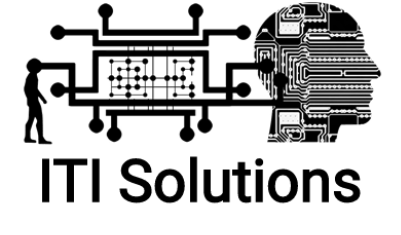
2/15/2020

Milestone 7

Technical Design



Aidan Campbell Beryon Clark Harley Lenton Justin Alho Evan Guest

ITI Solutions

Client Contact: Rachel Frantz

**Client Signature:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   **Date:** \_\_\_\_\_\_\_\_\_\_\_\_

Edenbridge Family Services

Scheduling and Time-Tracking Database

Software: Schedule ED

Table of Contents

[Current System 2](#_Toc32703208)

[Input Forms 3](#_Toc32703209)

[Input Client 3](#_Toc32703210)

[Update Client 4](#_Toc32703211)

[Input Staff 5](#_Toc32703212)

[Update Staff 6](#_Toc32703213)

[Schedule Shift 7](#_Toc32703214)

[Update Shift 8](#_Toc32703215)

[Update Recurring Shift 9](#_Toc32703216)

[Input Group Home 9](#_Toc32703217)

[Update Group Home 10](#_Toc32703218)

[Input Department 10](#_Toc32703219)

[Update Department 11](#_Toc32703220)

[Search for Staff/Client 11](#_Toc32703221)

[Timesheet Submission 12](#_Toc32703222)

[Outputs 13](#_Toc32703223)

[View Staff Information 13](#_Toc32703224)

[View Client Information 14](#_Toc32703225)

[Calendar View 15](#_Toc32703226)

[Detailed Shift View 16](#_Toc32703227)

[Reports 17](#_Toc32703228)

[Structure charts 18](#_Toc32703229)

[Program Specifications 44](#_Toc32703230)

[User Interface 110](#_Toc32703231)

[Login 110](#_Toc32703232)

[Worker Interface 111](#_Toc32703233)

[View Schedule 112](#_Toc32703234)

[View Details 113](#_Toc32703235)

[Submit Hours 114](#_Toc32703236)

[Bookkeeper Interface 115](#_Toc32703237)

[View Reports 116](#_Toc32703238)

[Search 117](#_Toc32703239)

[Coordinator Interface 118](#_Toc32703240)

[Add Staff Member 119](#_Toc32703241)

[Edit Staff Member 120](#_Toc32703242)

[Add Client 121](#_Toc32703243)

[Edit Client 122](#_Toc32703244)

[Add Group Home 123](#_Toc32703245)

[Edit Group Home 124](#_Toc32703246)

[Add Department 125](#_Toc32703247)

[Edit Department 126](#_Toc32703248)

[Schedule Shift 127](#_Toc32703249)

[Edit Shift 128](#_Toc32703250)

[Edit Recurring Shifts 129](#_Toc32703251)

[Schedule Recurring Shifts 130](#_Toc32703252)

[Approve Hours 131](#_Toc32703253)

[Search 133](#_Toc32703254)

[Logout 134](#_Toc32703255)

[Additional Process 135](#_Toc32703256)

[Backup 135](#_Toc32703257)

[Restoration 135](#_Toc32703258)

[Help 135](#_Toc32703259)

[Physical & Technical Architecture 136](#_Toc32703260)

[General Overview of Use 136](#_Toc32703261)

[Manual Processes 136](#_Toc32703262)

[Timesheet Submission 136](#_Toc32703263)

[Shift Creation 136](#_Toc32703264)

[Shift Rescheduling 137](#_Toc32703265)

[Shift Verification 137](#_Toc32703266)

[User Creation & Management 137](#_Toc32703267)

[Reporting Processes 137](#_Toc32703268)

[Restoration Process 137](#_Toc32703269)

[Requisite Software 137](#_Toc32703270)

[MySQL Database 137](#_Toc32703271)

[Apache 138](#_Toc32703272)

[PHP – PHP: Hypertext Pre-Processor 138](#_Toc32703273)

[FPDF – Free PDF PHP Libraries 138](#_Toc32703274)

[PEAR – PHP Libraries 138](#_Toc32703275)

[Data Distribution 138](#_Toc32703276)

[Client-Server Architecture 138](#_Toc32703277)

[Required Hardware 139](#_Toc32703278)

[Lessons Learned 140](#_Toc32703279)

# Current System

The system that has been requested for us to replace at Edenbridge is a mixture of paper forms and Microsoft Excel spreadsheets. In short, this system contains some mass inefficiencies that drastically lower the potential productivity at Edenbridge. Currently, Edenbridge uses a paper-based system in multiple binders shared between the coordinators. This makes it difficult to make sure the workers are not working overtime or being paid for work they have not done because the coordinators need to physically go over to the other coordinators’ binders and look at what shifts the workers already have.

# Input Forms

## Input Client

A screenshot of a cell phone

Description automatically generated

This form is for users with administrator capabilities to be able to add new client entries to the database. Some text inputs like Client ID will be pre-filled, but most other sections are required to be filled out manually.

## Update Client

A screenshot of a cell phone

Description automatically generated

Users with administrator permissions will be able to access this form. The form’s main intention is to allow a user with requisite permissions to adjust information regarding a specific client. The fields in this form will auto-fill with the client’s information, allowing for easy adjustment of a single value.

## Input Staff

A screenshot of a social media post

Description automatically generated

This form allows users with administrator permissions to enter information regarding new employees. Staff ID will be prefilled with a number unique ID polled from the worker table in the database. Other fields will be entered as required.

## Update Staff

A screenshot of a cell phone

Description automatically generated

This form allows users with administrator permissions to edit employee information. As with the form allowing users to edit client information, fields in this form will be automatically filled to allow for ease of adjusting information.

## Schedule Shift

A screenshot of a cell phone

Description automatically generated

This form allows users with coordinator permissions to schedule new shifts. Within this form, there are several ease of use options, including quick searching for a specific client, utilizing features embedded into browsers for date selection, and the ability to designate a shift as recurring. In the case of a shift being recurring, this form will output to two different tables, with information from this form being exported to both.

## Update Shift

A screenshot of a cell phone

Description automatically generated

This form enables users with coordinator privileges to update information pertaining to a shift. As with the client and employee forms of a similar nature, the fields in this form will autofill for ease of adjusting.

## Update Recurring Shift

A screenshot of a cell phone

Description automatically generated

This form enables users with coordinator privileges to be able to edit information pertaining to recurring shifts. Auto-filling forms allow for ease of adjustment.

## Input Group Home

## 

A screenshot of a cell phone

Description automatically generated

Form pertaining to adding new group homes, will only be available to users with administrator privileges.

## Update Group Home

A screenshot of a social media post

Description automatically generated

This form enables users with administrator privileges to edit information in the group home table. Fields in this form will auto-fill with information pertaining to a specific entry, allowing changes to the active supervisor of a group home or the phone number associated with it.

## Input Department

A screenshot of a cell phone

Description automatically generated

This form allows users with administrator privileges to add departments.

## Update Department

A screenshot of a cell phone

Description automatically generated

This form enables users with administrator permissions to edit department information. This includes the department code, with all fields being pre-filled.

## Search for Staff/Client

A screenshot of a cell phone

Description automatically generated

This is a sub-form that allows users to search for staff with a given criteria. This is used often when making reports, and the drop-down will be modified via AJAX to accommodate changes in search criteria.

## Timesheet Submission

A screenshot of a cell phone

Description automatically generated

After shifts have been worked, the staff need to submit their hours. This form will automatically populate with the start and end times for shifts that have the “worked” status for the latest pay period, and will include the name of the client as well as a colour code to identify what departments they worked shifts in. If times shown are accurate, then changes do not need to be made. Should they be incorrect, the staff member can modify either the start or end time to reflect the accurate times. After the timesheet is ready to be submitted, staff must first check the box that confirms the hours they submitted are accurate. Once they have done that, they can click the submit button to add their start and end times to the claimed start and claimed end fields for each shift. Doing this allows the system to begin processing the data and for coordinators to review and verify outstanding discrepancies.

# Outputs

## View Staff Information

A screenshot of a social media post

Description automatically generated

This form is modelled after the form allowing the creation/editing of staff members. While the format is similar, a primary difference is that this form will not allow any changes. All fields present in this page will be automatically filled and unavailable for editing.

## View Client Information

A screenshot of a social media post

Description automatically generated

As with the view worker output, this form acts as an output to view client information. The formatting and layout are similar to the forms for adding/editing client information, with a key difference in that all fields will remain non-active.

## Calendar View

A screenshot of a cell phone

Description automatically generated

This mock-up models what staff will be able to see when looking for shifts they are scheduled for or coordinators when viewing editable shifts. While staff will only be able to view the shifts they are directly scheduled for, coordinators can display a calendar populated with shifts and filtered by either staff, client, or department. The logic involved in displaying the calendar will select all shifts in the shift table between the start and end of a month with filtering specified for a staff ID, client ID, or department code. Names and IDs will be closely intertwined in this process, as a user will invariably use one or the other to filter through entries. With this filtering in place, the calendar will display shifts fitting criteria on the specified day, with the start/end times and the client’s name visible on the overview. Shifts displayed will be colour coded based on department, with an exception in place for those under the client name “Group Home”. For these shifts, they will be colour coded by which group home is specified, as separate group homes are considered different departments. Group home shifts will have the name on the shift be replaced with the department name.

## Detailed Shift View

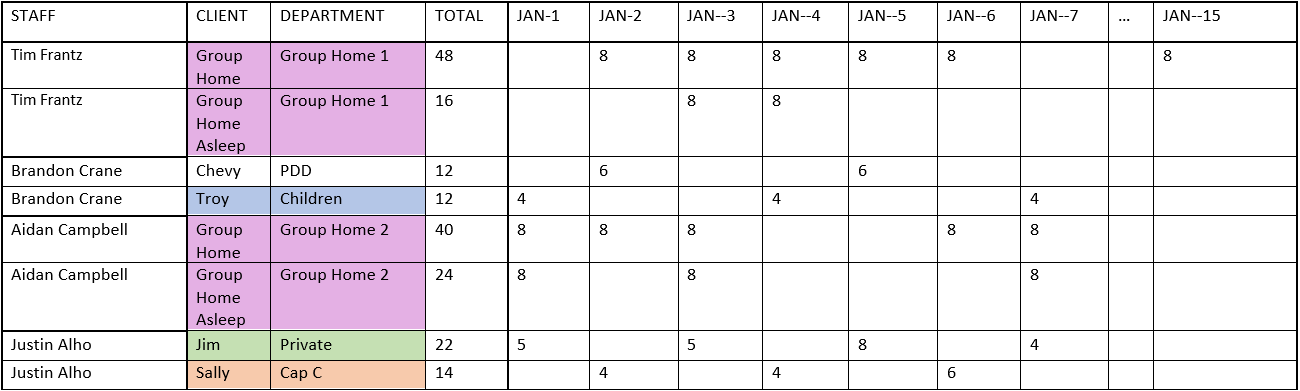
When viewing shifts in the calendar view, staff will be able to click on an individual shift to view it in greater detail.

A screenshot of a cell phone

Description automatically generated

Invoking this sub-menu logically involves selecting shift date, scheduled start/end times, and shift notes from the shift table. Also selected is the client name, client address, client phone number, and client notes from the client table. These are both pulled from the relevant records in the shift and client tables of the selected of the selected shift.

## Reports



The second half of the core components is time tracking and as such, a method is required for bookkeepers to view submitted hours. To serve this purpose, reports are required such as one like the example above. This style of report would have a user specify a date range and then query the database for shift entries with both the “approved” status and dates matching criteria. These reports will be modifiable with multiple fields and sorting options. Filtering option will additionally be available to restrict displayed entries to hours worked for a single client, or by department.

