Uncommon Solutions

Final Report

CMSC 495

12/15/2019

**Prepared By**

|  |  |
| --- | --- |
| Document Owner(s) | Project Role |
| Michael Kiefer | Project Manager |

**Requirements Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Change Description |
| 1.0 | 12/14/2019 | Michael Kiefer | Document creation |

**Summary Details**

|  |  |
| --- | --- |
| Participants | Name(s) |
| Project Manager: | Michael Kiefer |
| Project Team: | Hither Guzha – Technical Writer  Andrew Benson - Software Engineer  Donn Eddy - UX/HCI  Sean Mooneyham - Integration Engineer  Chase Thorpe - Test Engineer |
| End Users: | HR Departments |
| Description w/ Goal: | The purpose of this project is the implementation of an HR database and front end for personnel tracking. This document encapsulates the retrospective for the design and development process executed for this system*.* |

Table of Contents

**Overview3**

**Project Plan6**

**Requirements Specification12**

**System Specification13**

**User’s Guide16**

**Test Plan and Results22**

**Design 30**

**Development History38**

**Conclusions52**

**Overview**

# INTRODUCTION

## Purpose

The purpose of this document is to provide an overview of the development effort for the Uncommon Solutions HR Management System. The Uncommon Solutions HR Management System is being developed using an Agile SDLC framework. This document contains the necessary information required to effectively capture the development efforts of the team to include individual contributions based on assigned team roles.

## Background

The Uncommon Solutions HR Management System will be designed in a way that makes it easy to support multiple platforms such as Windows, macOS, iOS and Android. This web-based tool provides a direct method for storing and providing access to individual personnel records, and for all processes required for HR tracking and data aggregation requirements. The HR system will be implemented using AWS Elastic Compute Cloud (EC2) and Amazon’s Relational Database Service (RDS) in order to allow for universal deployability and access.

## Scope

This document covers the overall concept of the system implemented as the Uncommon Solutions project. It also includes a summary of individual contributions.

## Phases

The planned implementation of this project was broken down into three phases as follows:

1. Phase 1 consists of the creation of the database structure for information storage and the generation of the UI panels.
2. Phase 2 will consist of the functionality behind the login screen to include session management for the program. Additionally, user administration function to allow for management of user accounts will be implemented in this phase.
3. Phase 3 will consist of the data management functionality associated with this HR management system for the entry, modification, and management of the personnel information for the company.

## Schedule

The phased development schedule is a three-week process running from 18 November 2019 to 8 December 2019. There is an additional one-week flex time to allow for any schedule overruns and to allow for additional functionality to be added if time allows. This flex week runs from 9-15 December 2019.

## Design Considerations

The design for this program is as described in the Uncommon Solutions HR Management System Design Document. Any design variations will be validated by all members of the development team and incorporated into all design documentation to ensure that the entire development process is captured in documentation.

## System Overview

The system is a fairly simple implementation of a database and overlying UI for ease of customer access. Outside of the common login framework, the system branches to two different UIs depending on the account type being utilized. A system administrator account will go to the account management screen and have functionality and access not visible to any other users. The standard user accounts and more privileged user accounts will go to a separate screen with both individual HR record information but the ability to manage the records for other users with the correct privilege levels in place. For those users without elevated privilege, the additional management screens will not be visible.

By keeping the segregation between account management and data management we are adhering to a basic security tenet of least required privilege. A system administrator has no need to access the data within the HR system, and so is not even presented with the UI to access that information. By also including auditing of accesses, we can ensure that a trail will be present should any malicious attempt to access or abuse the system occur.

**Individual Contributions**

## Michael Kiefer

As Project Manager Michael was responsible for the organization and division of project effort for the duration of this development effort. In addition, he took responsibility for ensuring final edits and compilation of group member efforts prior to submission for each assignment. Also, during the development phases Michael maintained an ongoing report of phase progress to ensure that all development efforts were proceeding as planned in the design phase and that the developed software fulfilled the planned requirements. Finally, Michael ensured that the final submission was complete and prepared for individual submission.

## Hither Guzha

As Technical Writer Hither was instrumental in setting the stage for the appropriate documentation of the project and development effort. Her initial submission of the Project Plan has been utilized as the baseline template for all documents produced by the team during the course of the project. Her initial lift has driven the team success through this project and she continued to be of great value in drafting/editing documentation to include the Users Guide, ensuring a seamless final delivery.

## Andrew Benson

As Software Engineer Andrew was initially just pushed for a look at design considerations and overall system concepts. Once we neared development, Andrew has been a solid pillar of the team, providing the initial UI wireframes for inclusion in the finalized design. After that, he executed everything required of him in each phase, from the basic UI display in Phase 1, to login and system account management in Phase 2, culminating in records display and editing based on user access level in Phase 3. Some minor issues identified in Phase 3 testing were rectified within the first 3 days of our flex week, leading to a finalized product that completed successful testing before the scheduled delivery date.

## Sean Mooneyham

As Integration Engineer Sean was instrumental in the success of the project. While there was some confusion as to what exactly this role encapsulated, Sean took it to be the management and integration of backend services to present to the frontend software. He was responsible for the implementation of our cloud-based architecture through the creation of AWS instances for database and hosting capabilities. Also he instantiated and controlled our development platform on GitHub. Additionally, Sean assisted all team member with accessing the development and integration environment as well as ensuring that our baseline was updated as required. Finally, Sean developed and delivered the access APIs for database to program connectivity.

## Donn Eddy

As the UX/HCI member of the team Donn provided valuable inputs to the initial user design for the system. His schedule limitations precluded his participation in a lot of the group collaboration directly, but our work via chat on discord provided the opportunity for him to add his piece. Our chosen project and implementation were entirely outside of his scope of experience so this was a significant challenge for him he was still able to contribute to the concepts outlined in the system design.

## Chase Thorpe

As Test Engineer Chase developed the Test Plan for our software. He then revised the test plan as the implementation was completed to ensure that all aspects of the software were being appropriately tested. Finally, Chase executed the Test Plan and recorded the results of the testing for feedback to the development team. In addition, Chase has been an active contributor in group collaboration for planning and design for the system, as well as stepping up to fill gaps as needed.

**Project Plan**

# PROJECT OVERVIEW

## Project Title

HR Personnel Tracking Project, Uncommon Solutions, Group 3, CMSC 495 7980 Current Trends and Projects in Computer Science.

## Scope, Goals and Objectives

The overall goal of the HR Personnel Project is to implement a HR database and front end for personnel tracking to be used by several HR departments. This web-based tool is useful for storing and providing access to individual personnel records, and for all processes that HR want to track and from which they hope to gather useful and purposeful data (Heathfield 2019). It will include individual’s personal data, contact information, job title, job description, emergency contact information, training and certifications, resumes and any employee personnel actions records. Only users with the appropriate level of access will be able to login to the tool and modify/delete any records to ensure adherence to privacy standards. Additionally, in order to maintain compliance with secure coding standards, best practices for data storage and protection will be utilized.

The business goals and objectives for this project will focus on:

* Implementing a tool that will provide efficiency to human resources department by keeping all employees’ information at a central location (Brooks, 2019)
* Providing secure transmission and storage of employee’s details and information through appropriate use of encryption and access controls
* Provide a HR tool that will be easy for human resource management to navigate, track and analyze employee’s information
* Facilitate coordination and information sharing between employees and HR management
* Enhance employees’ ability and effectiveness to do their own updates to benefits, address, job description, and emergency contact information (Heathfield, 2019)
* Accomplish project goals and objectives within defined budget and time parameters

## Assumptions/Constraints

The following assumptions were made in preparing the Project Plan:

* The Project Plan may change as new information and issues are revealed
* Project must be completed within the timeline limitations of this course (8 weeks)
* Project team members will adhere to the Communications Plan

## Project Deliverables

The lists of project deliverables are:

* Finalized Project Plan
* Users Guide and Test Plan
* Design of the Project
* Peer Reviews
* Source codes
* Final Report

## Project Budget: For the implementation of a HR Database, the project will utilize FREE Tier instances provided through AWS hardware bases and will free download all software bases used. Should this project be implemented for use at a larger scale, it would require the budgetary allocation for the appropriate costs for AWS utilization from the customer’s cost center.

# PROJECT MANAGEMENT APPROACH

## Development Process

### Preparing Specifications for the Project Design

The specification defines the contents, source code, design and the customer’s needs of the HR web-based tool. This gives the project team a clear picture of what the final deliverables will contain and look like before it is developed. A well written and reviewed functional specification is important because it provides a clear and explicit description of exactly how the projects technical requirements are to be met (Maverick, 2012).

### Conducting Requirements Meetings

The project team will conduct meetings to gather requirements and ideas that will be analyzed to be able to develop a well-articulated HR web-based tool used for personnel tracking. Meetings will be conducted as defined in the project schedule, at a minimum of bi-weekly.

### Content Development

This process will involve researching, writing, gathering, organizing and editing information required for developing the final deliverables. Content will be developed by assigned team members with collaboration to ensure all final deliverables represent a cohesive team vision and product.

