Testing Plan

**Taylor’s Professional Services**

Team F

* 1. Purpose

This document describes the test plan for the client, staff, and manager-based Taylor’s Professional Services System. This Test Plan document supports the following objectives:

* Create a Unit Test policy
* Outline unit testing procedures and responsibilities
* Give unit test case results
* Create an Integration Test policy
* Define an integration strategy
* Outline integration and test procedures
* Provide Integration Testing results
  1. Scope

This Test Plan describes the integration and system tests that will be conducted on the TPS System prototype. Unit tests and integration will be tested through the following areas:

* Access Control
* Database Management
* Interface

2.1 Unit Testing

A. Unit Test Policy

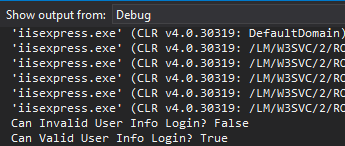
* Unit testing on a development ownership will be conducted to detect defective code in units and reduce risk of unit failure in production.

B. Unit Testing Procedures and Responsibilities

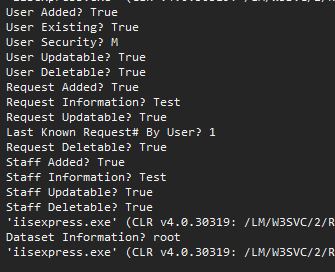
* + All programmers will be responsible in conducting unit tests of their own code.
  + Unit code will be created for every method and class instance, reporting results to the team leader.
  + Result cases are to be sent to the team leader.

C. Unit Test Case Results (From Visual Studio Output)

Access Control Results –



Database Management Results -



3.1 Integration Testing

A. Integration Testing Policy

* Integration testing on a development ownership will be conducted to detect defects in unit interfaces and reduce the risk of dataflow and workflow failures in production.

B. Integration Strategy

* + All programmers will be responsible for integration testing of the entire product.
  + Integration testing will be conducted on every update of the system as a whole, where all sections will be check based on how current changes affected any other parts of the system.
  + Result cases are to be sent to the team leader.

C. Integration and Test Procedures

* + Any changes to modules needs to be approved by the team leader to mitigate risks in how an update will affect the rest of the system.
  + When one module is approved for changes, all other modules that interact with the given module requires being updated and turned into the team leader at the same time.
  + When an update is made, all programmers will check integration.

Access Control – Requires testing of valid login information, with security controls to prevent accessing pages not available by the user.

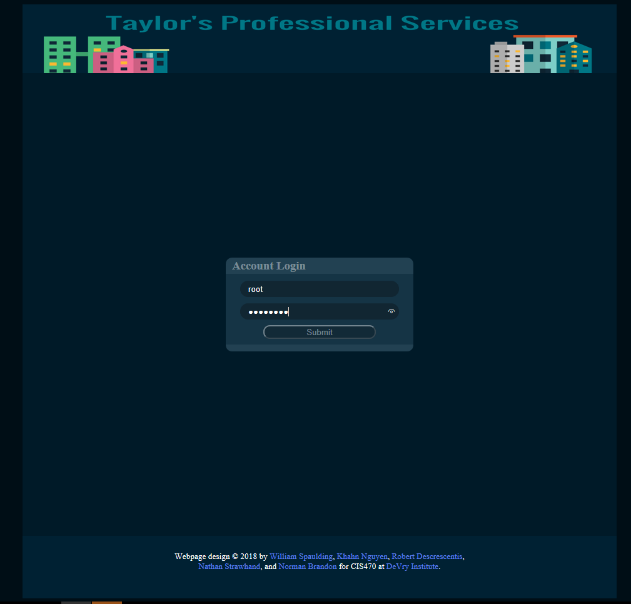
Database Management – Requires testing of interaction with the database, including adding, updating, grabbing and deleting data.

Interface and Business Rules – Requires testing of all different types of staff accounts, with the appropriate security levels assigned when logged in. Navigation as well tested for each are of the site accordingly.