# Structure charts

A close up of a map

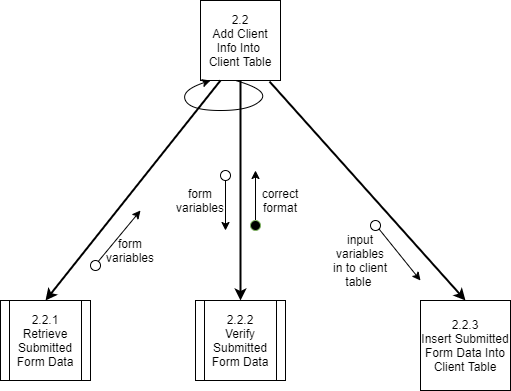
Description automatically generated

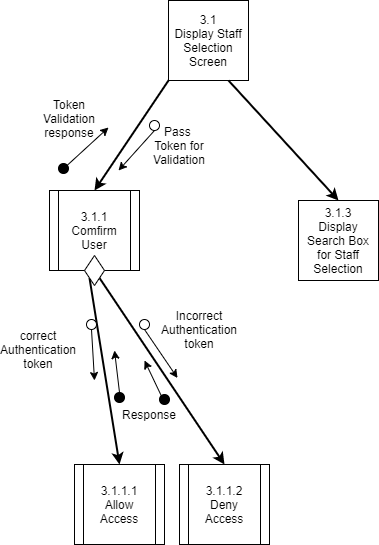
A close up of a map

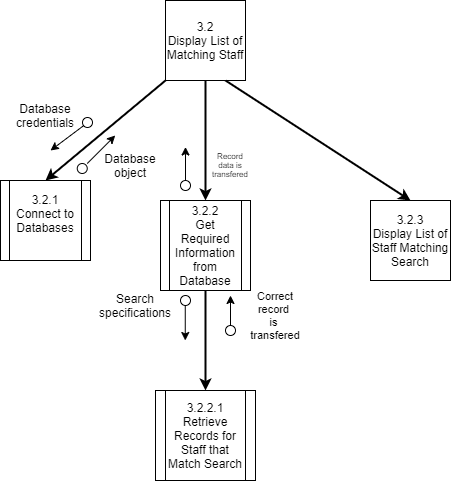
Description automatically generated

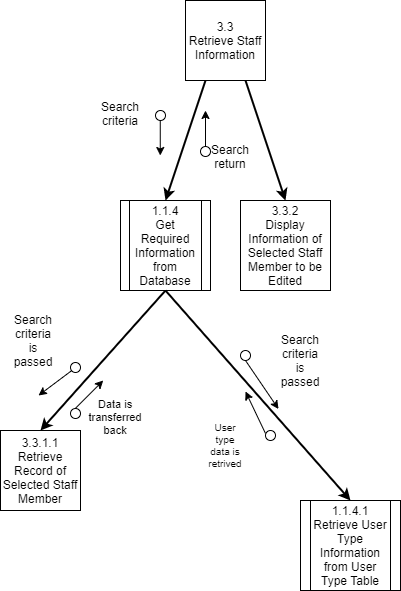
A close up of a map

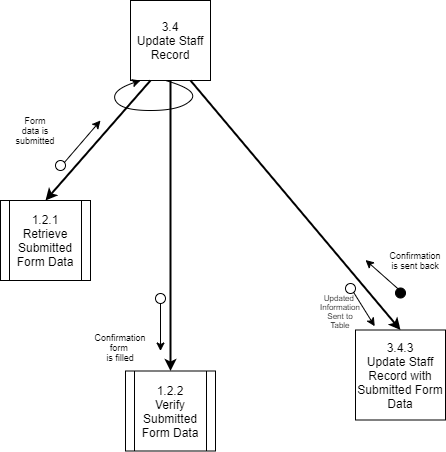
Description automatically generated

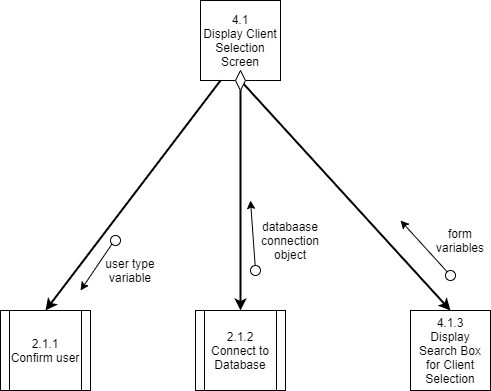


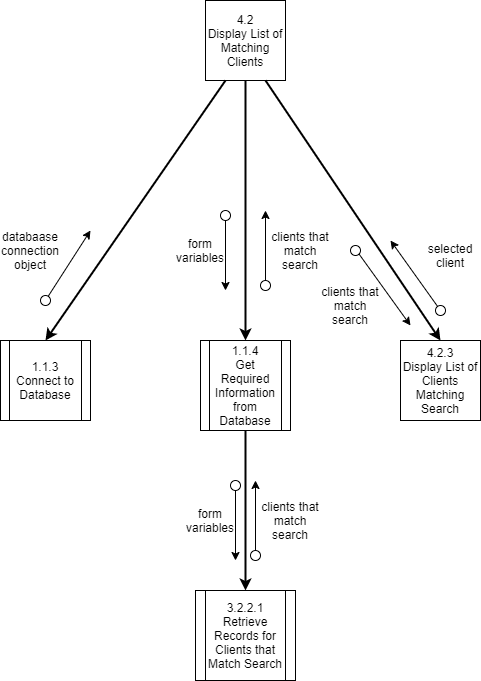


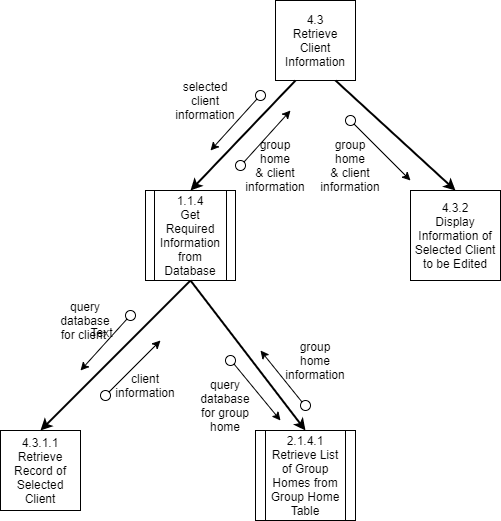


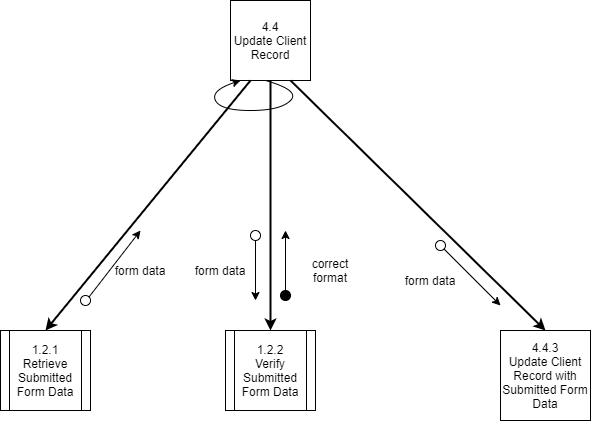


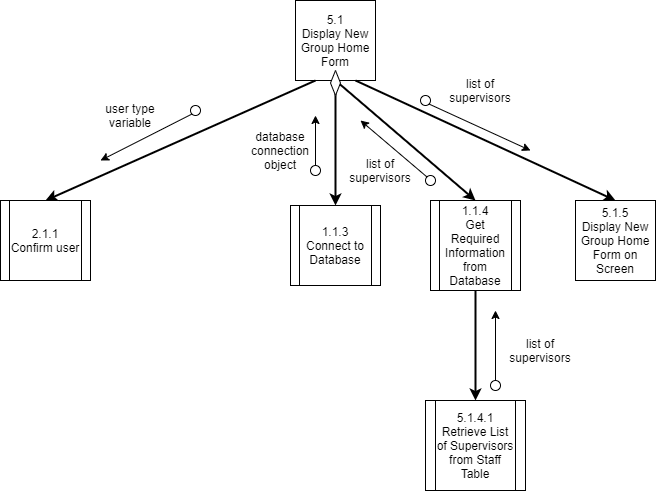




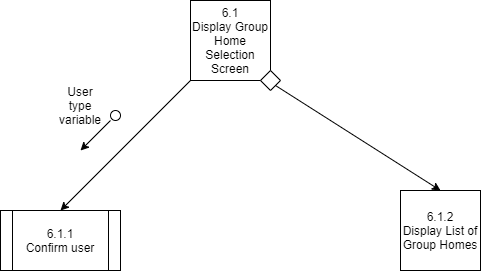


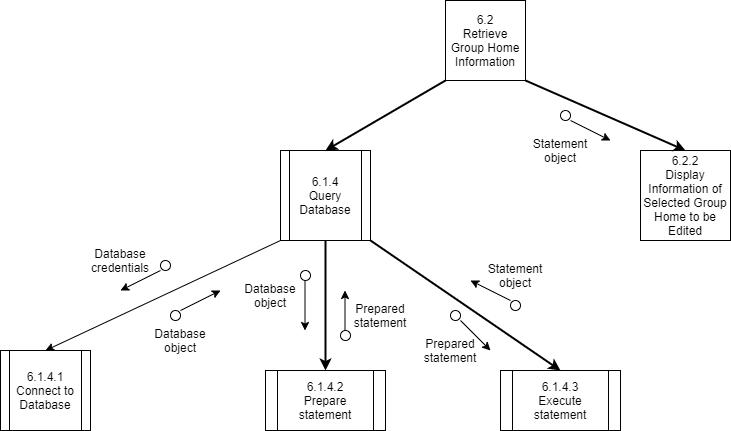


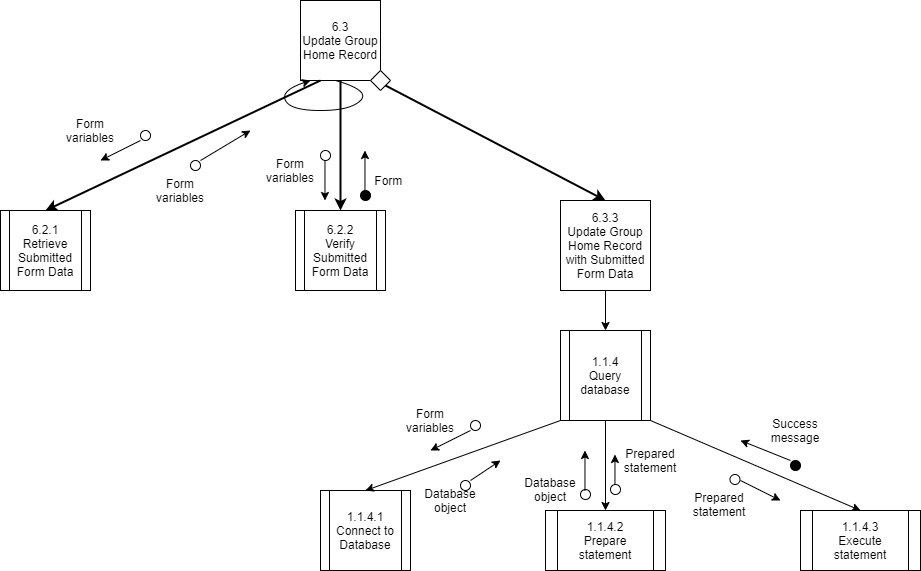


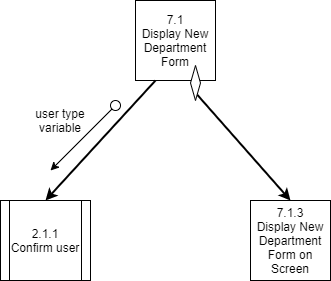
A close up of a map

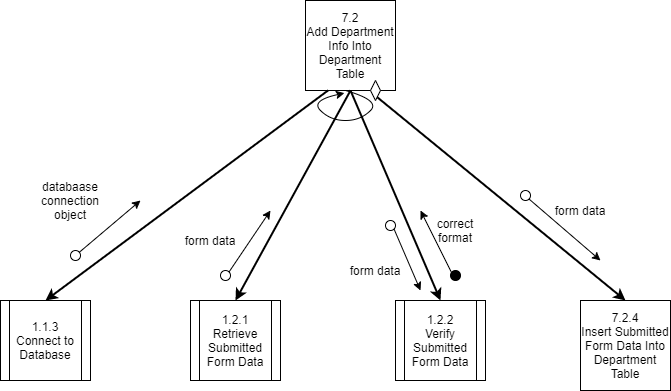
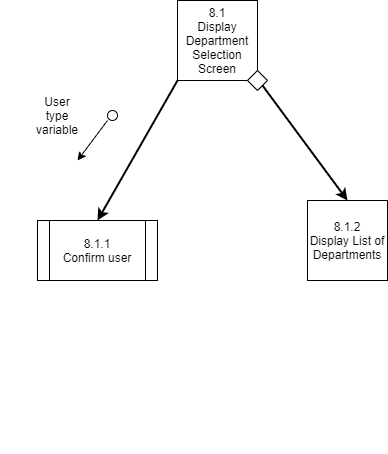
Description automatically generated