### Editing and Reviewing with Project Team

Once a draft of any document or code is completed, a copy will be checked in to GitHub for the whole project team to review. This is usually an informal review immediately after completing writing, but before editing. The project team will be asked to look for the following:

* Missing information
* Unnecessary information
* Technical accuracy

Project team inputs will be incorporated into the final product prior to delivery to the customer or designation as a final version of the product.

## Editing Draft

The initial substantive edit will be completed after the first draft of each section is completed and has been reviewed and revised by the project team before submitting to LEO for grading. Final versions will be all code files and documents for the project zipped together and submitted to LEO upon completion of the project.

## Project Timeline / Schedule

The following represent key project timelines and schedule which includes milestones, task dependencies, task duration, and delivery dates to complete the project.

|  | **Group 3–Uncommon Solutions Project** | | | | | |  | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | |
|  |  | **WK 1- 2** | **WK 3** | **WK 4** | **WK 5** | **WK 6** | **WK 7** | **WK 8** |
| **1** | Project Kick Off |  |  |  |  |  |  |  |
| 1.2 | Finalized Project Plan |  |  |  |  |  |  |  |
| 1.3 | Users Guide and Test Plan and Peer Reviews |  |  |  |  |  |  |  |
| 1.4 | Design |  |  |  |  |  |  |  |  |
| 1.5 | Phase 1 Source and Peer Reviews |  |  |  |  |  |  |  |
| 1.6 | Phase 2 Source |  |  |  |  |  |  |  |
| 1.7 | Phase 3 Source |  |  |  |  |  |  |  |
| 1.8 | Final Report and Final Peer Review |  |  |  |  |  |  |  |

## Project Roles and Responsibilities

| **Role** | **Responsibilities** | **SDLC Lead** | **E-mail address** |
| --- | --- | --- | --- |
| Project Manager | 1. Provide schedules and peer reviews 2. Provide project oversight and guidance 3. Review/approve some project elements | Michael Kiefer | mkiefer2@student.umuc.edu |
| Technical Writer | 1. Provide high level and detailed requirements 2. Identify and document existing business process 3. Provide draft Project Plan, User Guide and Test Plan | Hither Guzha | hguzha@gmail.com |
| Software Engineer | 1. Develops information systems by designing, developing, and installing software solutions | Andrew Benson | ab@bensonfamilies.net |
| UX/HCI | 1. Provide a better user experience of web-based applications | Donn Eddy | maindric2@gmail.com |
| Integration Engineer | 1. Evaluates and test engine calibrations with software systems 2. Integrate system architecture 3. Evaluate patches & plan release management | Sean Mooneyham | smooneyham@student.umuc.edu |
| Test Engineer | 1. Prepares test plans for web-based tool | Chase Thorpe | chasethorpe1@gmail.com |

## Risk Assessment

Risk Assessment is the process by which a project team tries to identify, characterize, prioritize and document any risk factors that could negatively impact the project. These assessments help identify the project risks and provide measures, processes and controls to reduce the impact of these risks to the project.

Here are some of the risk factors:

* Project Team availability: If identified as a current risk factor, it is the responsibility of any team member to let the Project Manager know in advance when they will not be available so he can find coverage for the project.
* Behind with the project timeline: The Project Manager will continuously review the timeline weekly to prevent unnoticed timeline issues and shift workload as identified in the previous risk.
* Unclear project scope: The project scope is initially defined in the project plan; therefore, the Project manager and team will continuously revisit and review the scope to ensure it is clear and every team member is on track. This will be mitigated through workload redistribution as required.
* Over budget: The project manager will ensure the project is on scope and stays within the budget. Our current projection is for a zero cost demonstration solution.

The Project manager and team will continuously monitor and document any risk factors throughout the life of the project, with weekly assessments included in the weekly meetings (see **Communications Plan**) and open to amendment by the Project Manager.

The Project Manager will convey amendments and recommend contingencies to the Project team weekly, or more frequently, as conditions may warrant.

## Staffing Plan/Content Development

The assigned Group 3 -Uncommon Solution Team have the skills and experiences necessary to ensure a successful project completion. Overlap in individual skills allows us to ensure that all tasks have both a primary and backup task owner at a minimum.

## Review & Approval of Contents

All project reference sections will be reviewed and approved by the project team prior to deployment.

## Issue Management

As the Project progresses, the information contained within the Project Plan will likely change to best meet the customers’ needs. To avoid project failure, the project team must have effective management processes to address any requirements issues in the very early stage of the project life cycle (Kumar, 2006). During a project, requirements are certain to change; therefore, it is important to note that any changes to the Project Plan will impact the budget of the project, availability of team members and quality of the project (Kumar, 2006). This project plan (including project scope and resources) may be modified by following the procedures listed below:

**Step 1:** The Project Manager should document all issues that impacts the Project Plan. The project manager and stakeholders all expect the same outcome; therefore, it helps to document, and track activities and issues related to the project.

**Step 2:** For issues found during the project, the Project Manager will review and determine how the issues will impact the project, and forward the issues, along with recommendations, to the Project Team for review and decision.

**Step 3:** The Project Team should then review the recommendations from the Project Manager and reach a consensus opinion on whether to approve, reject or modify the request. Should the Project Team be unable to reach consensus on the approval or denial of a change, the issue will be forwarded to the Project Manager with a written summary of the issue, for ultimate resolution.

**Step 4:** If the Project Manager does not agree with resolution provided by the team, the Project Sponsor will review the issue(s) and render a final decision on the approval or denial of a change.

**Step 5:** Following an approval or denial (by the Project Team or Project Sponsor), the Project Manager will notify the original requestor of the action taken. There is no further appeal process.

## Communications Plan

A project communication plan is essential because it sets clear guidelines of how information will be shared, as well as who’s responsible for what in a project (LaPrad, 2018). Disseminating knowledge about the project is essential because it increases visibility of the project and status and for project’s success. The Project Manager should continuously update the project team with the status of the project and how they are affected by any issues. This helps in boosting the productivity of the team (LaPrad, 2018). Furthermore, communication plans ensure that the project continues to align with goals. If there are any issues within the project, the team shall communicate all issues to the Project Manager for resolution.

This plan provides a framework for informing, involving, and obtaining buy-in from all participants throughout the duration of the project.

**Audience** This communication plan is for the following audiences:

* Project Manager
* Project Team members

**Communications Methods** The following is a list of communication methods that will be used for this project:

* UMGC LEO discussion platforms
* Email
* Discord
* Conference calls

**Communications Outreach** The following is a list of communication events that are established for this project:

* Bi-weekly team meetings
* Collaboration and finalization via electronic communication

### Weekly Meetings: The Project Manager shall provide meeting notes to the project team after every meeting. The meeting notes shall include the following information tracked against the Project Plan:

* Summary of tasks completed since last meeting
* Summary of tasks scheduled for completion in the next week
* Summary of issue status and resolutions

**Requirements Specification**

# INTRODUCTION

# Purpose

The purpose of this document is to detail the software requirements specification of the Uncommon Solutions HR Management System. The Uncommon Solutions HR Management System is being developed using an Agile SDLC framework. Any deviations from the planned Uncommon Solutions HR Management System will be reflected by updated changes to the associated design and system management documents. This document contains the necessary details to implement the software requirements for the planned development process.

## Background

The Uncommon Solutions HR Management System will be designed in a way that makes it easy to support multiple platforms such as Windows, macOS, iOS and Android. This web-based tool provides a direct method for storing and providing access to individual personnel records, and for all processes required for HR tracking and data aggregation requirements. The HR system will be implemented using AWS Elastic Compute Cloud (EC2) and Amazon’s Relational Database Service (RDS) in order to allow for universal deployability and access.

## Scope

This document provides a clear overview of the designed software requirements for the completion of the Uncommon Solutions HR Management System. Specific system requirements for development and deployment are covered within the System Specification accompanying this document.

## Assumptions

The following assumptions are relevant to the design of the proposed system:

* The proposed new system will leverage the Uncommon Solutions HR architecture.
* The existing architecture and system design will be used including all existing components and sub-systems.
* It is assumed that additional functionality will be added to the proposed solution as required during development and testing.

## Constraints

* There are no hardware or software technical constraints identified with this project.
* System interoperability may be a constraint since the design will leverage free tier AWS EC2 instance and RDS with the potential to expand to paid utilization at a larger-scale fielding.

System Specifications

# INTRODUCTION

# Purpose

The purpose of this document is to detail the system specification requirements of the Uncommon Solutions HR Management System. The Uncommon Solutions HR Management System is being developed using an Agile SDLC framework. Any deviations from the planned Uncommon Solutions HR Management System will be reflected by updated changes to the associated design and system management documents. This document contains the necessary details to implement the system level requirements for the planned development process.

## Background

The Uncommon Solutions HR Management System will be designed in a way that makes it easy to support multiple platforms such as Windows, macOS, iOS and Android. This web-based tool provides a direct method for storing and providing access to individual personnel records, and for all processes required for HR tracking and data aggregation requirements. The HR system will be implemented using AWS Elastic Compute Cloud (EC2) and Amazon’s Relational Database Service (RDS) in order to allow for universal deployability and access.

## Scope

This document provides a clear overview of the designed system requirements for the completion of the Uncommon Solutions HR Management System. Specific software requirements for development and deployment are covered within the Requirements Specification accompanying this document.