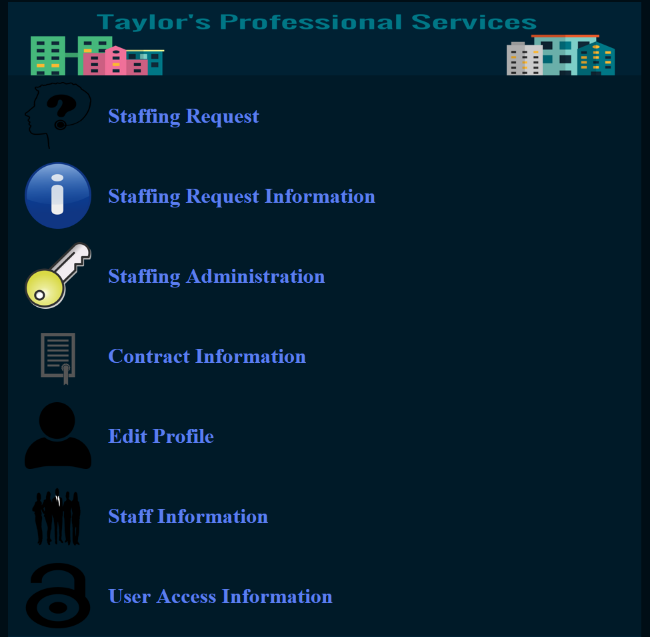
D. Integration Testing Results

Access Control Results –

Attempt at login with root manager userid

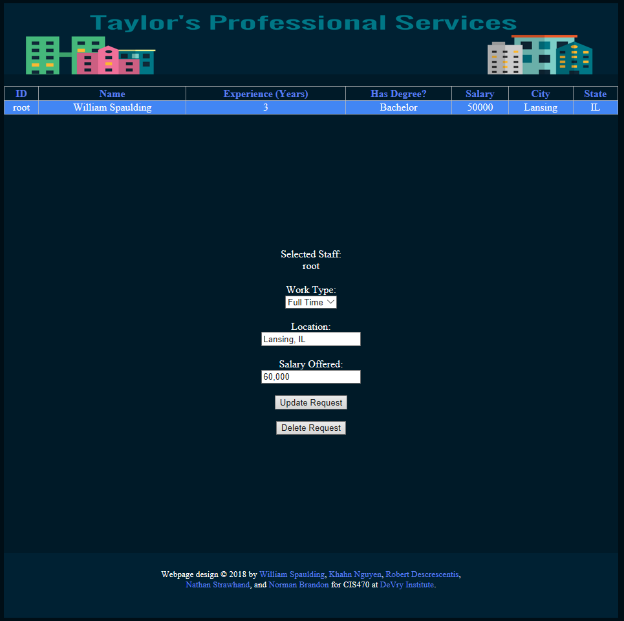


Login successful with appropriate security level