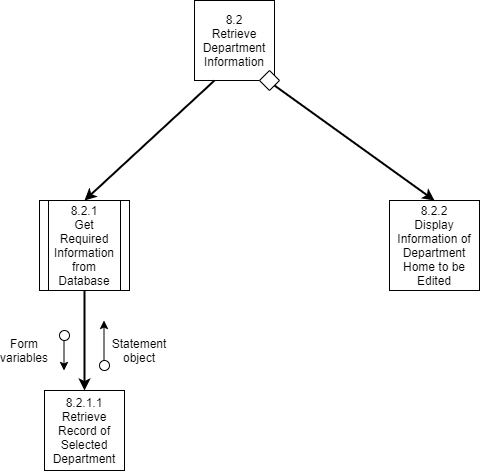


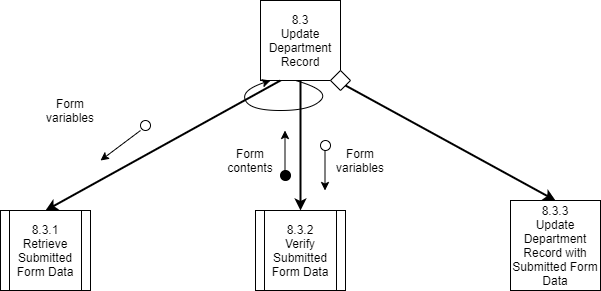


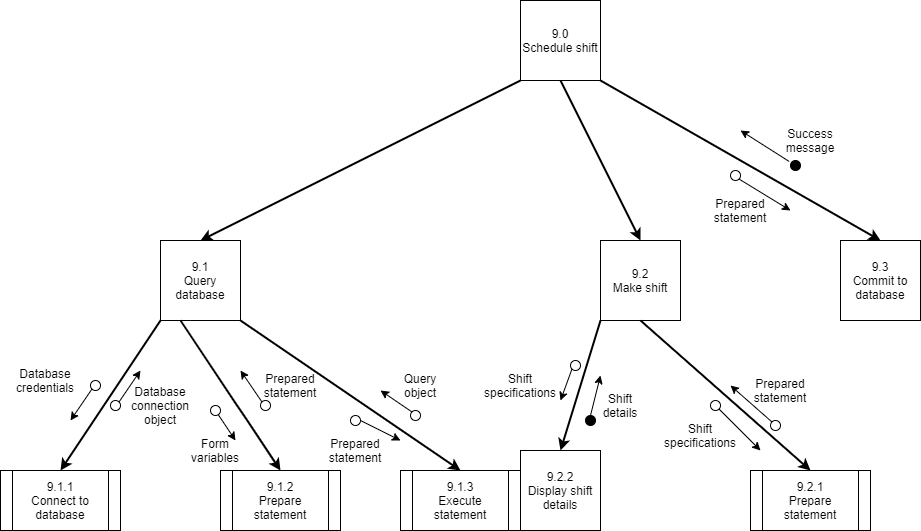


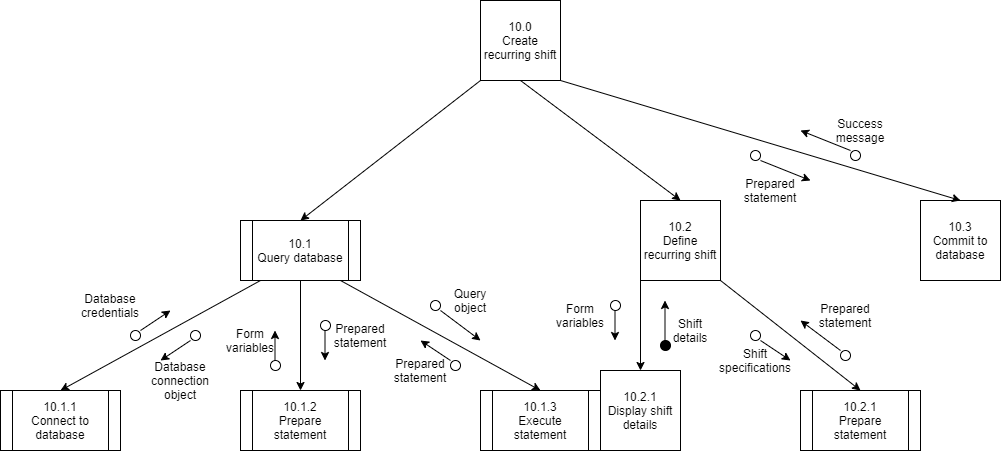


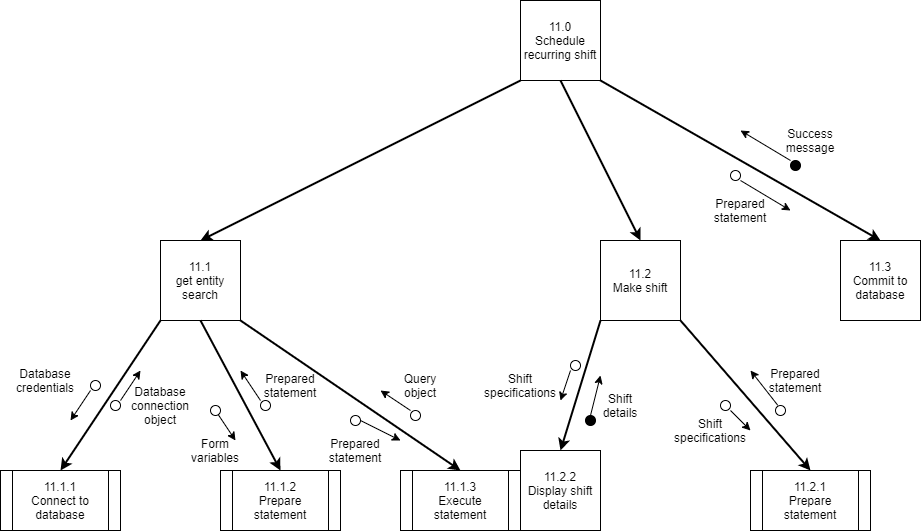


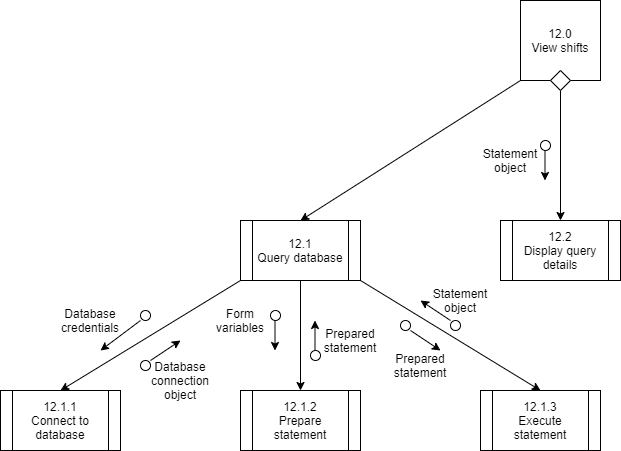


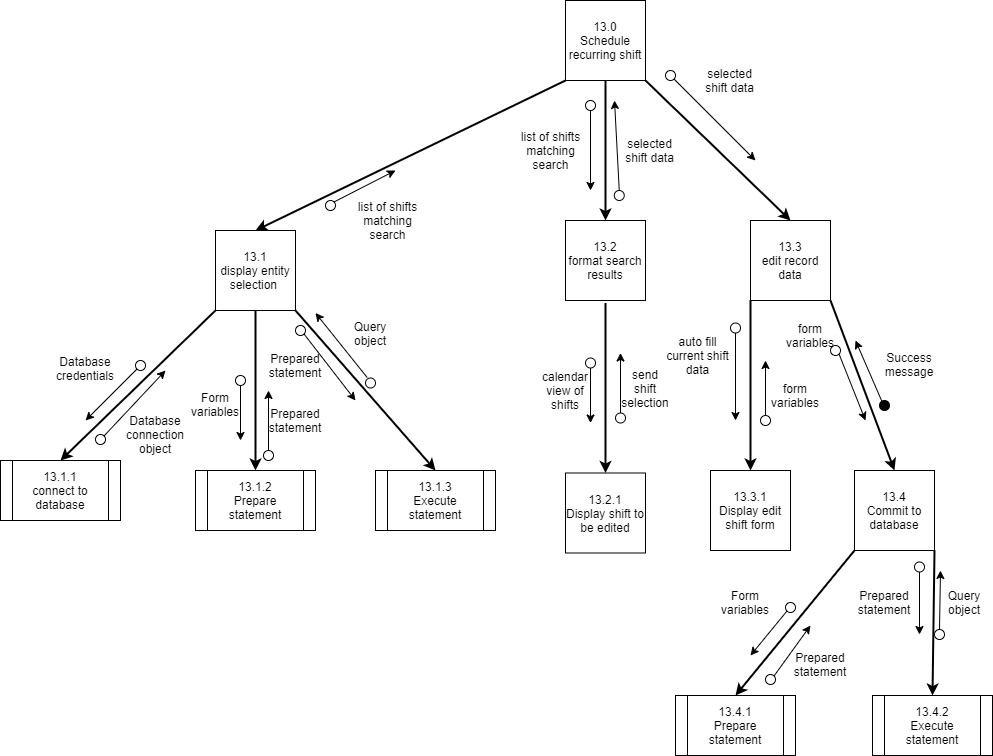


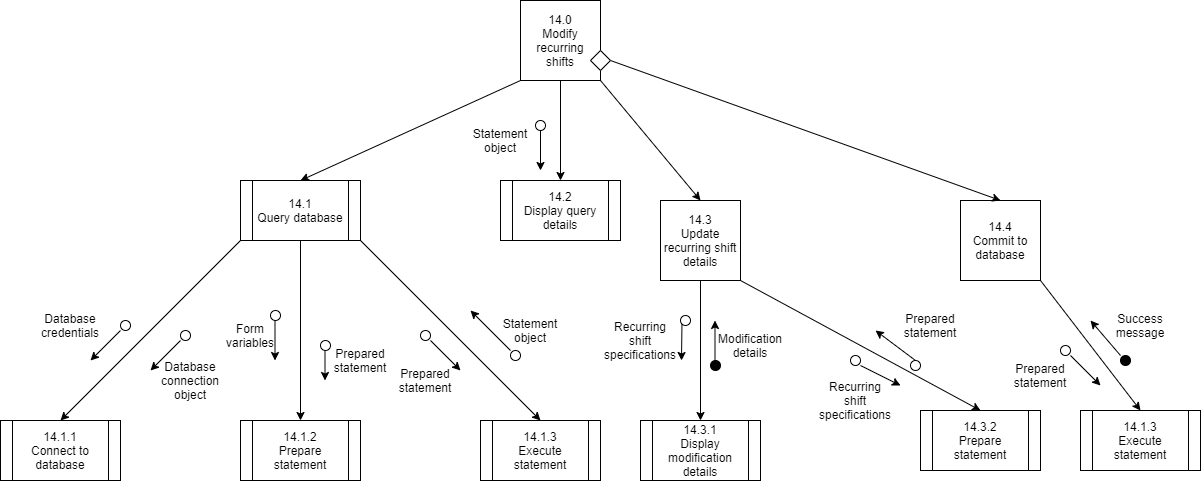


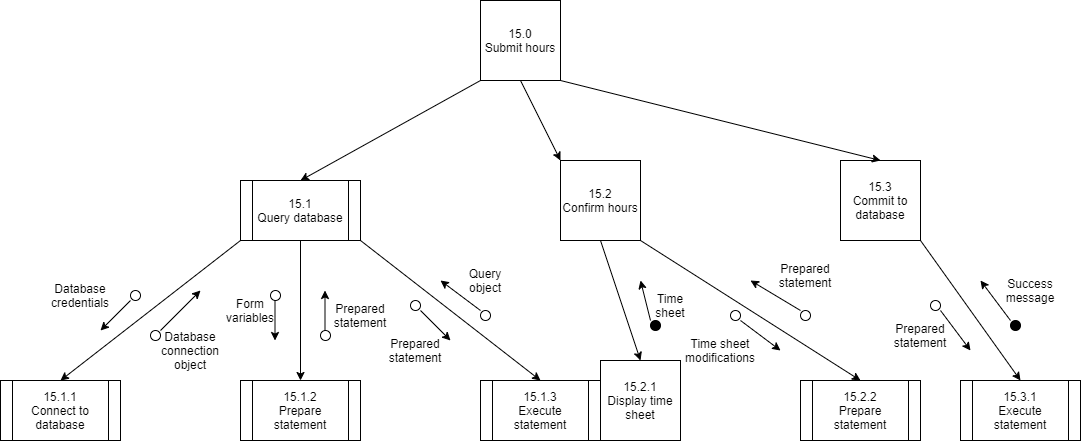








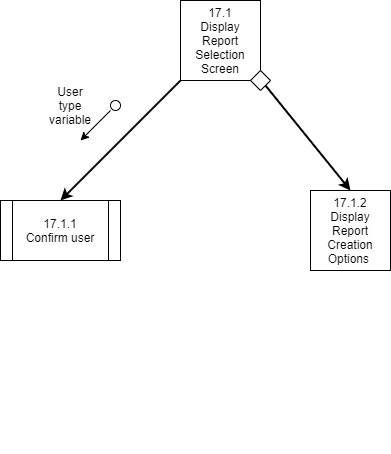


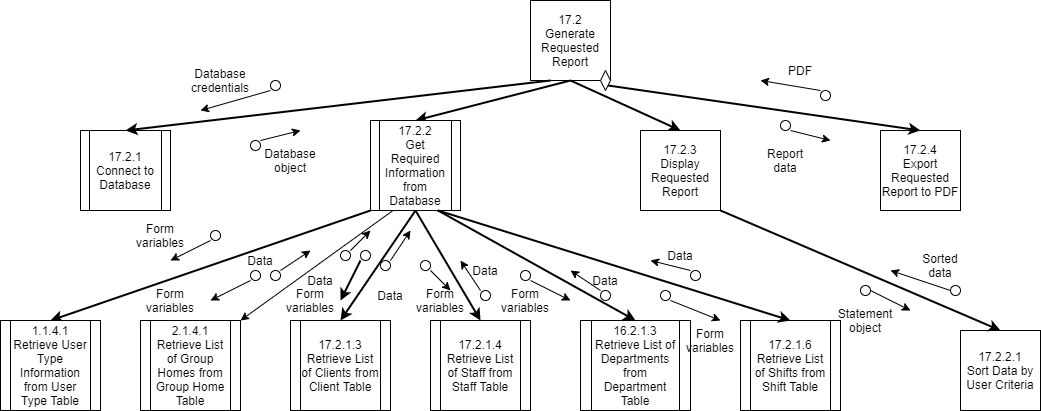
A close up of a map

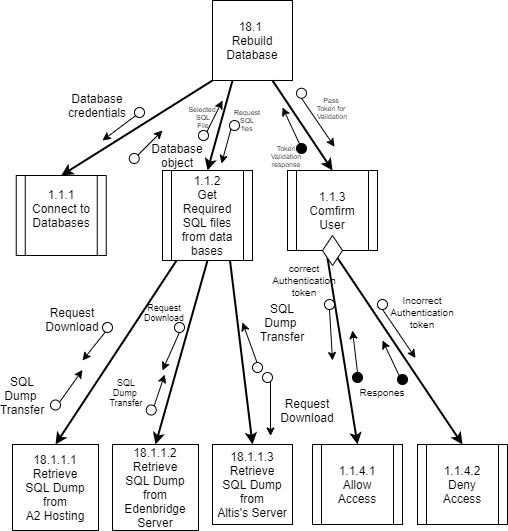
Description automatically generatedA close up of a map

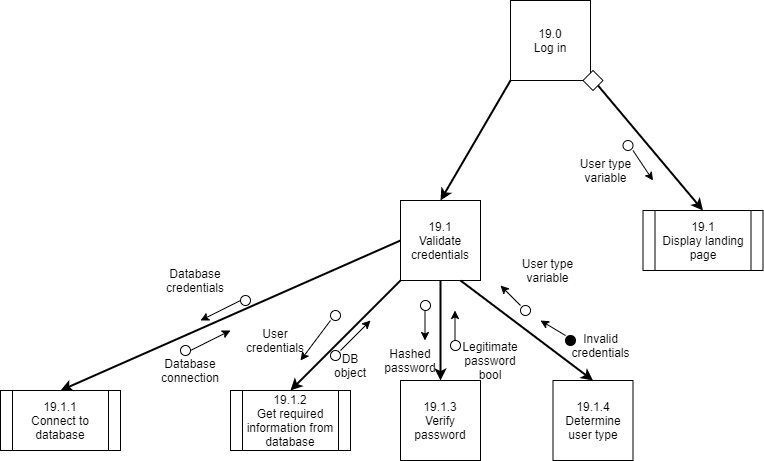
Description automatically generatedA close up of a map

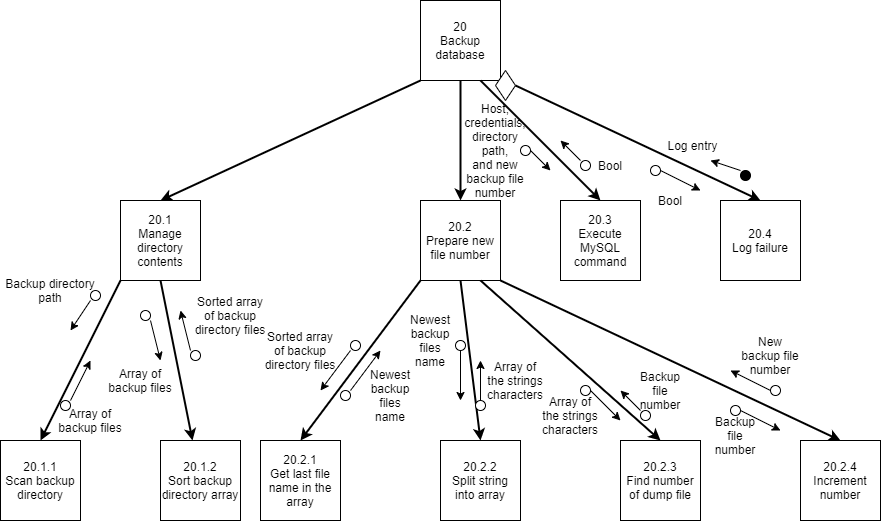
Description automatically generated











# Program Specifications

**Program specification 1.1.1 for 1.1 Display New Staff Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Confirm User

Purpose: Ensure the user trying to view the page is supposed to be there

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: A user is trying to access a page that should not be visible by unauthorized visitors

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| $\_SESSION | Global Array | 1.1.1 | This module retrieves the session information to see if users are logged in |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| loggedIn | boolean | 1.1 | loggedIn refers to whether the user is logged in, passed to the next modules to perform different tasks |

Pseudocode:

If $\_SESSION[“loggedIn”] = 1 and if $\_SESSION[“userType”] is equal to a valid user type

loggedIn = true

else

loggedIn = false

**Program specification 1.1.2 for 1.1 Display New Staff Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Query Database

Purpose: Retrieve Required Information from the Database

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: Information needs to be retrieved from the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Form variables | array | 1.1.2 | This module retrieves information from an HTML form |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Database Credentials | string | 1.1.2.1 | Credentials to create a database connection |
| Form variables | array | 1.1.2.2 | Retrieved information from the form |
| Prepared statement | string | 1.1.2.3 | Prepared SQL statement to query the database |

Pseudocode:

Call 1.1.2.1

Call 1.1.2.2

Call 1.1.2.3

**Program specification 1.1.2.1 for 1.1 Display New Staff Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Connect to Database

Purpose: Create a database connection

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML, SQL

Events: Process 1.1.2 provides it with database credentials

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Database credentials | string | 1.1.2.1 | This string is used to connect to the database |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Database connection object | MySQL object | 1.1.2.2 | Database connection is created and passed as an object to the next process |

Pseudocode:

Create new MySQL connection using database credentials

**Program specification 1.1.2.2 for 1.1 Display New Staff Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Prepare Statement

Purpose: Prepare an SQL statement

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML, SQL

Events: A database connection has been created and an SQL statement needs to be prepared to retrieve information from the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Form variables | array | 1.1.2.2 | These are the variables that need to be put into the SQL statement |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Prepared statement | string | 1.1.2.3 | Prepared statement that will be used to query the database |

Pseudocode:

Create SQL statement that uses form variables

**Program specification 1.1.2.3 for 1.1 Display New Staff Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Execute Statement

Purpose: Execute the prepared SQL statement

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML, SQL

Events: An SQL statement has been created

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Prepared statement | string | 1.1.2.3 | This is the statement prepared in 1.1.2.2 |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Statement object | object | 1.1.2 | Object storing the requested data from the database |

Pseudocode:

Execute(Prepared statement)

**Program specification 1.1.3 for 1.1 Display New Staff Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display New Staff Form on Screen

Purpose: Display new staff form on screen

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: Required information has been retrieved from the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Statement object | object | 1.1.3 | This is the information retrieved from the database in 1.1.2.3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Statement Object | object | 1.1.3.1 | Object storing the requested data, being passed to the 1.1.3.1 process |

Pseudocode:

<?php

<html>

<new staff form>

</html>

?>

**Program specification 1.1.3.1 for 1.1 Display New Staff Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display query details

Purpose: Display the information retrieved from the database

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: Required information has been retrieved from the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Statement object | object | 1.1.3 | This is the information retrieved from the database in 1.1.2.3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Query results | object | 1.1.3.1 | The information retrieved from the database, being output on to the screen |

Pseudocode:

<?php

Foreach (item in statement object)

Display item

?>

**Program specification 1.2.1 for 1.2 Add Staff Info into Staff Table**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Retrieve Submitted Form Data

Purpose: Retrieve the form data submitted by the user