## Assumptions

The following assumptions are relevant to the design of the proposed system:

* The proposed new system will leverage the Uncommon Solutions HR architecture.
* The existing architecture and system design will be used including all existing components and sub-systems.
* It is assumed that additional functionality will be added to the proposed solution as required during development and testing.

## Constraints

* There are no hardware or software technical constraints identified with this project.
* System interoperability may be a constraint since the design will leverage free tier AWS EC2 instance and RDS with the potential to expand to paid utilization at a larger-scale fielding.

**5 System Requirements**

## Operating Environment

The product system design utilizes two different AWS services, Elastic Compute Cloud (EC2) and Amazon’s Relational Database Service (RDS). Using AWS as a platform for fielding the application is the most cost-effective way to host the application while also providing access to the application for all project team members and product owners. AWS is an excellent platform for quickly spinning up application prototypes and rapidly deploying solutions to multiple customers.

The application uses Apache version 2.4.29 as a web server and is installed on the EC2 T2 instance and uses Ubuntu 18.04 as an operating system. All source code including the user interface and API will be on the apache server. The application uses a MySQL database installed on an RDS instance to house all the application data.

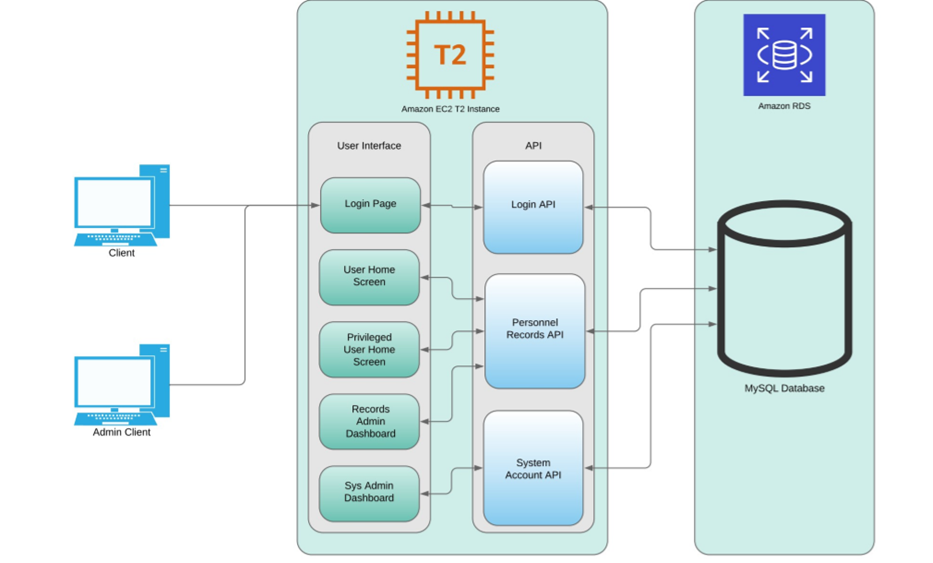


Figure 1: System Design Diagram

## Design and Implementation Constraints

As this is a proof of concept demonstration and not a final fielding, the implementation will be restricted to the storage and processing capabilities provided within the free tier of AWS processing. A fielding for customer utilization would require instances generated on that customer’s AWS hosting services account and could be scaled to the level of storage and processing required for the size of their organization. The implementation plan must be something that can be completed by a small and relatively inexperienced team within the implementation timeline of three weeks with a fourth week available as project flex if required.

## Assumptions and Dependencies

The system is being developed based on the assumption of existing cloud services being present and available for development and implementation. Final functionality is dependent on the availability of hosting services for the database and front-end hosting and functionality. The underlying hosting services for both are expected to remain functional on the hosting server. While the designed implementation makes use of AWS hosting services, any client with an independent web hosting and database hosting capability will be able to internally execute the client with minor configuration changes.

## Hardware Interfaces

As the hardware is controlled by a third-party entity (AWS) there are no hardware interfaces for our system. The Uncommon Solutions HR Management System is platform agnostic so long as configuration files are set to represent the hardware hosting the system.

## Software Interfaces

As the hardware is controlled by a third-party entity (AWS) there are no software interfaces for our system. The Uncommon Solutions HR Management System is platform agnostic so long as configuration files are set to represent the software hosting the system. The current implementation utilizes MySQL and PHP interfaces, so long as the hosting system is executing these system level services there are no additional interface requirements.

## Communication Interfaces

In order to correctly function, the Uncommon Solutions HR Management System must be hosted on an environment with network access for all client systems. The initial build implementation makes use of AWS hosting, allowing for access from any client systems with internet access. Should a client wish to implement on a closed network system, the only requirement is that all intended clients be able to have network access to the hosting site.

## Software Quality Attributes

All software design will adhere to industry standards for modularity, programming structure, algorithm efficiency, object-oriented design and clear and understandable in-source documentation (commenting). Known security vulnerabilities will be protected against and the software will be built in such a way as to ensure that future optimizations, security fixes, and expansions will be able to be implemented without additional effort to understand existing source code.

## Performance Requirements

The program will execute making efficient use of processing resources. Page loads should never exceed 5 seconds on a client system considered to be of current (within 5 years) processing capabilities. System load generated should not exceed that required to execute required processing and the use of efficient algorithms should be maximized to keep per-user load and storage costs to a minimum.

## Security Requirements

Access controls must be in place and function to ensure that only authorized users have access to the system. Privilege escalation protection will be in place to ensure that only user accounts with the appropriate privilege level have access to administrative functions and larger data level access. This will be managed through internal security controls of the software to validate session management and user access level within every program module, preventing any known-source attacks.

**User Guide**

# Purpose

The purpose of this document is to provide user guides for the Uncommon Solutions HR Management System to all users. The Uncommon Solutions HR Management System is a HR database and front end for personnel tracking to be used by several HR departments This document contains the necessary information required to effectively access the HR Management System.

## Scope

This document describes best practice and technical user activities for all users. This is a living document and will be updated as changes are made to the HR Management System.

## Assumptions

The following assumptions are relevant to the user guide of the proposed system:

* The User has a basic understanding of using the system.
* The User has appropriate privileges to access the system.

## Cautions & Warnings

Users of The Uncommon Solutions HR Management System must be aware of warnings regarding unauthorized access to the system. There will be logs to track user’s activities on the system. The user must read and agree to the Terms and Conditions including the Privacy Act statement and the Rules of Behavior before accessing the online application.

# Getting Started

## Set-Up Considerations

The user should get correct access from the administrators to access the system. User can access the system from any browser.

## User Access Considerations

The Uncommon Solutions HR Management System consists of the following types of system users:

* Application Users.
* Privileged Users.
* Administrators.

## Accessing the System

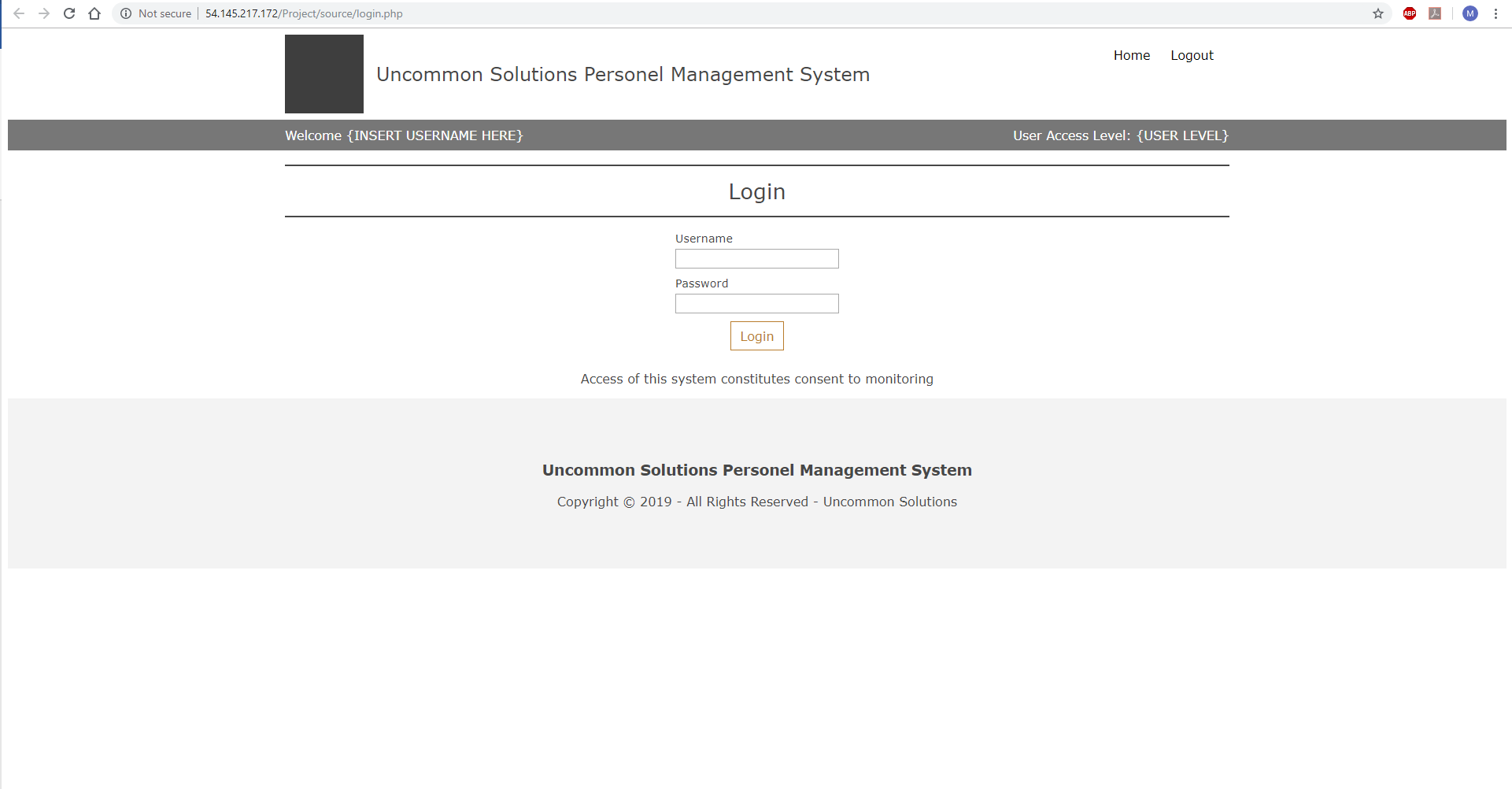
The Uncommon Solutions HR Management System can be accessed through any web browser. Users may access the system by using the following link

