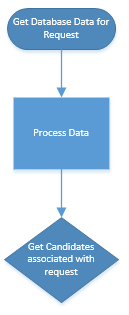
You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used to retrieve and update client data from the database. It also has the ability to create new staffing requests with provided information.

## Request Class

### Processing Description



### Interface Description

This component has no GUI.

### Pseudocode

The processing rules for the component – in logical order.  But remember, language independent. (link to a great example) <http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

Put things here if it is needed to help explain the business rule functionality of the screen. For example if you need to check to make sure a customer has no outstanding balance past 60 days in order for them to generate another order identify what data values need to be used to do this and the math functions that need to be used.

If Customer Balance > 0 and Last Purchase Date < (current date – 60 days) then   
 Deny Purchase  
Else  
 Allow Purchase

You can break this down into functions or procedures if that is the best way to approach this for your project.

### Modules Uses

This component is used to hold request data and retrieve candidate data from the database.