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: A user has submitted a form and the info must be retrieved

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| $\_POST | Global Array | 1.2.1 | This array stores the information sent by a form |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Form variables | array | 1.2.1.1 | Form variables are saved into an array to be used to query the database |

Pseudocode:

For each variable:

$variable = $\_POST[“variable”]

**Program specification 1.2.2 for 1.1 Display New Staff Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Verify Submitted Form Data

Purpose: Verify the data in the submitted form

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: Form has been submitted

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| $\_POST | Global Array | 1.2.2 | This is the data submitted by the form using the post method |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| confirmation | html | 1.2.3 | Html page asking users to confirm the data to be entered into the database |

Pseudocode:

If $errors = 0

Display confirmation.html

**Program specification 1.2.3 for 1.1 Display New Staff Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Commit to Database

Purpose: Create SQL statement to submit information into database

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML, SQL

Events: A user has confirmed they want to enter information into the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| $\_POST | Global Array | 1.2.3 | This stores the information that is to be put into the database |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Statement object | object | 1.2.3.1 | SQL statement to add information into database |

Pseudocode:

Prepare SQL statement with elements from $\_POST array

**Program specification 2.1.3 for 2.1 Display New Client Form**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display New Client Form on Screen

Purpose: Display new client form on screen

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: Required information has been retrieved from the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Statement object | object | 2.1.3 | This is the information retrieved from the database in 1.1.2.3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| New Client Form | Html file | 2.1 | New client form, gets displayed on screen |

Pseudocode:

<?php

<html>

<new staff form>

</html>

?>

**Program specification 2.2.3 for 2.2 Add Client Info into Client Table**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Insert Submitted Form Data into Client Table

Purpose: Insert submitted form data into client table

Programmer: Justin Alho

Date due: 3/17/20

Languages:PHP, HTML, SQL

Events: A user has confirmed they want to enter information into the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| $\_POST | Global Array | 2.2.3 | This stores the information that is to be put into the database |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| confirmation | HTML file | 2.2 | Confirmation that info has been added into database |

Pseudocode:

Prepare SQL statement with elements from $\_POST array

Execute(SQL Statement)

**Program specification 3.1.2 for 3.1 Display Staff Selection Screen**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display Search Box for Staff Selection

Purpose: Display search box for staff selection

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: User has been confirmed, they need to select a staff member

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Request | request | 3.1.3 | The user requests to select a staff member |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Staff Selection Box | HTML input | 3.1 | The input to search for staff member |

Pseudocode:

<input id=”staffSearch”/>Search for staff

**Program specification 3.2.3 for 3.1 Display List of Matching Staff**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display List of Staff Matching Search

Purpose: Display list of staff matching search

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: User enters letters into the worker search box

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| search | string | 3.2.3 | The user’s entered search |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Matching staff | Array | 3.2 | An array of staff members matching the search |

Pseudocode:

onCharacterChange(showMatches())

**Program specification 3.3.3 for 3.1 Display List of Matching Staff**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display Information of Selected Staff Member

Purpose: Display information of selected staff member

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: User selects a staff member

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Statement Object | object | 3.3.3 | The information in the database about the selected staff |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Information | array | 3.3 | The information about the staff member, being displayed on the screen |

Pseudocode:

Foreach(info in Statement Object)

Display inf

**Program specification 3.4.3 for 3.1 Display List of Matching Staff**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Update Staff Record with Submitted Form Data

Purpose: Update staff record

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML, SQL

Events: A user has confirmed they want to enter information into the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| $\_POST | Global Array | 3.4.3 | This stores the information that is to be put into the database |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Confirmation | HTML file | 3.4 | Confirmation that info has been added into database |

Pseudocode:

Prepare SQL statement with elements from $\_POST array

Execute(SQL Statement)

**Program specification 4.1.3 for Display Client Selection Screen**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display Search Box for Client Selection.