<http://3.81.54.213/Project/source/login.php>

## The Uncommon Solutions HR Management System Access Page

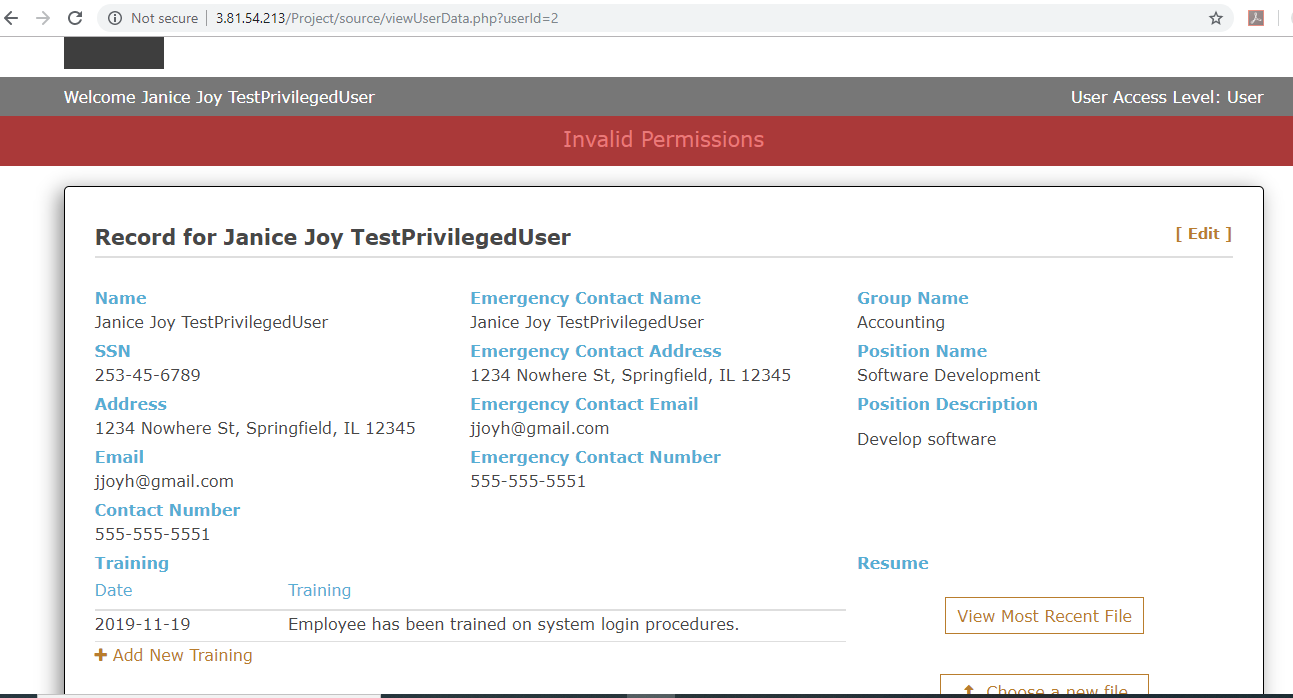
The Uncommon Solutions HR Management System Access Page is the main page for accessing the system’s application and self-service features.

To access the system, login with your credentials on this page.



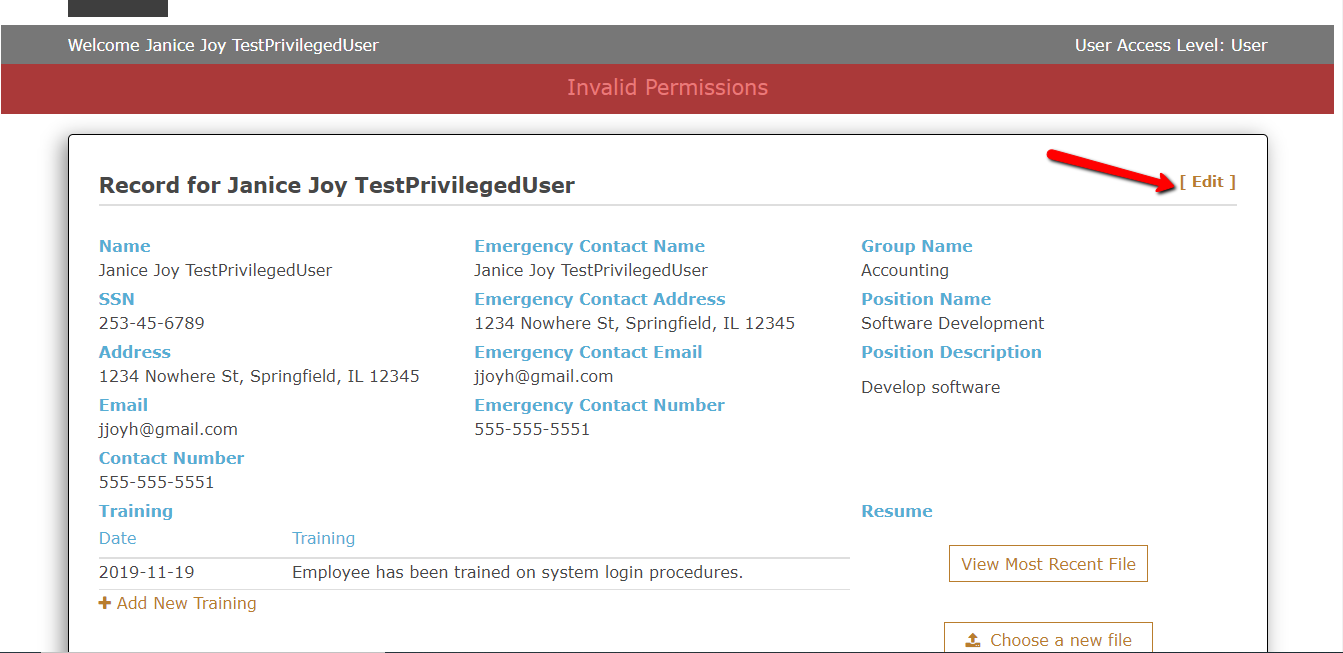
## Overview of the Management Console

Personnel information of every user is displayed on the Management Console after successfully login.