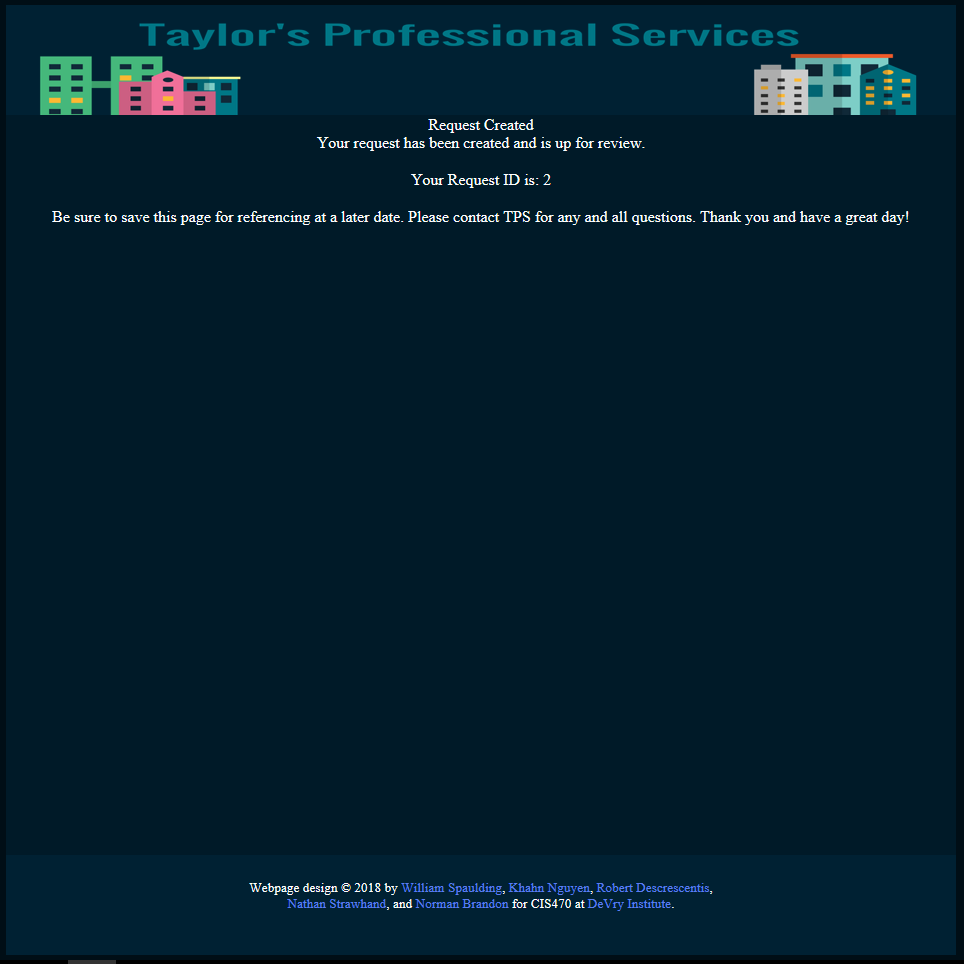


Database Management Results –

Attempting to create a staff request



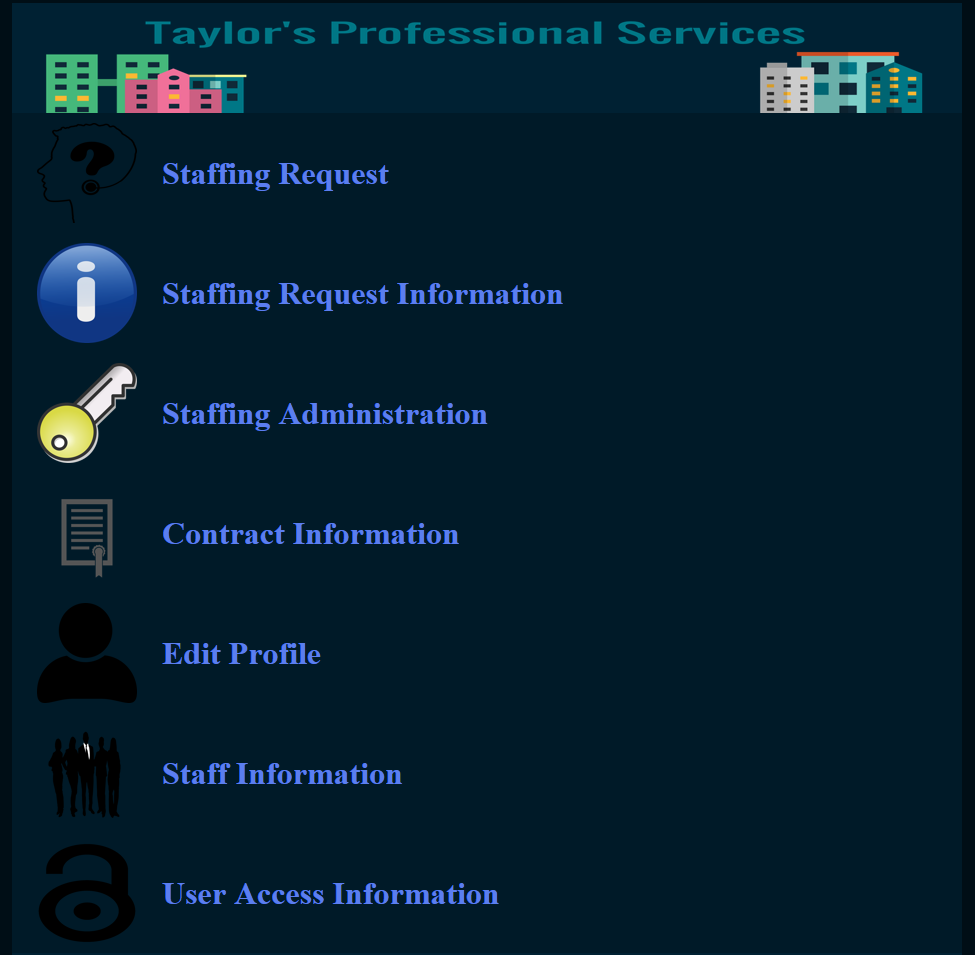
Request created, database interaction successful