Purpose: To display the client selection screen.

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: Required information has been retrieved from the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Statement object | object | 4.1 | This is the information retrieved from the database in 4.1.2.3. |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Statement Object | object | 4.1.3.1 | Object storing the requested data, being passed to the 4.1.3.1 process. |

Pseudocode:

<?php

<html>

<Show the client selection screen>

</html>

?>

**Program specification 5.2.3 for 5.2**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: insert submitted data into group home table

Programmer: Aidan

Date due: 17/03/20

Languages: php

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Form data | Post array |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Success message | Bool |  |  |

Pseudocode:

This will take in the form data and format it in to a sql statement to put it into the data base

Other:

**Program specification 6.1.2 for Display List of Group Homes**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display list of group homes.

Purpose: To display a list of group homes.

Programmer: Justin Alho

Date due: 3/17/20

Languages: PHP, HTML

Events: Required information has been retrieved from the database

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Statement object | object | 6.0 | This is the information retrieved from the database. |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Statement Object | object | 6.1.2 | Object storing the requested data, being passed to the 6.1.2 process. |

Pseudocode:

<?php

<html>

<Show the group homes>

</html>

?>

**Program specification 7.1: Display New Department Form for 7: Create Department**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display New Department

Purpose: Draws “New Department” form for purposes of designating information for a new department.

Programmer:

Date Due: 13/03/20

Languages:

* PHP
* HTML

Events: Receive request to access form.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| userAuth | bool | 7.1.1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Form | HTML, PHP | 7.2 |  |

Pseudocode:

If (userAuth($userID) == true){

If(uTypeAuth($userType) == true){

drawForm(cDept);

}

}

Other:

**Program specification 7.2: Add Department Info into Department Table for**

**7: Create Department**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Add Department Information into Database

Purpose: This module adds all information regarding the creation of new departments into a new record of the department table in the database.

Programmer:

Date Due: 13/03/2020

Languages:

* PHP
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Form Data | HTML POST Request | 7.2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Database Entry | SQL Statement | Database |  |

Pseudocode:

Other:

If (testConn($conn) == true){

$deptName = $\_POST[‘deptName’];

$deptCode = $\_POST[‘deptCode’];

$deptDesc = $\_POST[‘deptDesc’];

If (verifyData($deptName, $deptCode, $deptDesc) == true){

$sqlInsert = ‘insert $deptName, $deptCode, $deptDesc into dept\_name, dept\_code,

dept\_descripton;’;

insertData($sqlInsert);

}

}

**Program Specification 8.1: Display Department Selection Screen for**

**8: Update Department**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display Selection Screen

Purpose: This module will authenticate via library modules and display a list of all departments.

Programmer:

Date Due: 13/03/2020

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| userAuth | Bool | 8.1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Form | HTML | 8.2 |  |

Pseudocode:

If (userAuth($userID) == true){

If(uTypeAuth($userType) == true){

listAllDepts();

}

}

Other:

**Program Specification 8.2: Retrieve Department Information for**

**8: Update Department**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display Department Info

Purpose: This module populates a form containing the desired department information.

Date Due: 13/03/2020

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Database Info | SQL | 8.2 |  |
| Selected Record | HTML POST | 8.2 | From menu in 8.1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Form | HTML | 8.3 |  |

Pseudocode:

If (testConn($conn) == true){

$sqlRet = ‘select \* from department where dept\_code = ‘ . $\_POST[‘desireDept’] . ‘;’;

populateForm(uDept, $sqlRet);

}

Other:

**Program Specification 8.3: Update Department Record for**

**8: Update Department**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Update Record

Purpose: This module will handle the actual updating of entries.

Date Due: 13/03/2020

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Form Data | HTML POST | 8.3 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| SQL Statement | SQL | Database |  |

Pseudocode:

If (testConn($conn) == true){

$grabPOST($deptName, $deptCode, $deptDesc)

If (verifyData($deptName, $deptCode, $deptDesc) == true){

$sqlInsert = ‘insert $deptName, $deptCode, $deptDesc into dept\_name, dept\_code,

dept\_descripton;’;

insertData($sqlInsert);

}

}

Other:

**Program Specification 9.1: Query Database for 9: Schedule Shift (LIBRARY)**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Query Database

Purpose: This module simply queries the database for shift entries.

Programmer:

Date due: 13/03/2020

Languages:

* PHP
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Database Connection | SQL | 9.1, 9.2, 9.3 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Query Object | PHP, SQL | 9.2 |  |

Pseudocode:

tryConn(){

$conn = new [mysql connector]($EdenbridgeServer, $dbAccessAccount, $password)

If ($conn->connect\_error){

Die(“Connection failed”

}  
 }

prepStatement($statementType){ // For Process 9, this will be cShift for variable

switch{

case cWorker // Create Worker

{ Define statement for creating worker form}

Case uWorker // Update Worker

{ Define statement for updating worker form}

Case cShift // Create Shift

{ Define statement for creating shift form}

If ($\_POST[‘recShift’] == true){

{ Define statement for recurring shift entry}

}

Case uShift // Update Shifts

{ Define statement for updating shifts}

Case uRec // Update Recurring

{ Define statement for updating recurring shift form}

Case cDep // Create Department

{ Define statement for create department form}

Case uDep // Update Department

{ Define statement for update department form}

Case sDep // Select Departments

{ Define statement for select department form}

Case sWorker // Select Department

{ Define statement for select worker form}

Case sShift // Select Shift

{ Define statement for select shift form}

Case sRec // Select Recurring Shift

{ Define statement for select recurring shift form}

// This is repeated for rest of case statements

}

executeStatement($case, $\_variables\_$){

$sqlStatement = bind\_statement(“s”, $\_variables\_$);

$sqlStatement->execute();

}

**Program Specification 9.2: Make Shift for 9: Create Shift**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Make Shift

Purpose: Creates the statement to schedule a shift.

Programmer:

Date due: 13/03/20

Languages:

* PHP
* SQL
* HTML

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Query Object | PHP, SQL | 9.2 |  |
| Shift Details | SQL | 9.2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Prepared Statement | SQL, PHP | 9.3 |  |

Pseudocode:

$shiftStartTime = $row[‘shift\_sched\_start’]; // This is repeated for all variables returned by the form

// Insert handwavey process for retrieving information.

prepareStatement(cShift);

Other:

**Program Specification 9.3: Commit to Database for 9: Create Shift**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Commit Shift Info

Purpose: Commits the new shift to database.

Programmer:

Date due: 13/03/20

Languages:

* PHP
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Prepared Statement | PHP, SQL | 9.3 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| SQL Statement | SQL | Database |  |

Pseudocode:

executeStatement($prepStatement);

Other:

**Program specification 11.2 for 11.0**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: schedule recurring shift

Purpose: to schedule recurring shift

Programmer: Aidan

Date due: 25/02/20

Languages: php/html

Events:

This module will take a input from the worker, client and department table and will put the results in to a html form. After the form is filled out it the post array will be sent to 11.2.2 this will then return a prepared statement. Once the statement is returned this module will send it off to the main module.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Query object from 11.1.3 | Sql output | 11.2.1 |  |
| Shift details | Html post from form | 11.2.2 |  |
| Prepared statement | SQL input |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Shift specifications | Html post | 11.2.2 |  |
| Prepared statement | SQL input | 11.3 | Passes this to the commit to database module |

Pseudocode:

This module will take in the list of workers and clients that are active and populate a form to schedule recurring shifts. This will then take the filled-out form and send the post array to module 11.2.2. this will return a SQL statement to put the post array into database. This module will send this statement to 11.0

Other:

**Program specification 11.2.1 for 11.2**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: 11.2.1

Purpose: to display a html form and pass off

Programmer: Aidan

Date due: 17/03/20

Languages: php/html

Events:

Take in

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Shift specifications | SQL | 11.2.1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Shift details | Html post | 11.2.2 |  |

Pseudocode:

Take in worker, client, and department list and populate the respective field after the coordinator enters the data an submits the form it will send a post array to 11.2

Other:

**Program specification 11.2.2 for 11.2**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: 11.2.2

Purpose: format a SQL statement

Programmer: Aidan

Date due: 17/03/20

Languages: php

Events:

Receives html post and will process it

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Shift specifications | Html post | This |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Prepared statement | SQL statement | 11.3 |  |

Pseudocode:

Take a post array and put it in to a prepared statement to add recurring shifts to the data base

Other:

**Program specification 11.3 for 11.0**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: 11.3

Purpose: to take a prepared statement and send it to the SQL database

Programmer: Aidan

Date due: 17/03/20

Languages:php

Events:

Receives a prepared statement and puts it in to the data base

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Prepared statement | SQL input | this |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Success message | String | 11.0 |  |

Pseudocode:

This will take in data from the main module and submit it to the data basae

Other:

**Program specification 13.2 for 13.0**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: format search results

Purpose: format search results

Programmer: Aidan

Date due: /03/20

Languages:

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| List of shifts matching search | array | this |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Selected shift | Shift id |  |  |

Pseudocode:

This will format the shifts matching filters and send them to be displayed in the calendar api

It will then receive the selected shift and send that shift id

Other:

**Program specification 13.2.1 for 13.2**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: calendar api

Purpose: to show shifts in a format filmier with the coordinators

Programmer:

Date due: 17/03/20

Languages: pear api

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Shifts |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Links back with a shift id to edit | int | 13.3 |  |

Pseudocode:

Use pear to make a calendar

Other:

**Program specification 13.3 for 13.0**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: edit record

Purpose: to edit record

Programmer: Aidan

Date due: 17/03/20

Languages: php

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Shift data |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Form variables | Post array |  |  |

Pseudocode:

This will take in shift information and send it off to 13.3.1 it will return the edited form variables. This will then send them off to 13.3.2

Other:

**Program specification 13.3.1 for 13.3**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: display edit form

Purpose: display edit form

Programmer: Aidan

Date due: 17/03/20

Languages: php/html

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Shift information | array | this |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Form variables | Html post array | 13.3.2 |  |

Pseudocode:

This will take the shift information and auto fill a from. The coordinator will be able to change the values and submit it.

Other:

**Program specification 13.3.2 for 13.3**

Name: commit to data base

Purpose: commit to data base

Programmer: Aidan

Date due: 17/03/20

Languages: php

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Form variables | Post array | 13.3.2.1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Success message | bool |  |  |

Pseudocode:

This will take in the post array and send it off to 13.3.2.1 that will return a prepared statement and this module will send it off to 13.3.2.2

Other:

**Program specification 14.3 for 14.0**

Name: update recurring shifts

Purpose: update recurring shifts

Programmer: Aidan

Date due: 17/03/20

Languages: php

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Recurring shift information | database output |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Prepared statement | SQL input |  |  |

Pseudocode:

this will take in the selected shift and send it to 14.3.1 to display the form and let the coordinator update the shift.