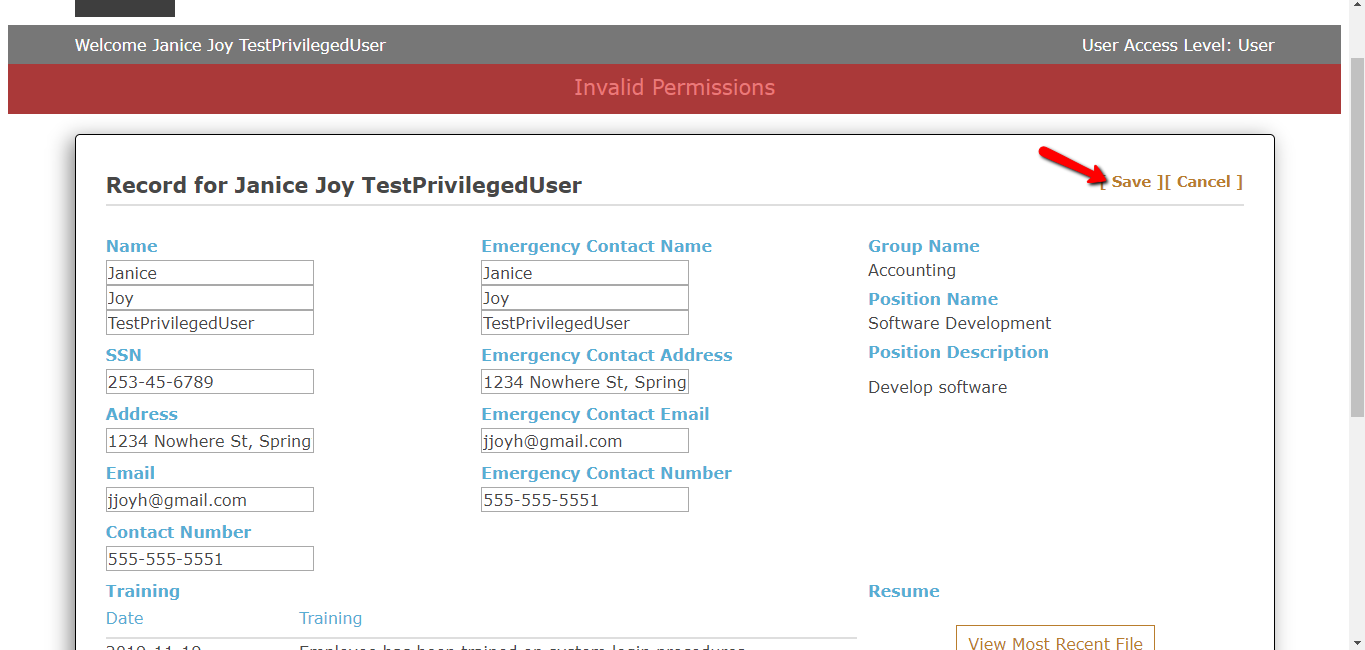


## Editing User’s Information

To edit any information, click on the **Edit** button

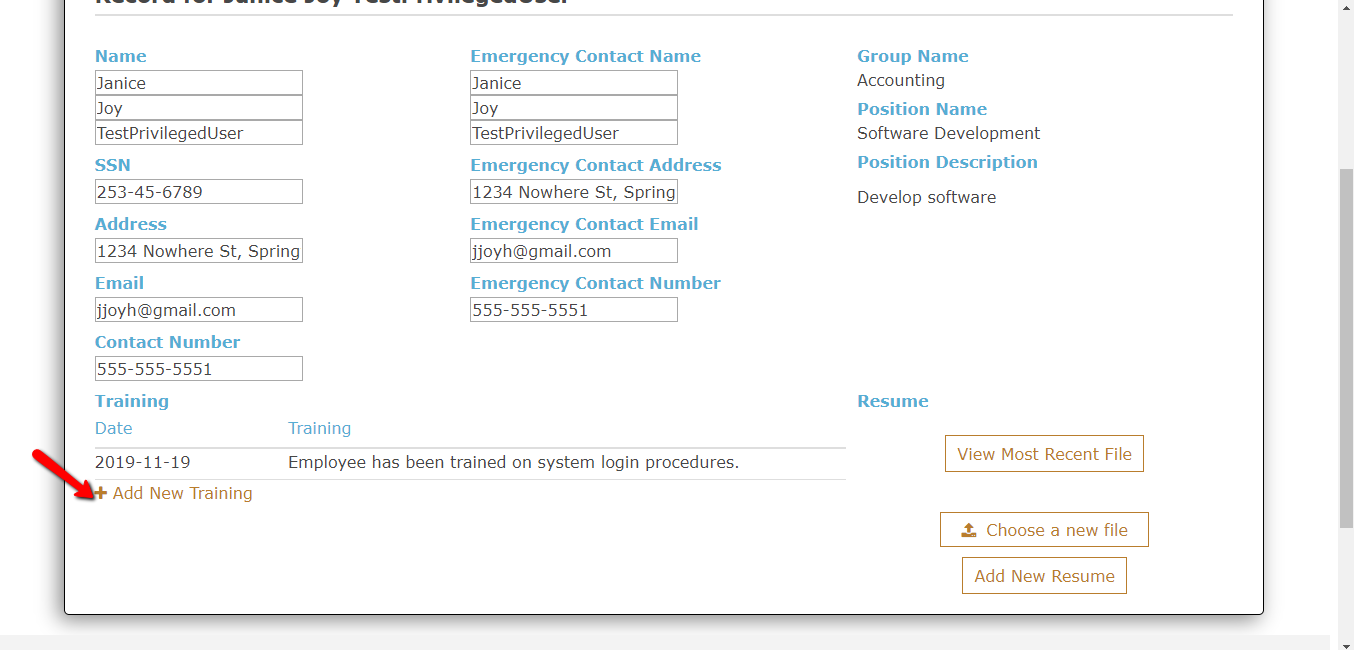


Select the section in which you need to edit. After edits, click the **Save** button to save all changes made.

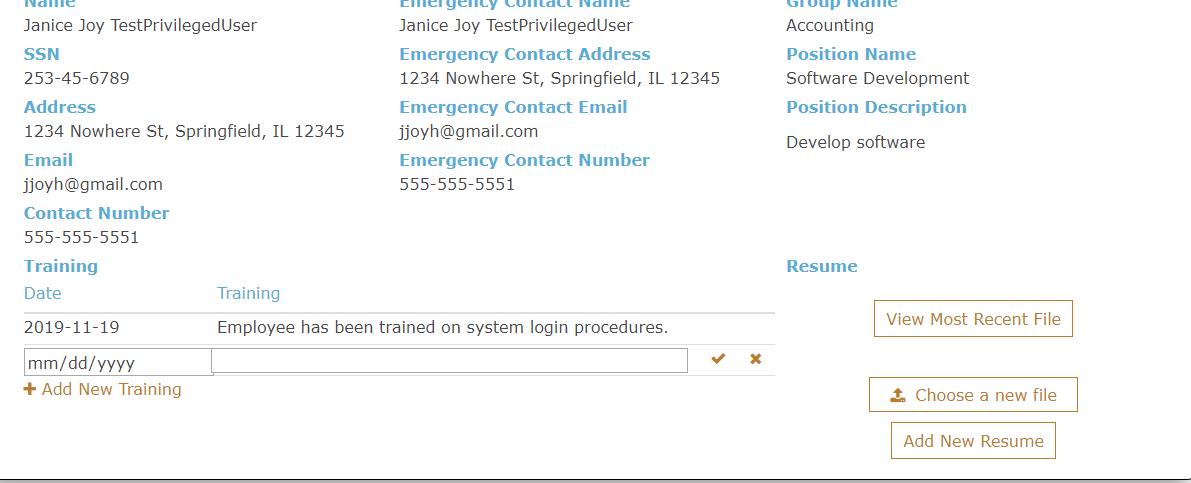


## Adding A Completed Training

To add a completed training, click on the **Add New Training** button.

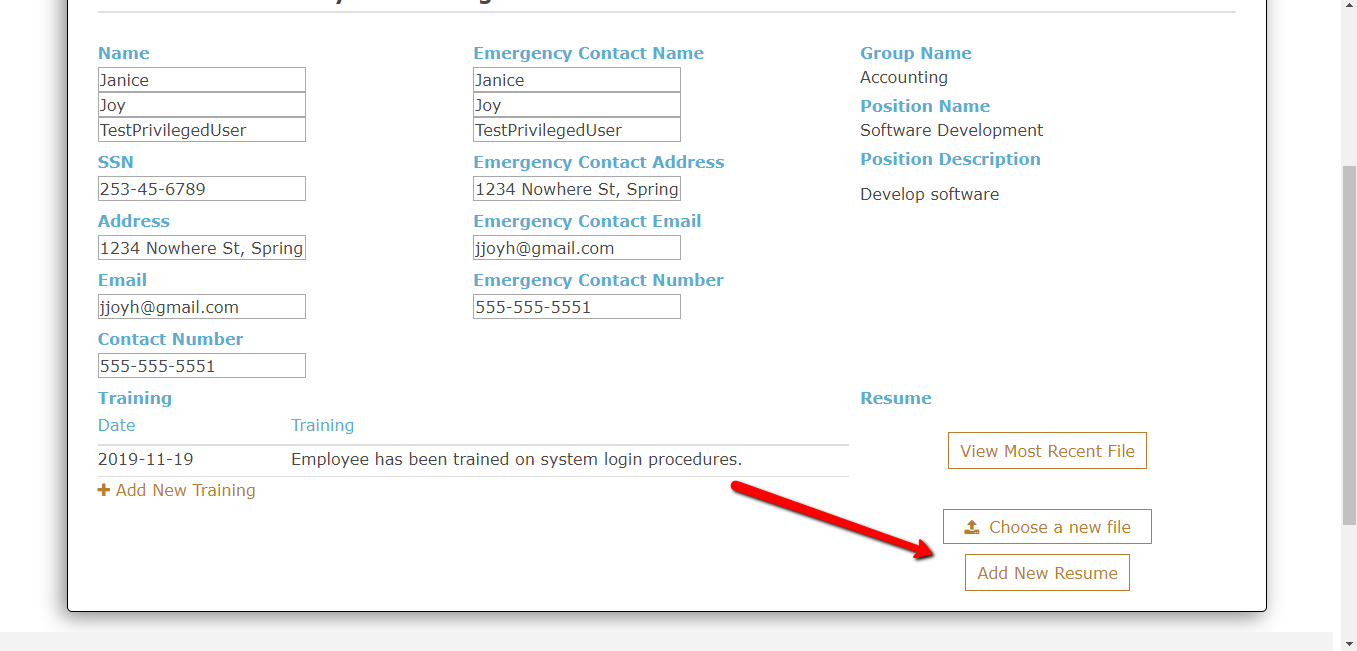


Enter the date the training was completed and the training details, click **Save** to save new details.