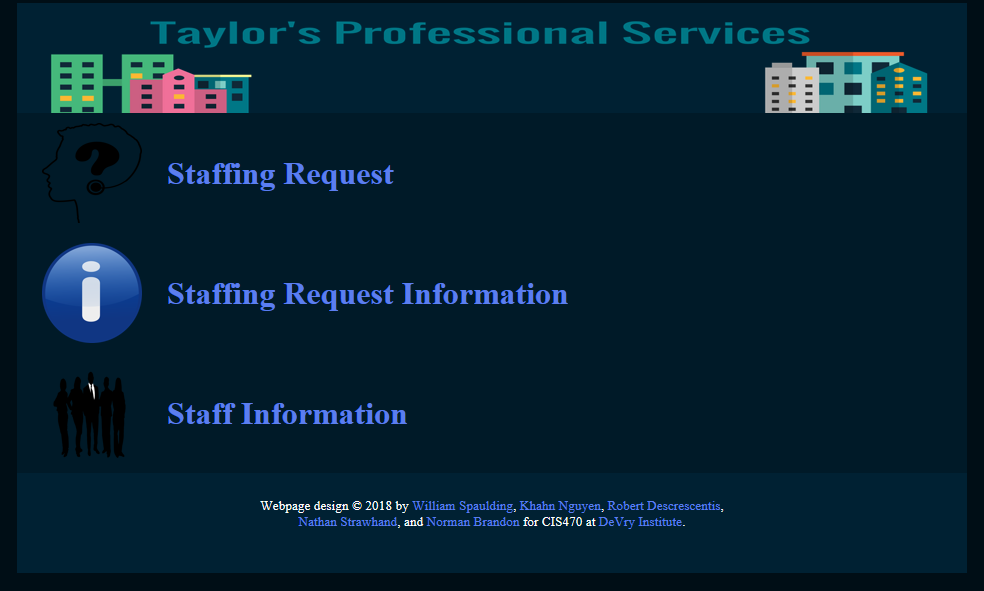
Interface and Business Rules –

* Allows a client to enter a staffing request into the database.
* Allows a client to retrieve staffing request information.
* Allows the contract manager to retrieve a staffing request from the database
* Allows the contract manager to retrieve contract information
* Allows the contract manager to validate the staffing request
* Allows the contract manager to close out the staffing request.
* Allows a staff member to update their personal information, resume, availability, and picture.

Manager has all permissions as outlined



Client has all permissions as outlined



Staff has all permissions as outlined