Other:

**Program specification 14.4 for 14.0**

Name: commit to data base

Purpose: commit to data base

Programmer: Aidan

Date due: 17/03/20

Languages: php

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Prepared statement | SQL input |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Success message | bool |  |  |

Pseudocode:

Takes in a prepared statement and submits to database

Other:

**Program specification 15.3 for 15.0**

Name: commit to data base

Purpose: commit to data base

Programmer: Aidan

Date due: 17/03/20

Languages: php

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Prepared statement | SQL input |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Success message | bool |  |  |

Pseudocode:

Takes in a prepared statement and submits to database

Other:

**Program Specification : 16.1** **Display List of Submitted Timesheets**

**16: Approve hours**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display List of Submitted Timesheets

Purpose: It displays a list of submitted time sheets

Programmer: Evan Guest

Date due: 17/03/20

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Display Submitted Timesheets | string | 16.1.3 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Display List of Submitted Timesheets | string | 16.1 |  |

Pseudocode:

Will display a list of time sheets for the user to review utilizing SQL.

Other:

**Program Specification : 16.2** **Display Staff Timesheet**

**16: Approve hours**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display Staff Timesheet

Purpose: Shows the timesheet of the staff members.

Programmer: Evan Guest

Date due: 17/03/20

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| SQL Code | String | 16.2.2 | Takes SQL and outputs a timesheet |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Display Shift | string | 16.2 |  |

Pseudocode:

Takes SQL code and outputs a display of the time sheet.

Other:

**Program Specification : 16.3.1** **Retrieve Approved Hours from**

**16: Approve hours**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Retrieve Approved Hours from Timesheet

Programmer: Evan Guest

Date due: 17/03/20

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| $\_POST. | Global array | 16.3.1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Approved Hours | Array | 16.3.2 |  |

Pseudocode:

Foreach(hours in $\_POST)

Save hours into array

Other:

**Program Specification : 16.3.2** **Insert Approved Hours**

**16: Approve hours**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Insert Hours into Shift Table

Programmer: Evan Guest

Date due: 17/03/20

Languages:

* PHP
* HTML
* SQL

Events: Hours are approved

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Approved Hours | array | 16.3.2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Confirmation | HTML | 16.3 | Confirmation that hours have been submitted |

Pseudocode:

Foreach(hours in Approved Hours)

Insert into Shift hours as approved hours

Other:

**Program Specification : 16.3.** **Retrieve Approved Hours from**

**16: Approve hours**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Change Shifts to Approved

Programmer: Evan Guest

Date due: 17/03/20

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| none |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Confirmation | HTML | 16.3 | Confirmation that shifts have been changed |

Pseudocode:

Change claimed shifts to approved

Other:

**Program Specification : 17.1 Display Report Selection Screen**

**17: View Reports**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display Report Selection Screen

Purpose: Displays reports

Programmer: Evan Guest

Date due: 17/03/20

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| options | Long text | 17.1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| report | Long text | 17.1.3 |  |

Pseudocode:

Gives a list of option to allow the system to output a report to view

Other:

**Program Specification : 17.2.3 Generate Requested Report**

**17: View Reports**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Generate Requested Report

Purpose: Creates reports depending on information entered

Programmer: Evan Guest

Date due: 17/03/20

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Export PDF | Long text | 17.2.3 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| report | Blob | 17.2.3 |  |

Pseudocode:

FPDF will be used to generate PDFs from the reports.

**Program Specification : 17.2.2 Display requested report**

**17: View Reports**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display Requested Report

Purpose: Displays reports

Programmer: Evan Guest

Date due: 17/03/20

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Information | Long text | 1.1.4 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| report | Long text | 17.2.2 |  |

Pseudocode:

Uses Gathered information to display the report.

Other:

**Program Specification : 17.2.2.1 Sort Data by User Criteria**

**17: View Reports**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Sort Data by User Criteria

Purpose: Sorts user data to make a better report

Programmer: Evan Guest

Date due: 17/03/20

Languages:

* PHP
* HTML
* SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Report data | Long text | 17.2.2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Sorted Report | Long text | 17.2.2.1 |  |

Pseudocode:

Uses user specifications to sort by the user preference.

Other:

**Program Specification 18.1: Rebuild database with SQL dump file**

**18: Recovery**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Rebuild database With SQL Dump File

Purpose: This module is designed to rebuild the SQL database using backups

Programmer: Evan Guest

Date due: 17/03/20

Languages:

PHP

HTML

SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Request SQL | Request | 18.1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| SQL File | Blob | 18.2 |  |

Pseudocode:

The data bases are connected to the backups using the given credentials after the connection to the data base is established, the system can get the required SQL dump from the desired location. The restoration process requires confirmation before its run ensuring the user is of the proper permissions.

Other:

**Program Specification 18.2.2: Store Rebuilt Database in Webhost's Filesystem**

**18: Recovery**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Store Rebuild Database in Webhost's File System

Purpose: Takes the rebuilt SQL Database and uploads it to A2Hosting

Programmer: Evan Guest

Date due: 17/03/20

Languages:

PHP

HTML

SQL

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| SQL File | Blob | 18.2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| SQL File | Blob | 18.2 |  |

Pseudocode:

Takes the file from process 18.1 and using user credentials for A2Hosting connects and uploads the rebuilt file for use.

Other:

**Program specification 19.1 for Log in**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Display landing page.

Purpose: To display a different landing page based on user type.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP HTML

Events: This module will take in the type of user that has logged in and display a different page for each user type.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| User type variable. | String. | Display landing page. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
|  |  |  |  |

Pseudocode:

Switch(userType)

{

Case “typeOfUser”:

Printf(Specific HTML);

Break;

Case “OtherUserType”:

Printf(This users HTML);

Break;

Case “”:

Printf(Other HTML);

Break;

}

Other:

**Program specification 19.1.1 for Log in**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Connect to database.

Purpose: To connect to the database.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP

Events: This module will connect to the MySQL database.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Credentials. | String. | Connect to database. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Database connection. | Database connection object. | Any module that needs a connection to the database. |  |

Pseudocode:

Connection = databaseConnectionType(Credential arguments);

Other:

**Program specification 19.1.2 for Log in**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Validate credentials

Purpose: To validate the credentials of the user.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP SQL HTML

Events: This module will get the information from the database using the credentials submitted in the log in form. It will then return the hashed password for the specific user to compare.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Credentials. | Strings. | Get required information from database. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| hashedPassword. | String. | Compare submitted form data with information in database. |  |

Pseudocode:

PreparedStatement = databaseConnection->(SQL goes here with placeholder variables);

Execute(PreparedStatement);

Other:

**Program specification 19.1.3 for Log in**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Verify password.

Purpose: To check if the user’s credentials are valid.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP HTML

Events: This module will use the hashed password as an argument for the password\_verify method of PHP to determine whether the password is accurate.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Hashed password. | String. | Verify password. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Legitimate password bool. | Boolean. | Compare submitted form data with information in database. |  |

Pseudocode:

Password\_verify(hashedPassword);

Other:

**Program specification 19.1.4 for Log in**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Compare submitted form data with information in database.

Purpose: To determine user type if the user is valid.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP HTML

Events: This module will send back the type of user that has signed in if the user was determined to be valid.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Legitimate password bool. | Boolean. | Determine user type. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| User type variable. | String. | Display landing page. |  |

Pseudocode:

If(legitPass)

{

Session[‘userType’] = user type field for the user select statement executed earlier;

}

Other:

**Program specification 20.1.1 for Backup database**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Scan backup directory.

Purpose: To put the file names into an array of strings.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP

Events: The module will create an array of the file names.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Data type | Used by | Notes |
| Backup directory path name. | String. | 1.1.1 Scan backup directory. | The name of the path where the backups are stored. |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Data type | Used by | Notes |
| Array of backup files. | Array. | 1.1 Manage directory contents. | An array containing all the names of the files contained within the backup directory. |

Pseudocode:

Array = Scandir(directory, sort order);

Other:

**Program specification 20.1.2 for Backup database**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Sort backup directory array.

Purpose: To sort the array of file names by number.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP

Events: The module will sort the array of file names in numeric order.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Array of file names. | Array. | 1.1.2 Manage directory contents. | This array will contain all the file names of the backup directory. |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Sorted array of filenames. | Array. | 1.1 Manage directory contents. | This array will be sorted in numeric order making it easy to refer to the most recent backup file. |

Pseudocode:

sort(array, sort\_numeric flag);

Other:

**Program specification 20.2.1 for Backup database**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Get last file name in the array.

Purpose: To get the most recent backup files name.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP

Events: This module will refer to the first index of the array which should be the most recent backup file after being sorted.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Sorted array of backup directory files. | Array. | 1.2.1 Get last file name in the array. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Newest backup files name. | String. | 1.2. |  |

Pseudocode:

$string = $array[0];

Other:

**Program specification 20.2.2 for Backup database**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: 1.2.2 Split string into array.

Purpose: To split the file name into an array of strings representing each of the characters of the string.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP

Events: The module will take the file name in and output an array of strings one character in length.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Newest backup files name. | String. | 1.2.2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Array of the strings characters. | Array | 1.2 |  |

Pseudocode: array = str\_split(string);

Other:

**Program specification 20.2.3 for Backup database**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Find number of dump file.

Purpose: To find the number of the backup file.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP

Events: This module will iterate through the array checking each value to see if it is numeric.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Array of the file name strings characters. | String. | 1.2.3. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Backup file number. | String. | 1.2.4. |  |

Pseudocode:

String = “”;

foreach(array as ph)

{

If( is\_numeric(ph) )

{

string = string + ph;

}

}

Other:

**Program specification 20.2.4 for Backup database**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Increment number.

Purpose: To increment the number of the backup file to ensure a unique file creation.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP

Events: The module will take the file number string, cast it to integer, and increment that integer.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Backup file number. | String. | 1.2.4. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| New backup file number. | Int. | 1.3. |  |

Pseudocode:

fileNumber = (int) string;

fileNumber = fileNumber + 1;

Other:

**Program specification 20.3 for Backup database**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Execute MySQL command.

Purpose: To execute the command for the database to be backed up.

Programmer: Harley Lenton

Date due: DD/MM/20

Languages: PHP Shell command

Events:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| Command to execute a “mysqldump”. | String. | 1.3. | This string will contain the name of the host, credentials for signing in, backup directory path concatenating the backup number to the end of the file name. |
| Backup file number. | Int. | 1.3. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Shell output. | Array. | 1.4. |  |
| Succeeded. | Boolean. | 1.4. |  |

Pseudocode:

succeeded = false;

fileNumber = (string) fileNumber;

If( exec(“mysqldump command /backupDir/backupFile” + “” + “.sql”, array output[] ) && (output != null ) )

{

succeeded = true;

}

Else

{

Succeeded = false;

}

Other:

**Program specification 20.4 for Backup database**

Module\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Log failure.

Purpose: Adds and entry to a log file if the backup was not successful.

Programmer: Harley Lenton.

Date due: DD/MM/20

Languages: PHP

Events: This module will take a bool check if it is true or false and

|  |  |  |  |
| --- | --- | --- | --- |
| Input Name | Type | Used by | Notes |
| succeeded | Bool | 1.4 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Output Name | Type | Used by | Notes |
| Log entry | Text in a file. | 1 |  |

Pseudocode:

If(succeeded == false)

{

dateString = date(formatString);

//This string will be “\nBackup unsuccessful” + “dateString” //append

file\_put\_contents(filePath, entryString, FILE\_APPEND | LOCK\_EX);

}

Other:

# User Interface

## Login

A screenshot of a cell phone

Description automatically generated

This is the login page for the system. Users will enter their username and password and then depending on what type of user they are, such as staff, coordinator, or bookkeeper, they will have a completely different view of the system.

# Worker Interface

A picture containing screenshot

Description automatically generated

This interface will be visible to the workers and will be relatively simple. It will allow workers to view their currently scheduled shifts, as well as review the shifts they have completed and submit the hours they have worked.

## View Schedule

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

The first main component of the staff interface is the ability for a staff member to view their upcoming shifts. The staff member’s shifts will be shown in a calendar view, where it provides staff with a look at every shift with their staff ID a month at a time. They will be also be able to view the shifts for the next month or for previous months by clicking the arrows to the left and right of the month’s name. If they want to view a shift’s details, they can click on a shift in the calendar to bring up a screen with the details for that shift.

## View Details

A screenshot of a cell phone

Description automatically generated

After a worker selects a shift, they will see this screen. Contained in it is the full details of a selected shift. Workers will be unable to modify any information in this screen, though should a worker work hours that differ from what was scheduled, they can either contact the coordinator to have them modify or claim hours different from what was scheduled on the timesheet screen.