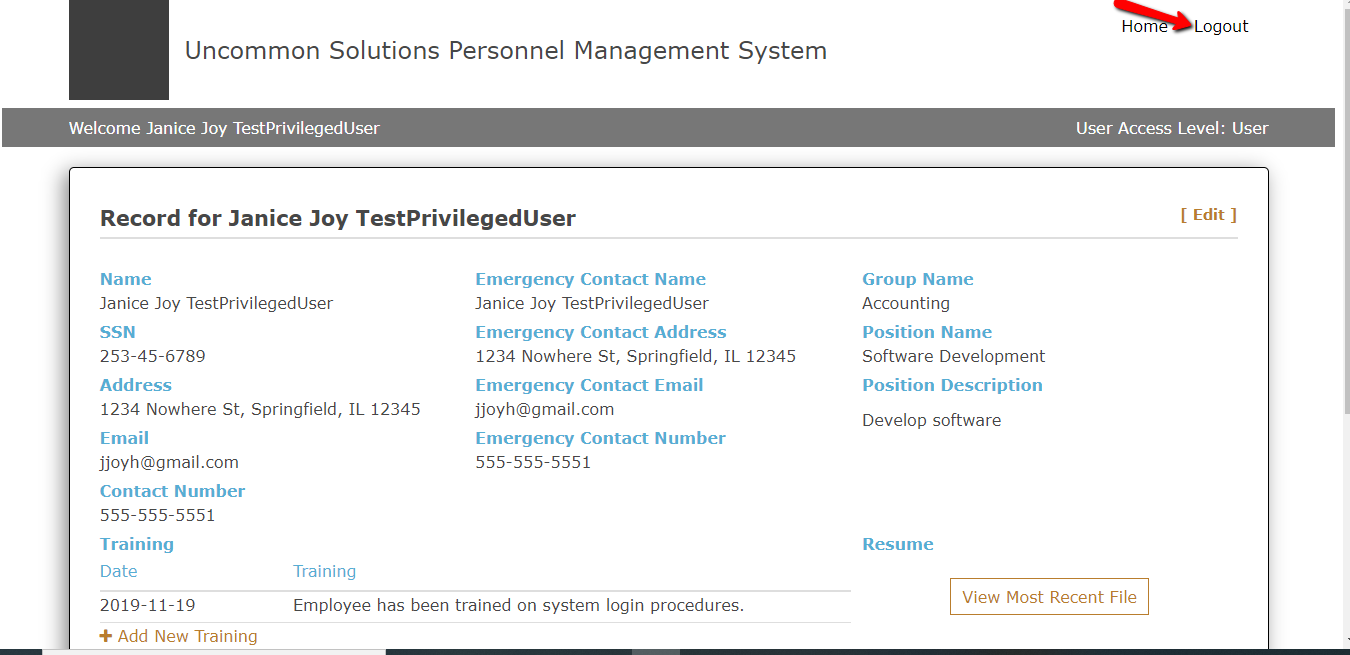
## Adding A New File and Resume

To add new resume or file click on the **Choose a New File** and select the desired file, upload the file and click save.



## Logout to the System

When you have completed all necessary modification, you can logout to the system by clicking **Logout**.



## User Assistance

You can contact the Administrators for any assistance.

**Test Plan and Results**

# PURPOSE

The Uncommon Solutions HR Management System provides access to personnel records of a company. The system separates user roles into four categories: User, Privileged User, Records Admin, and Account Admin. Each of these roles possess different levels of access and control over personnel records. Access to this system is controlled via authentication mechanisms. Relevant actions taken by users (outlined below) are logged by the system.

## BACKGROUND

Due to geographic separation of team, testing will be performed on cloud infrastructure; specifically, on an AWS EC2 instance. This will also serve to demonstrate execution of a deployed installation with users at multiple locations.

## SCOPE

***In Scope:***

The test plan for the Uncommon Solutions HR Management System will consider the correct execution of all code blocks relevant to the creation, modification, retrieval, and deletion of personnel records as well as correct authentication/authorization handling and logging as IN SCOPE.

***Out of Scope:***

Performance, scalability, stress testing, database testing, and anything not directly linked to the items listed above will be considered OUT OF SCOPE.

## GLOSSARY

NA

## LIMITATIONS AND TRACE ABILITY

## Limitations

**Risk Mitigation**

Time Constraints Set test priority for each activity

Team lacks experience in software Conduct research on industry best practices

development/testing and collaborate to mitigate individual gaps

# TEST PLANS

## Test Levels

Due to time/budget constraints, only the following types of testing will be performed for this system:

### API Testing

All read/write operations against personnel records, authentication processes, and logging functions will be tested independent of other modules.

### Integration Testing

At this level of testing, interaction between relevant modules will be tested. For example, ensuring that the correct audit logs are generated when a user logs in or logs outs, etc.

### System Testing

This level of testing will examine the system as a whole and ensure that the system functions as intended based on predefined acceptance criteria.

## Test Environment and Schedules

This section provides a brief description of the inputs, outputs, and functions of the software being tested.

### Software Description

The Uncommon Solutions HR Management System is a standalone system that manages the personnel records for the employees of a company.

### Milestones

Test Plan: Due 10NOV19

Project Design: Due 17NOV19

Phase 1 Source Code: Due 24NOV19

Phase 2 Source Code: Due 01DEC19

Phase 3 Source Code: Due 08DEC19

Final Product: Due 15DEC19

### Organizations and Locations

This software will be tested on an AWS EC2 instance.

Public DNS (subject to change): ec2-54-145-217-172.compute-1.amazonaws.com

IPv4 Public IP (subject to change): 54.145.217.172

### Resource Requirements

This section and associated statements define the resource requirements for the testing.

#### Equipment

The only required equipment is a computer with an internet connection that will connect to the AWS EC2 instance where the system lives.

#### Software

The computer must have a modern browser installed (Chrome, Firefox, Edge, Safari, Opera). Much of the testing will be conducted with manual inputs so very little additional software is required. A 3rd party testing framework like Jest or Mocha may be included as needed.

#### Personnel

Testing will be performed primarily by the Test Manager, Chase Thorpe. The entire team will provide oversight and support on testing.

### Testing Material

NA

### Test Training

NA

### Test Methods and Evaluation

This section documents the test methodologies, conditions, test progression or sequencing, data recording, constraints, criteria, and data reduction.

#### Methodology

Much of the initial testing for this system will involve unit testing of the various components of the system to ensure they function as intended. Each function in the read/write operations of personnel records, authentication processes, and logging functionality will be tested in isolation before being tested at the subsystem level.

#### Conditions

Test data will be entered in real time. Because stress/performance testing is out of the scope of this test, transactions per second and similar time-based testing will not be performed.

#### Test Progression

As mentioned in the methodology section, individual unit tests will be conducted in succession. After the unit tests have been completed, interactions between the components of the system will be tested.

#### Data Recording

Test results will be recorded in a shared document between the team.

#### Constraints

The team foresees no constraints other than those mentioned above at this time.

#### Criteria

Each test will be performed with two valid inputs and two invalid inputs. If all of these inputs line up with expected output, the test will be considered successful.

#### Data Reduction

NA

# TEST DESCRIPTION

## User Login (Valid)

### Test Description

This test will ensure that a valid user can successfully login with a username and password

### Control

This will be a manual test. Username and Password will be entered, and the success or failure of the attempt will be recorded.

### Inputs

Two inputs will be required for this test. First, a valid username. Second, the password associated with that username. These inputs will have to comply with the predetermined username and password policy.

### Outputs

With a valid username and password, we expect to be successfully authenticated and shown the user dashboard. In addition, the successful login will be recorded in the authentication audit log.

### Procedures

Prior to the test being performed, a user account will be manually created. Next, the tester will navigate to the login screen and enter the username and password into the specified fields. Upon submission of credentials, either a success or failure message will be displayed. This result will be recorded. The expectation for a successful attempt is the success message and redirection to the user dashboard.

## User Login (Invalid)

### Test Description

This test will ensure that an invalid user is prevented from logging in.

### Control

This will be a manual test. Username and Password will be entered, and the success or failure of the attempt will be recorded.

### Inputs

Two inputs will be required for this test. The first invalid input will be an invalid username and the second invalid input will be an invalid password. These inputs will have to comply with the predetermined username and password policy.

### Outputs

With an invalid username and password, we expect to be prevented from logging in to the system. In addition, the failed attempt will be logged in the authentication audit log.

### Procedures

No user account will need to be created for this test. Two incorrect inputs will be entered and submitted. Regardless of the combination of incorrect inputs (correct username/incorrect password, incorrect username/correct password, incorrect username/incorrect password), we expect to see an authentication failure message and be prevented from moving past the login screen.

## Unique Username

### Test Description

This test will ensure that duplicate usernames cannot be created.

### Control

This will be a manual test at the user creation screen.

### Inputs

The only input required will be the username being tested.

### Outputs

We expect a message to be shown for user creation error stating that an account of that name already exists.

### Procedures

Prior to the test being performed, a user account will be manually created. Next, the tester will navigate to the user creation screen and attempt to create a new user with the same username as the previously created account.

