**Component: Persistence**

**Module Name: config.php**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
|  | Makes mysqli connection with database. |  |
|  |  |
|  |  |

**Inputs:**

**Process Description**

config.php creates a connection to the database. It is included on any other page in the application that needs to query the database.

**Outputs**

**APIs/Objects:**

Uses the mysqli API to create a connection object.

**Pseudocode:**

Set database parameters.

Create connection object.

If there’s an error, kill the connection.

**Component: Presentation**

**Module Name: Index.php**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
|  | Has links to other pages with descriptions of the major modules. |  |
|  |  |
|  |  |

**Inputs:**

**Process Description**

This page serves as the home page of the application. It holds links and descriptions to all of the major features of the application.

**Outputs**

**APIs/Objects:**

**Pseudocode:**

Set each link.

Print description of each link.

**Component: Presentation**

**Module Name: TicketMenu.php**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
|  | Presents clickable links that all direct to TicketSummary.php but which link is selected determines the output. | menu\_choice |
|  |  |
|  |  |

**Inputs:**

**Process Description**

Several links are placed on the screen. Each link directs to TicketSummary.php, but each link has a different value for menu\_choice in the URL. This will be input to the TicketSummary page.

**Outputs**

menu\_choice

**APIs/Objects:**

**Pseudocode:**

Set links on page, each will filter the ticket list differently based on each entry’s ticket status.

In the URL of each link, create variable menu\_choice and set that equal to All, Open, Resolved, Closed, or OnHold.

**Component: Presentation /Persistence**

**Module Name: TicketSummary**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
| Menu\_choice | Dynamically builds a sql query based on input then displays query results in a table. Buttons to View, Update or Delete each of the entries in the table. | ticket\_id |
| Keyword\_String |  |
| searchField |  |

**Inputs:**

Menu\_choice is from TicketMenu.php and contains information telling TicketSummary how to filter the ticket list based on ticket status.

Keyword\_String is from Search.php and contains string data that will be searched for in the database.

searchField is from Search.php and contains string data indicating which fields to search in.

**Process Description**

Dynamically builds sql statement based on input.

Organizes results into a table with a dynamic length.

Each entry has links to other major features View, Update, or Delete. All of these pass the specific ticket\_id to that page.

**Outputs**

Ticket\_id can be passed to TicketView.php, TicketUpdate.php, or delete.php where it will be used as the primary key to identify which ticket entry to operate on.

**Pseudocode:**

If menu\_choice is set, build sql statement based on its value where Status = menu\_choice.

Else if keywordString is set, build sql statement using all possible combinations of keyword and searchField to be used in LIKE statements in the WHERE clause.

Else query all tickets.

For each result, display result in a table row with links to TicketView.php, TicketUpdate.php, and delete.php. Include the ticket\_id in the URL as output.

**Component: Presentation/Persistence**

**Module Name: TicketView.php**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
| Ticket\_id | Query all fields for the entry that matches ticket\_id and display results. |  |
|  |  |
|  |  |

**Inputs:**

Ticket\_id is from TicketSummary.php and contains the primary key value to be looked up in a query.

**Process Description**

Build sql statement using ticket\_id. Then display that result in a table.

**Outputs**

**Pseudocode:**

Read in ticket\_id.

Create sql statement with ticket\_id.

Run query.

Display Results.

**Component: Persistence/Presentation**

**Module Name: TicketUpdate.php**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
| Ticket\_id | Builds a sql statement from ticket\_id, populates form, submits to UpdateSuccessful.php | Form data of every field |
|  |  |
|  |  |

**Inputs:**

Ticket\_id comes from TicketSummary.php and contains the info on which entry to update.

**Process Description**

Builds a sql statement from ticket\_id, populates a form and submits to UpdateSuccessful.php. All fields of the form will be initialized with their current values for both readability and security. Ticket\_id will also be read-only for security.

**Outputs**

Every field in our TicketDataSheet table will have its own output from this form. It will default to current value but will be overwritten by user input. This output is sent to UpdateSuccesful.php to build the sql statement and execute the query.

**Pseudocode:**

Build sql statement based on ticket\_id.

Query and store the result as the current data.