## Submit Hours

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

At Edenbridge, staff members are required to submit their hours twice per month: on the 15th and the 30th or the date otherwise defined as the end of the month. When a staff member goes to submit hours in our system, they will be presented with this form. The pay period that is displayed with this form will be the most recent pay period that has unclaimed shifts and no scheduled shifts. The form will be automatically filled with information on the shifts the worker has worked including details regarding the clients of which shifts are for, start times, and end times for each shift. This screen is organized by the clients, to provide some additional clarity in going through the form. Should there be no unsubmitted hours in the specified period, a prompt will appear stating as such.

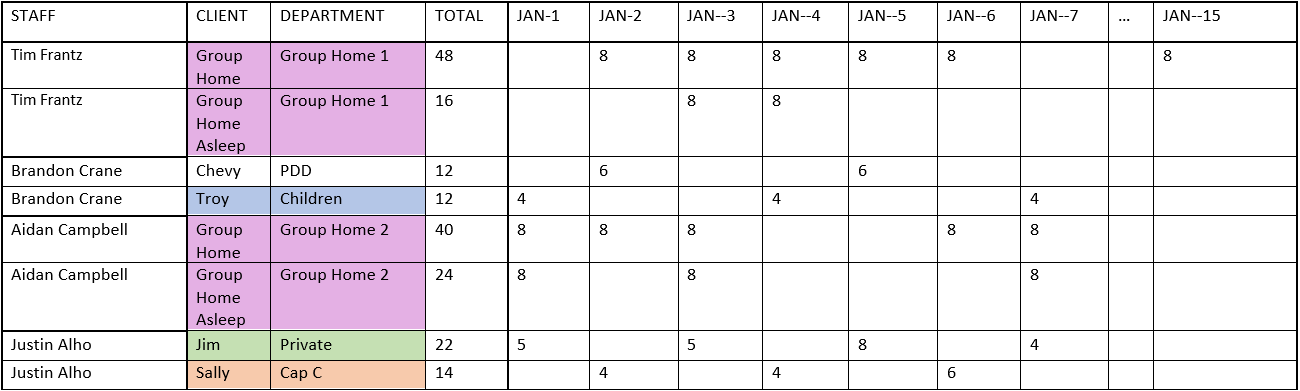
# Bookkeeper Interface

A screenshot of a cell phone

Description automatically generated

The dashboard for bookkeepers is simple, with the only major option being to view reports. Through that option, bookkeepers will be able to view and generate several different reports. Additionally, they will be able to do searches for information regarding clients and/or staff.

## View Reports

A screenshot of a cell phone

Description automatically generated

This interface will primarily be used by bookkeepers, although coordinators will have access to the ability to view reports as well. Through this screen, sorting and filtering options are available for users to be able to tailor a report to their needs. Additionally, options to include fields used in exception reports will be available. Once the report is generated, users will have access to the ability to export the report to PDF allowing them to either print the reports or to email them as necessary.

## Search

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

The search feature will allow users to view information regarding staff members or clients. While bookkeepers will have the ability to view the information presented in this interface, they will be unable to edit. Coordinators on the other hand will possess the ability to edit all information presented.

# Coordinator Interface

A screenshot of a cell phone

Description automatically generated

The interface for coordinators is moderately extensive, allowing users with requisite privileges to manage a variety of aspects. This includes database information such as the addition and modification of clients, staff members, departments, and group homes. The interface also allows them to search information regarding staff members and clients akin to the way that bookkeepers can. A primary feature of this interface however is the ability to access an array of menus regarding shifts including the scheduling and modification of shifts, recurring shifts, and verifying hours submitted by staff members.

## Add Staff Member

A screenshot of a cell phone

Description automatically generated

The main purpose of this interface is to allow users with the requisite coordinator permissions to add new workers. This interface follows the layout and formatting of the related form. Information submitted through this form will be inserted as a new entry in the worker table of the database.

## Edit Staff Member

A screenshot of a cell phone

Description automatically generated

This interface allows coordinators to edit details of a staff member. When a coordinator selects a staff member from any of the interfaces where this is an option, that staff member’s information is polled from the database and is used to automatically fill out the form. All information in this interface can be edited, except for the ID which must be kept the same. Once the submit button is clicked, the related record as identified by the ID is modified and changes are committed with the new details.

## Add Client

A screenshot of a cell phone

Description automatically generated

This form allows coordinators to add information for new clients. After the submit button is pressed, the information contained in all form fields is processed and inserted into a new record in the client table of the database.

## Edit Client

A screenshot of a cell phone

Description automatically generated

Editing clients is done through this interface. Through it, client information is pulled into the form from the database and automatically fills it out with the specified information. All information in this interface save for the ID can be modified. Once the submit button is pressed, the information including any changes is processed and overwrites the specified entry in the database.

## Add Group Home

A screenshot of a cell phone

Description automatically generated

This interface will allow coordinators to add information corresponding to a new group home. A major facet of this interface is a confirmation prompt that alerts the coordinator that it will modify two separate tables: group home and department. If they confirm, entries will be added into both tables with the information in the fields as attributes in both.

## Edit Group Home

A screenshot of a cell phone

Description automatically generated

The “Edit Group Home” interface will allow coordinators to edit details pertaining to a group home. After the desired group home is selected, related information polled from the database and used to pre-fill the form fields. The formatting of this interface is near exactly like that of the interface used for adding new group homes with the addition of the ID. All elements except for the group home ID are editable. Extra precautions are taken with editing group home information will change fields in multiple tables, including the department name used to identify the group home as a department.

## Add Department

A screenshot of a cell phone

Description automatically generated

This form will allow coordinators to add required information for a new department. The layout in this interface mimics that of adding a new group home. This includes an extra confirmation screen, with information regarding how submitting the form updates multiple tables.

## Edit Department

A screenshot of a cell phone

Description automatically generated

The interface for editing departments is restricted to coordinators. In this interface, the ability is granted to edit all information related to a department. The information for editing departments is pulled from the relevant department record, with it being determined by whichever entry is selected by the coordinator in a preceding menu. As is a standard with menus allowing editing information, the department code cannot be changed due to data integrity concerns. As with other department and group home interfaces, a confirmation prompt will appear upon hitting submit with cautionary information regarding how it changes multiple tables.

## Schedule Shift

A screenshot of a cell phone

Description automatically generated

This form allows coordinators to schedule new shifts. Both the “Search for Staff” and “Search for Client” fields when used will cause their related result field dropdowns to become populated with matching results. When empty, both fields will be populated with all options marked active. Staff have additional filtering in place to ensure that they are available for the day/time of a shift. Selecting the checkbox for recurring shifts will cause the shift to be designated as a recurring shift, scheduling the shift on the initial day specified and adding a record to the recurring shift table. The day for the recurring shift entry is replaced by the day of the week and used to schedule all following shifts following that record.

## Edit Shift

A screenshot of a cell phone

Description automatically generated

The edit shift interface will most often be used in instances where a worker is unable to work a specific shift or ends up working less than initially expected. As is standard for edit interfaces, fields will be pre-filled with the associated entry in the database.

## Edit Recurring Shifts

A screenshot of a cell phone

Description automatically generated

The interface to edit recurring shifts will be locked to coordinators. Fields in this interface are automatically filled out with the corresponding information from the recurring shift table. Simply updating the recurring shift entry will not update any shifts already scheduled from a recurring shift template.

## Schedule Recurring Shifts

A screenshot of a cell phone

Description automatically generated

This screen will allow coordinators to schedule every recurring shift by department, including recurring group home shifts as each group home has an associated department. Coordinators simply pick a start date and an end date, then click “Schedule Shifts” to generate each individual shift associated with that recurring shift within the two dates specified.

## Approve Hours

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

When a coordinator chooses to approve submitted hours, they will see a screen like this showing a list of staff members with the hours they have submitted. Each staff member listing will also be either coloured green if they have no discrepancies between their scheduled and claimed hours, or red if there are discrepancies. Coordinators have the option to instantly approve the hours submitted by each staff member, since if there are no discrepancies between their scheduled and claimed hours, there may not be any reason to review them in greater detail. However, if the hours they have claimed do not match what was scheduled, coordinators have the option to review their timesheet in greater detail.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

If the coordinators choose to review a staff member’s submitted hours, they will see a screen like this, with a copy of the staff member’s timesheet with any discrepancies highlighted in red. From there, coordinators can either modify the submitted hours, or approve the hours the staff member has submitted. When the hours are approved, the start and end times currently in the timesheet are added to the corresponding shift records as approved start and end times and the status of the shift is changed to approved.

## Search

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Like the bookkeepers, coordinators will be able to search for client and staff information. However, unlike bookkeepers, coordinators will have an option to edit the information for the client or staff member that they search for. By choosing to edit their information, the coordinator is taken to the edit staff or edit client screens.

## Logout

A screenshot of a cell phone

Description automatically generated

This is the logout page. When a user logs out, they will see this page briefly, then be redirected to the login page. If they are not redirected, there is also a link that will redirect them back to the login page.

# Additional Process

## Backup

In order to reduce the risk of data loss and ensure data can be restored easily, system backups are a necessity. For organizations like Edenbridge where new information is being created and accessed every day, regular backups are especially vital.

For our system, backups will be done regularly with a backup process being executed once per day locally. This will be done automatically by utilizing the “cron” utility in Linux, which is a utility that allows jobs to be run at certain times or intervals, to schedule a “mysqldump” event once every day at a time with low server activity. This will “dump” the contents of the database by saving a file with a set of MySQL statements that can be used to rebuild the database if anything happens to it. We will also be taking advantage of Altis' current backup plan with Edenbridge in which they back up the data from Edenbridge’s server twice per day. We can take advantage of this by sending the backups to the server located in their office with SFTP. In addition to local backups, A2 Hosting, which is Edenbridge’s web hosting service company, performs regular off-site backups to ensure that the system itself is not compromised. As a last resort, we will also provide Edenbridge with copies of all the source code for the system so that even if a catastrophic event were to occur, the system could be restored to the condition it was in when it was first implemented.

## Restoration

The data restoration process would be executed in the event irreparable damage has been done to the structure of the database, the data contained therein, or in the event of data loss by the host. The data that would be restored in such an instance is that of the latest verified backup.

The process of restoration would involve using the host’s backup if it is available. Should this is not an option, we will use the file created by the “mysqldump” from one of our backups. This file can be run on the MySQL server to restore the database to the same state it was in when the data was dumped.

## Help

Since this system is new to our client, they will likely need help figuring out how to use it effectively.

# Physical & Technical Architecture

## General Overview of Use

A screenshot of a social media post

Description automatically generated

Above is a diagram detailing the use of the system. This includes an organization of where any user might be located, the classification of the user, and the device they might use at that location. As well, it provides a general idea of the flow of data within the system.

## Manual Processes

There is a selection of processes along the human/machine boundary in our system that will remain manual. Many of these processes will trigger other automatic processes that will process inputs or manage the outputs requested.

### Timesheet Submission

This process must be triggered by an employee. Although the database contains all the scheduled information, we require employees’ input to verify the hours worked. This is primarily a verification step for any given shift that is worked.

### Shift Creation

Process is triggered by coordinators. This process is only partially automatable at present. This process will be manually triggered for a variety of reasons, the foremost being that there needs to be some oversight on the shifts being created, including recurring shifts.

### Shift Rescheduling

This process is started by either workers or coordinators. In the ideal scenario, we will configure this process to populate a list with a selection of quick possibilities, making this step as efficient as possible. Both coordinators are required in this process to give the clear for any rescheduling and the workers are needed to confirm that they can work the rescheduled shift.

### Shift Verification

Many aspects with this process will be automated, but a validation will be requested should a discrepancy outside a tolerable range arise. Such a discrepancy would arise if the hours as scheduled and reported by the worker are different enough. While a difference of 15 minutes either way from a scheduled end time would not trigger any alerts, 3 hours past a scheduled end time would.

### User Creation & Management

This process is more a matter of practicality and security. By restricting this process to system administrators, numerous avenues for individuals to get unauthorized access to the system are blocked. The benefit to the security of the system is well worth the cost to the ease of onboarding new workers.

### Reporting Processes

While our implementation with this process can ensure that reports and requisite files are properly prepared, the initiation of this process should be left to manual control. Simply exporting the files could create problems later with storage as well as reports may not cover the specific range of dates or other criteria that is required of a report.