## Verification of User Role Privilege

### Test Description

This test will ensure that a valid user can interact with the system within the established controls and privileges granted by the various roles within the system (user, privileged user, records admin, account admin)

### Control

This will be a manual test. Each account type will be logged into and various actions will be attempted. Results of successes and failures for both authorized and unauthorized actions will be recorded.

### Inputs

Usernames and passwords for each account will be used as inputs. Each account will attempt to perform authorized actions within their role as well as unauthorized actions to ensure permissions are performing as expected.

### Outputs

We expect regular users to be able to access their personnel records and make minor changes to certain fields. We expect privileged users to have read access to all personnel records but only be able to make minor changes to their own personnel record. The Records Admin should have full read/write access to all personnel records. The Account Admin has full read/write access to user accounts and no access to the personnel records.

### Procedures

Prior to the test being performed, accounts will be created for each role. Each account will be logged into and various authorized actions will be attempted. Next, the tester will attempt to take unauthorized actions such as accessing data outside of their role. Results of all of these actions will be recorded for mitigation.

## Authentication Logging

### Test Description

This test will ensure that authenticating to the system trigger the logging function to run.

### Control

This will be a manual test. Username and Password will be entered, and the authentication log will be checked for an entry.

### Inputs

A valid username and password will be entered. Upon logging in, the tester will check the authentication log.

### Outputs

We expect a log to be generated in the authentication log showing the user who authenticated, a timestamp for the attempt, as well as a success/fail status

### Procedures

The tester will attempt to login to the system with a valid username and password. Upon success, the authentication log will be checked for the correct entry.

## Logging Changes to Personnel Record

### Test Description

This test will ensure that any modification of a personnel record will trigger a log to be generated.

### Control

This will be a manual test. A personnel record will be created, modified, and deleted by the tester.

### Inputs

Credentials for the Records Admin role will be needed as inputs. A personnel record will be created with all necessary fields.

### Outputs

We expect three logs to be generated. One upon record creation, one upon modification, and one upon deletion.

### Procedures

Tester will log in as the Records Admin. Next, the tester will attempt to create a new personnel record. After creation, the tester will modify any field in the record. After the modification, the tester will delete the personnel records. Upon successful deletion, tester will navigate to the records log and check that the three logs were generated correctly for each action taken by the tester. Results will be recorded.

## Test Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Id | Test Name | Inputs | Expected Output | Actual Output | Pass/Fail |
| 7.1 | User login (valid) | Valid username and password | Successful login and redirection to user’s dashboard | Output coincided with expectations | Pass |
| 7.2 | User login (invalid) | Invalid username and password | Incorrect credentials warning and remain at login screen | Output coincided with expectations | Pass |
| 7.3 | Unique username | Duplicate username at user creation | “User already exists” warning | Output coincided with expectations | Pass |
| 7.4 | Verification of privilege | Credentials for each user type | No warnings or system prevention while performing authorized user functions | Output coincided with expectations | Pass |
| 7.5 | Authentication Logging | Valid credentials | Log entry generated in authentication log | Output coincided with expectations | Pass |
| 7.6 | Changes to personnel record modification logging | Records admin credentials and test data for modifications | Three logs to be generated. One upon record creation, one upon modification, and one upon deletion. | Output coincided with expectations | Pass |

**Design Document**

# INTRODUCTION

# Purpose

The purpose of this document is to detail the design and architecture of the Uncommon Solutions HR Management System. The Uncommon Solutions HR Management software is being developed using an Agile SDLC framework. While this document describes the design aspects of certain features within the Uncommon Solutions HR Management System, specific details about the requirements and design considerations that were made to meet those requirements can be found in the system requirement specifications document. This document contains the necessary information required to effectively define the architecture and system design to give the project team clearly defined guidance on the architecture of the system to be developed.

## Background

The Uncommon Solutions HR Management System will be designed in a way that makes it easy to support multiple platforms such as Windows, macOS, iOS and Android. This web-based tool provides a direct method for storing and providing access to individual personnel records, and for all processes required for HR tracking and data aggregation requirements. The HR system will be implemented using AWS Elastic Compute Cloud (EC2) and Amazon’s Relational Database Service (RDS) in order to allow for universal deployability and access.

## Scope

This document describes the overarching design and guiding principles of the Uncommon Solutions HR Management System. This is a living document and will be updated as changes are made to the HR Management Systems design.

## Assumptions

The following assumptions are relevant to the design of the proposed system:

* The proposed new system will leverage the Uncommon Solutions HR architecture.
* The existing architecture and system design will be used including all existing components and sub-systems.
* It is assumed that additional functionality will be added to the proposed solution as required during development and testing.

## Constraints

* There are no hardware or software technical constraints identified with this project.
* System interoperability may be a constraint since the design will leverage free tier AWS EC2 instance and RDS with the potential to expand to paid utilization at a larger-scale fielding.

## Risks

There are very minimal risks associated with the system design.  This is primarily because the existing system design and architecture will not be modified to meet the needs of the proposed solution. Ongoing maintenance of the system will also be a concern.

## Design Considerations

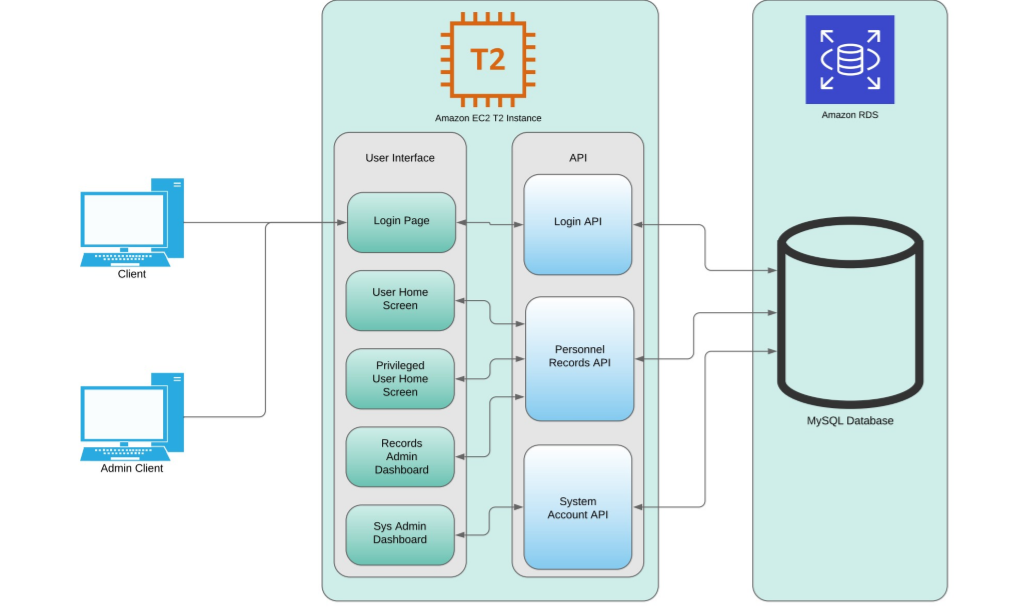
The major design considerations for the proposed solution are related to system performance and scalability of the solution. The data center is hosted in AWS which provides a tremendous amount of flexibility in terms of scaling for performance and storage requirements. Processor speed, memory, peripherals, and stakeholder support will be factored in the design.

# SYSTEM OVERVIEW

## System Design

The product system design utilizes two different AWS services, Elastic Compute Cloud (EC2) and Amazon’s Relational Database Service (RDS). Using AWS as a platform for fielding the application is the most cost-effective way to host the application while also providing access to the application for all project team members and product owners. AWS is an excellent platform for quickly spinning up application prototypes and rapidly deploying solutions to multiple customers.