Create a form with an input field for each column in our database table.

Set the default value of each field equal to the current value.

On submission, link to UpdateSuccessful.php and POST the form data.

**Component: Persistence**

**Module Name: Update Successful**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
| Form data of every field | Dynamically build sql update statement based on input then execute query. |  |
|  |  |
|  |  |

**Inputs:**

Form data of every field

**Process Description**

Dynamically build sql update statement based on input then execute query.

**Outputs**

**Pseudocode:**

Store each value in $\_POST to a local variable.

Start sql update statement

For each field, check if it is NULL or an empty string based on the field type. If not NULL or Empty string, add to the sql statement to set each field equal to the new value from the form. If it is null or empty string, otherwise move on to the next field.

Finish sql update statement.

Execute Query.

Display whether successful or not.

**Component: Persistence**

**Module Name: delete.php**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
| Ticket\_id | Create sql delete statement based on input, then execute query. |  |
|  |  |
|  |  |

**Inputs:**

Ticket\_id from TicketSummary.php

**Process Description**

Create sql delete statement based on input, then execute query.

**Outputs**

**APIs/Objects:**

**Pseudocode:**

If ticket\_id is set, store it to a local variable.

Build sql delete statement where ‘ticket\_id’ = ticket\_id.

Execute the query.

Print whether it was successful or not.

**Component: Presentation**

**Module Name: search.php**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
|  | Create a form that allows the user to set searchField based on selection and enter a string for keywordString | searchField |
|  | keywordString |
|  |  |

**Inputs:**

**Process Description**

Create a form that allows the user to set searchField based on selection and enter a string for keywordString. On submission of the form redirect to TicketSummary.php with searchField and keywordString as output.

**Outputs**

searchField: an array of strings that represent which fields in the table to search for the keyword in.

keywordString: a string that will need to be parsed by white space that contains which substrings to search the table for.

**APIs/Objects:**

**Pseudocode:**

Create form.

Checkbox input allows selection of multiple enumerated search fields.

Text input allows entry of a String.

On submission redirect to TicketSummary.php and send searchField and keywordString as output.

**Component: Presentation**

**Module Name: Create**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
|  | Create a form to gather user input. Ouput data to Created.php | Form data of all fields |
|  |  |
|  |  |

**Inputs:**

**Process Description**

Create a form to gather user input. Certain fields are mandatory to create the bare minimum of a service ticket. On submission output all data fields suing post and redirect to Created.php

**Outputs**

Form data of all fields.

**APIs/Objects:**

**Pseudocode:**

Create form with input to match each type of field in the table.

Make critical fields required.

Submission of the form uses post to output all fields and redirects to Created.php

**Component: Persistence**

**Module Name: Created**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
| Form Data of all Fields | Dynamically create sql insert statement using input. Execute query. Print if successful and redirect to TicketSummary.php | Menu\_choice |
|  |  |
|  |  |

**Inputs:**

Form data of all fields in the table.

**Process Description**

Dynamically creates sql insert statement using input. Executes query. Prints if successful and redirects to TicketSummary.php with menu\_choice set to All.

**Outputs**

Menu\_choice

**Pseudocode:**

Save each $\_POST element as a local variable.

Start sql Insert statement and start sql statement holding values to be inserted.

For each field variable, if it is not null or empty string (based on the data type) add the name of the field to the first sql statement and add the value of the field to the second sql statement, otherwise, move on to the next field variable.

Concatenate the two sql statements.

Execute query.

Print if it is successful.

Redirect to TicketSummary.php with menu\_choice = All

**Component: Presentation**

**Module Name: functions.php**

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
|  | Creates a menu bar header that is included on all other user facing pages. |  |
|  |  |
|  |  |

**Inputs:**

**Process Description**

Creates a menu bar header that is included on all other user facing pages. Menu bar has navigation links to the major functions of the application.

**Outputs**

**Pseudocode:**

pdo\_connect\_mysql function:

Makes connection to sql database using PDO connection.

template\_header containing navigation bar

Styling for nav bar

Home anchor tag link back to index page

Create anchor tag link to create view

Ticket menu anchor tag links to view list of tickets

Search anchor tag links to search.php   
template\_footer

contain empty body