### Restoration Process

The former half of the backup & restoration process will be fully automated. The same is not true of the restoration process. This process will be initiated by only system admins if components of the system or database are damaged. This process is explicitly manual for some key reasons. The primary one is that should a false flag be set for an error, it would result in an unwanted restoration. Secondarily, an automated system would not be able to easily verify that backups are properly working. If it were automatic and to restore a corrupted backup, it could potentially wipe the entire system.

## Requisite Software

### MySQL Database

MySQL Database represent the essential core of our system. We will be utilizing the community edition as the backbone of the other components and what all the information is drawn from. Through this we are building all our tables and relationships for the data, as well as providing points on which other portions can be built from.

### Apache

Apache represents the framework upon which everything stems out from. Apache provides the hosting capability and is managed by A2Hosting. The version A2Hosting utilizes runs on Linux which extends to the system some extra capability via numerous other extensions that would otherwise not operate properly.

### PHP – PHP: Hypertext Pre-Processor

PHP effectively makes up the second half of our core infrastructure with this system. With our pages being done primarily through the PHP scripting language, the PHP infrastructure and libraries extended through it become vital. Most of the functionality with our system as well as the interface itself will be made with incorporating PHP. PHP is available as a free download and if it is not already incorporated into the hosting solution including Apache, it is a simple inclusion.

### FPDF – Free PDF PHP Libraries

The libraries that are contained within FPDF’s system provide the portion of what we will be using for the reports in our system. Available as a package to import into our system, it extends the segments to our system that will generate PDF files that will consist of reports and other formatted files required as outputs. This is a simple inclusion and can be packaged with the system.

### PEAR – PHP Libraries

In the same vein as FPDF, PEAR’s collection of libraries extends to our system vital functionality and time saving measures. Specifically, we will be utilizing PEAR’s calendar and date functionality to have graphical representations of shifts allocated on a monthly basis.

## Data Distribution

The system we are developing will primarily have data in the system be centralized. The primary location for the database will be at Edenbridge’s physical location. The website will be managed at A2Hosting’s location, with some backups of the database be managed by them. Finally, general backups of the system will be managed by Altis. The database being located with Edenbridge ensures some level of security, as only those with access already should be able to directly manipulate the data within. The website hosting solution with A2Hosting was one already in place at Edenbridge and thus one we must comply with. The distributed backups are the only true level of distribution of data, which is preferable as multiple off-site backups for different sections of the system gives us a level of reliability that we would not have otherwise.

## Client-Server Architecture

The system we are developing incorporates client-server architecture heavily. With our system, the entirety of the interface, while stored on the hosting server for the interface, is handled by the client computer or device. In effect, we have a three-tier client-server architecture in place. Client devices will handle presentation logic, the server at A2Hosting will handle application logic, and the server at Edenbridge will handle data access and data storage logic.

## Required Hardware

At present time, we have established that all required hardware components are in place. The hosting servers are already in place, with the primary one that will handle operations like backups and hosting the site managed by A2Hosting and the secondary one at Edenbridge prepared. Other components like storage have their needs met in a related manner. At this point in time, we have no immediate requirements but potentially some optional recommendations for a later point in time involving either NAS or RAID storage to further ensure services are maintained.

# Lessons Learned

| Date | Submitted By | Milestone | Experience | Lesson | Lesson Type | Effect | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Feb. 15, 2020 | Aidan | Milestone 7 | Spending too much time figuring out how certain modules work and not moving on to modules we can make | Get the stuff you know how to build and deal with a problem when the other stuff is done | Technical | We will be able to get more work done up front and work on the problems when we have leftover time | |
| Feb. 15, 2020 | Justin | Milestone 7 | We split up the work and assigned tasks to everyone | By dividing up the workload and giving everyone a specific task, more work can get done and there is less risk of people being without something to do. | Technical | From now on, we will assign specific jobs to everyone to ensure everyone has a role to play and nobody is without something to work on. | |
| Feb 15, 2020 | Harley | Milestone 7 | Made structure charts. | Everyone has a different way of making structure charts which makes it difficult to switch from chart to chart. | Technical | In the future I will be more inclined to have one dedicated structure chart person. | |
| Feb 14, 2020 | Beryon | Milestone 7 | 1) Had to play the role of HR this milestone.  2) Got annoyed being told last minute that people need drastic help with their sections. 10 hours is not enough to get in-depth for both structure diagrams and program specifications. | I know better now how to approach some intra-team issues. Still feel unsure about the results but it’s workable. On account of the second point, I need to make myself more clear about boundaries. | Personal | Should be able to better adjust to issues in the team like we had. Also will be much more explicit about the acceptable boundaries. I am fine with adjusting my workload, but I need time to figure out what I’m doing. | |
| Feb. 8, 2020 | Justin | Milestone 7 | Database had to be modified quite a bit after a meeting at Edenbridge and a close look at some financial forms. | Review the information we have more thoroughly before coming up with our own ideas of how things work. | Technical | In the future, we will review all the information we have to ensure our ideas will actually work with the way the system is implemented. | |
| Jan. 25, 2020 | Beryon | Milestone 6 | To be slightly off topic, very much dislike being sick like I have been for the last two weeks. More on topic, found the importance of ensuring that we have everything we need to know in order, like with archiving. | I now know the importance of keeping everything in order, and to ensure that we remain knowledgeable about the tools we intend to use. On the cold front, to be a little more proactive in damage control. | Personal/Academic | Going forwards, planning on looking a little more critically at what we are planning and evaluating if specific aspects absolutely must be implemented. Should help guard against feature creep. Also, buying hand sanitizer. | |
| Jan. 24, 2020 | Justin | Milestone 6 | After spending a lot of time on devising an archival plan, we realized it is not necessary | Consider whether or not something is necessary before deciding to implement it, especially if it would require a lot of extra work. | Academic | From now on, we will determine if any potential additions to our system are actually necessary before planning how to implement them. | |
| Jan. 17, 2020 | Harley | Milestone 6 | Information I relayed to the team from external sources was misinterpreted. | I must be as clear as possible leaving absolutely no ambiguity in my messages.  Otherwise, it is likely that occasionally, people will  not understand what I am saying. | Personal | I will attempt to communicate in a way that does not allow for alternate interpretation. Clarifying all details of what I am saying should help. | |
| Dec. 2, 2019 | Justin | Milestone 5 | By getting a head start on some projects, I was able to finish them early, giving me more time to focus on more important things, like the Systems assignments. | Get a head start on assignments, do not start working on things the day before they are due. | Academic | From now on, I will try harder to start working on things sooner, as I enjoy not having to stress about finishing assignments a few hours before they are due (except when milestone documents are due apparently). | |
| Dec. 2, 2019 | Beryon | Milestone 5 | Overly broad? Maybe. End of this milestone, I feel like I have a shockingly concise view about how this system will look at the end of it. The little technical details are coming together in my head enough to visualize it. | The entire waterfall process we’ve gone through so far has been especially helpful to be able to grasp how this system will come together. | Personal | Going into the second half of this project I feel far more confident about it. Enough so that I am starting to get a feel for how individual components would come together. This means that I can start building a framework to be able to run off of once we begin the related sections. | |
| Dec. 2, 2019 | Harley | Milestone 5 | Overestimated the cost of the project | Professionals would be significantly more experienced, which would allow them to achieve the same thing we are in a third of the time or less. | Personal | I will spend more of my spare time working on programming projects to gain experience, and make it second nature for myself to complete the tasks involved in development. |
| Nov. 29, 2019 | Aidan | Milestone 5 | Wearing myself out with work and not letting myself have some time to sit down and rest | Next semester I will allow for an hour between school and work | Personal | This will give me some time to sit and rest between the two major things I need to do in a day | |
| Nov. 26,  2019 | Evan | Milestone  5 | Learning how project design works | I have no interest in making systems for a living | Personal | When I pursue a job in I.T. I will not choose one involving system design for outside sources. | |
| Nov. 18, 2019 | Evan | Milestone 4 | Having diminished cognitive activity due to 8am classes | I now know what it feels like to have Alzheimer's, it is an ungodly fate to anyone who develops it. | Personal | Will plan my next semester with 8am classes on alternating days. | |
| Nov. 18, 2019 | Beryon | Milestone 4 | Flash of inspiration coming from a different class, an idea to use Windows credentials to act as the login for the web portal. | Upon further reading, it appears possible to utilize existing Windows login credentials for logging into the site. | Technical | This is entirely subject to the existing structure and how the site is currently set up, as this segment of the system would depend heavily on the utilization of IIS. If it is utilized for the site, then we have our login system. Again, pending some questions though. | |
| Nov. 17, 2019 | Harley | Milestone 4 | I became acquainted with SQLyog. | SQLyog is a powerful GUI based tool for interfacing with MySQL. Previously I wrote scripts for everything I did with MySQL databases. | Technical | I should be able to create and modify databases a bit faster now that I have SQLyog. | |
| Nov. 14, 2019 | Aidan | Milestone 4 | I learned more about how php functions | Php does not remember where you are when it sends back the web page so you have to program it to save values that you want to manipulate later | Technical | This will help me to make the web app function and remember where the user is along the process | |
| Nov. 14, 2019 | Justin | Milestone 4 | I learned the basics of integrating databases into PHP, which will be essential for our app when it is developed. | Now that I know how to actually make everything connect, creating the web app actually seems like something that can be done. | Technical | Now that we know how to create the web app, we have an idea of how we can get it to work. | |
| Oct. 26, 2019 | Beryon | Milestone 3 | This may be similar to an earlier one I did but found out the designer aspects within the Office Suite is a fair deal more powerful than I initially expected, even with 5+ years of experience using it for educational things. | Design and layout aspects within Microsoft Office Suite has far more functionality than it initially looks like. | Technical | With this Milestone as can be observed, there’s a good deal more effort that I put into the presentation of the document and the layout such as with the table of contents which is almost automated with using the styles bar on the home tab. This knowledge will be carried forwards into not just other documents, but all files created during this project. | |
| Oct. 26, 2019 | Evan | Milestone 3 | Visio’s sharing policies are not as optimized as the rest of the office suite. At multiple times what was the newest version was unclear. | Don’t rely on Microsoft office to share and maintain files across users. | Personal | If Visio or a program similar is used in the future, a repository type program needs to be used. | |
| Oct. 24, 2019 | Aidan | Milestone 3 | Not knowing enough about the scheduling process and how it is going to function | Focusing too much on the extra details of the system and not asking questions about some essential parts of the scheduling system | Academic | I will try to look at the big picture more often to check if I am only looking at one part | |
| Oct. 24,  2019 | Harley | Milestone 3 | Received a poor grade on Milestone 2 | Proofreading everything extensively is very important to ensure the documentation is up to the standard expected of us. | Academic | I will now be aiding in the proofreading and revision going over the deliverables multiple times to ensure what we produce is high quality as much as I’m able to. | |
| Oct. 18, 2019 | Beryon | Milestone 3 | Lesson? Reminder? Whichever, finally sunk in that this is an actual system being developed. | Far more detail than what I was accustomed to doing with classwork is required for this systems project. | Personal | Going forwards, will be redoubling efforts to ensure every little detail with what we do to ensure that it not only fulfills the deliverable requirements, but also to whatever specification the client provides on an aspect. | |
| Oct. 17, 2019 | Justin | Milestone 3 | We started focusing too heavily on aspects of the project that were not very important | Focus on important things first, then think about extras later | Academic | From now on, we will focus on the functionality of the system, anything extra will only be considered after the basic requirements are met | |
| Oct. 11, 2019 | Beryon | Milestone 2 | Project has a substantial feature-set, most of which was previously unknown from the first couple times. | Learned a good amount about how to work Project, including different ways to set up tasks and organize them. | Technical | Going forwards, should be far easier to arrange events and plan things out via Project. As a side effect, there’s a measure of knowledge attained in how to potentially display events in the system being developed. | |
| Oct. 11, 2019 | Justin | Milestone 2 | Milestone presentation was not great, no intro or conclusion | Make sure assignments include all components | Academic | We will have a team member observe presentations from the other class to know what needs to be included for future presentations | |
| Oct 11,  2019 | Harley | Milestone 2 | Organization  of presentation  was not ideal. | The order in which the information is presented is very important if we want it to be easy to follow. | Academic. | Someone will be assigned the duty of analyzing the presentation to ensure the flow of information makes more sense. | |
| Oct 11,  2019 | Aidan | Milestone 2 | My work school balance was not great for the first month of school | I learned how far I can push myself before my school begins to suffer from it | Personal | Going forward I am not taking as much hours at work and am going to say no to more hours when asked so I can focus on this project | |