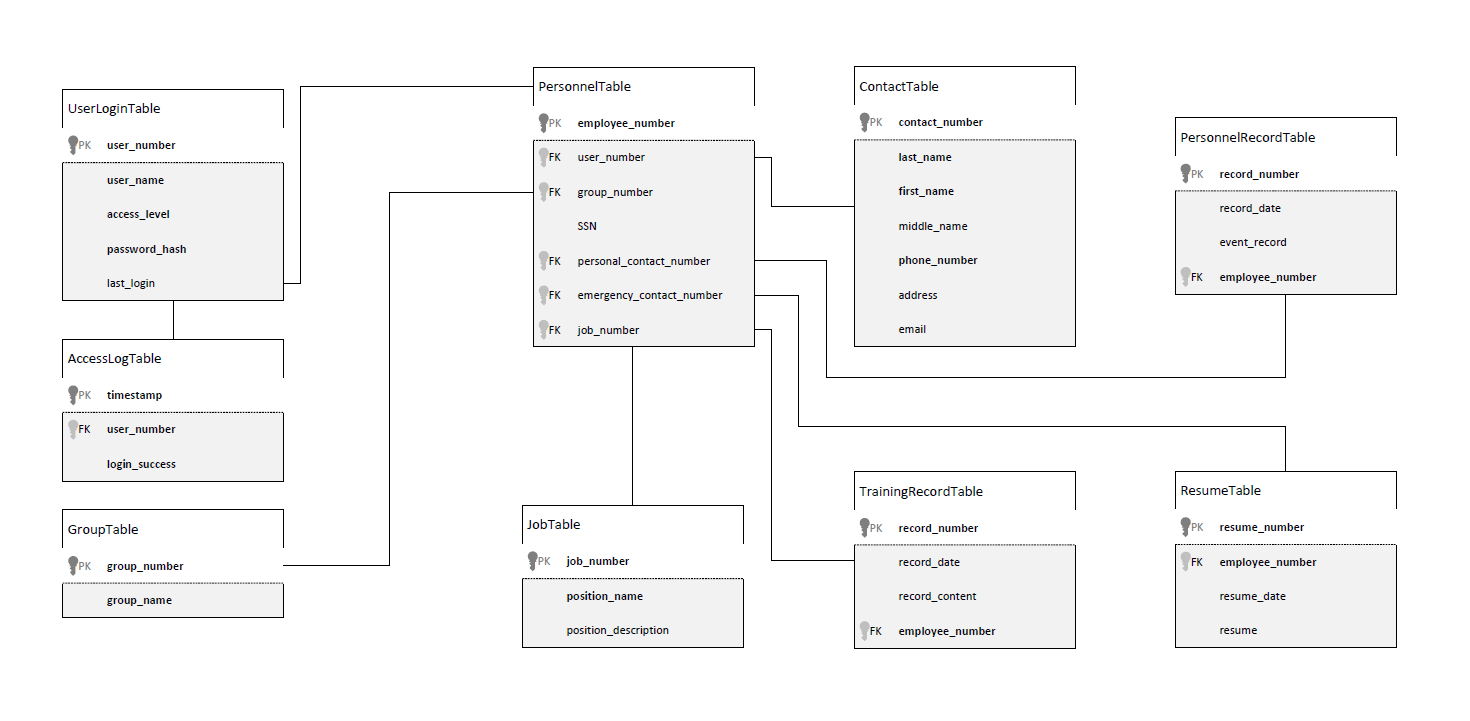
The application uses Apache version 2.4.29 as a web server and is installed on the EC2 T2 instance and uses Ubuntu 18.04 as an operating system. All source code including the user interface and API will be on the apache server. The application uses a MySQL database installed on an RDS instance to house all the application data. To access the application the client or admin client simply connects to the EC2 instance currently located at “**ec2-3-81-54-213.compute-1.amazonaws.com**” OR “**3.81.54.213**”.



**Figure 1: The System Design Diagram**

# SYSTEM ARCHITECTURE DESIGN

## Database Design



**Figure 2: The Data Design Diagram (DDD)**

The database design for this project is divided into two main sections; authentication data and personnel data. There is a relational connection between the two sections for most accounts, except those accounts for system administrators. The Data Design Diagram (DDD) provides a visual of the tables and their relationships, but a basic breakdown is described in the following sections.

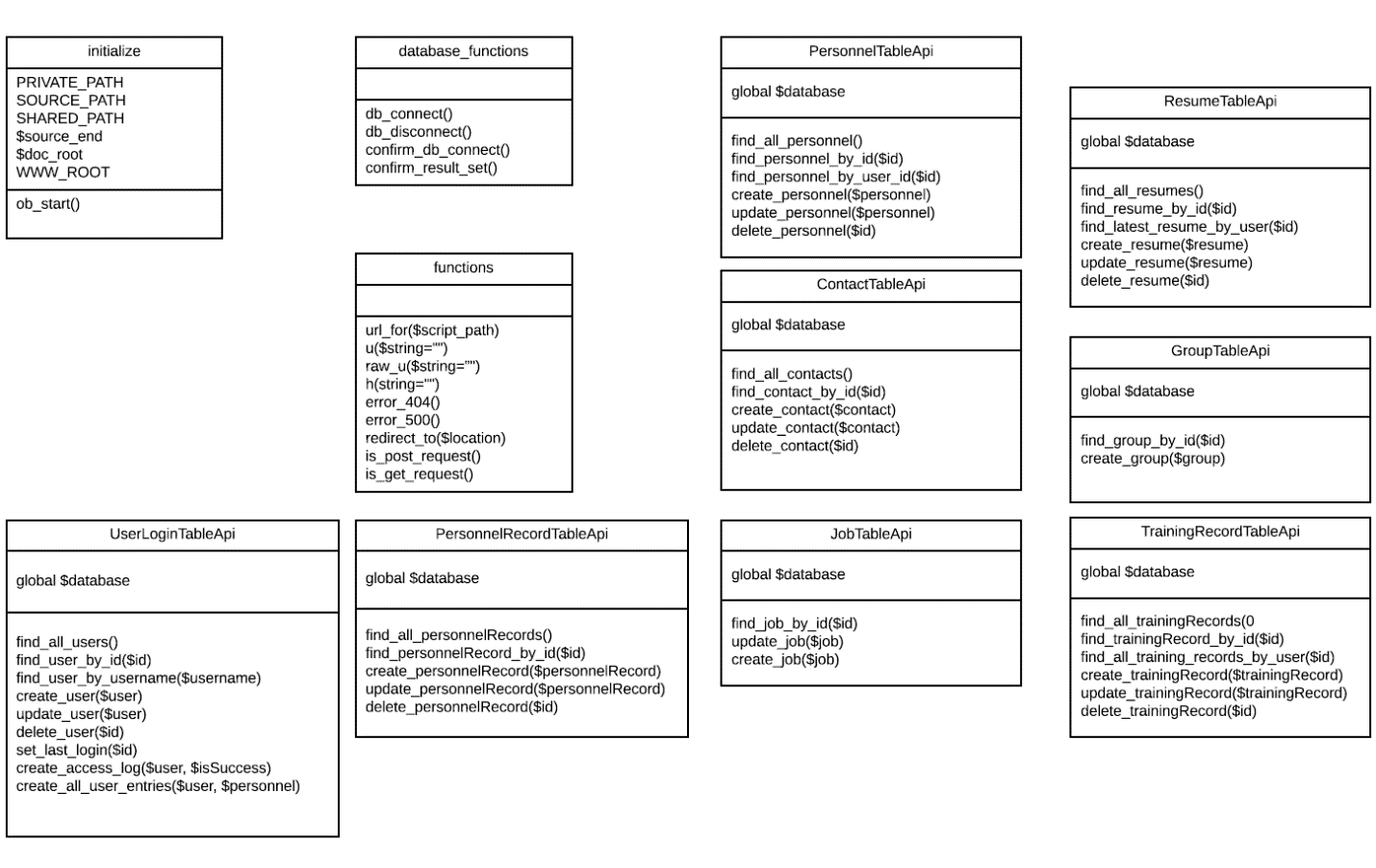
The authentication data section is simple, containing the table with information on the individual user accounts and a linked table to log authentication attempts in. The presence of the AccessLogTable allows for database level tracking of frequency of failed login attempts as well as an audit log of all successful system logins.

The personnel data section is a little more complicated than that of the authentication data, but only to allow for more flexibility in data storage for individual records. In much the same way, as personnel change jobs, the jobs themselves are still present and can be reassigned to a new personnel record to reflect the change in who is occupying that position without requiring the re-entry of the job title or job description. As individuals progress within the company, the number of training records, personnel records, and resumes on file can shift and grow with the individual, in order to effectively capture all their experience and capabilities. The same table is used to store contact information for both the employees and their emergency contacts, as we will have instances where an emergency contact is another employee of the company, and there’s no need to duplicate data in those cases.

Overall, it’s a simple database that takes advantage of the capabilities of the relational model to streamline data storage and avoid space allocation for empty records, while still allowing the flexibility for a large-scale employee record in the cases that warrant.

## Class Design

The design of the API uses a simple MVC pattern where the models use an Object-Oriented design and have standard attributes such as private properties that are accessible with getter and setter methods. Since the scope of the application is small only one controller will be used to handle all the database operations. Classes will be implemented to handle database connectivity, user login, and audit logging.



**Figure 3: The Class UML Diagram**

## User Interface

The User Interface for the will adopt a flat design. Not only would this provide the design with a modern aesthetic, but would also minimize the system resources consumed to render the web pages expanding the compatibile range of platforms. The design will be seamless between pages to ensure that there are no sudden shifts in design. Refer to the Appendix for page wireframes.

The User Interface presented after completion of the login process will depend on the level of access associated with the individual user account. The interface presented to a standard user will vary significantly from that presented to a system administrator as outlined in the wireframes presented in the appendix.

# SYSTEM INTERGRITY CONTROLS

The following security and integrity controls are relevant to the design of the proposed system:

* Source code for the HR software components and products will be stored securely with need-to-know access controls applied.
* The system will have log files for all modification that will be maintained and preserved for future analysis.
* User passwords will be encrypted in the database utilizing a secure hashing algorithm.
* Internal security will be implemented to restrict access of critical data items to only Data Administrators and users with the required access levels.
* Each employee will be restricted to only access their personnel data unless they have higher level accesses.
* Audit procedures will be implemented to meet control, reporting, and retention period requirements for operational and management reports
* Verification processes for additions, deletions, or updates of critical data
* The system will have the ability to identify all audit information by user identification, network terminal identification, date, time, and data accessed or changed.

# Software Requirements

## Requirements Overview

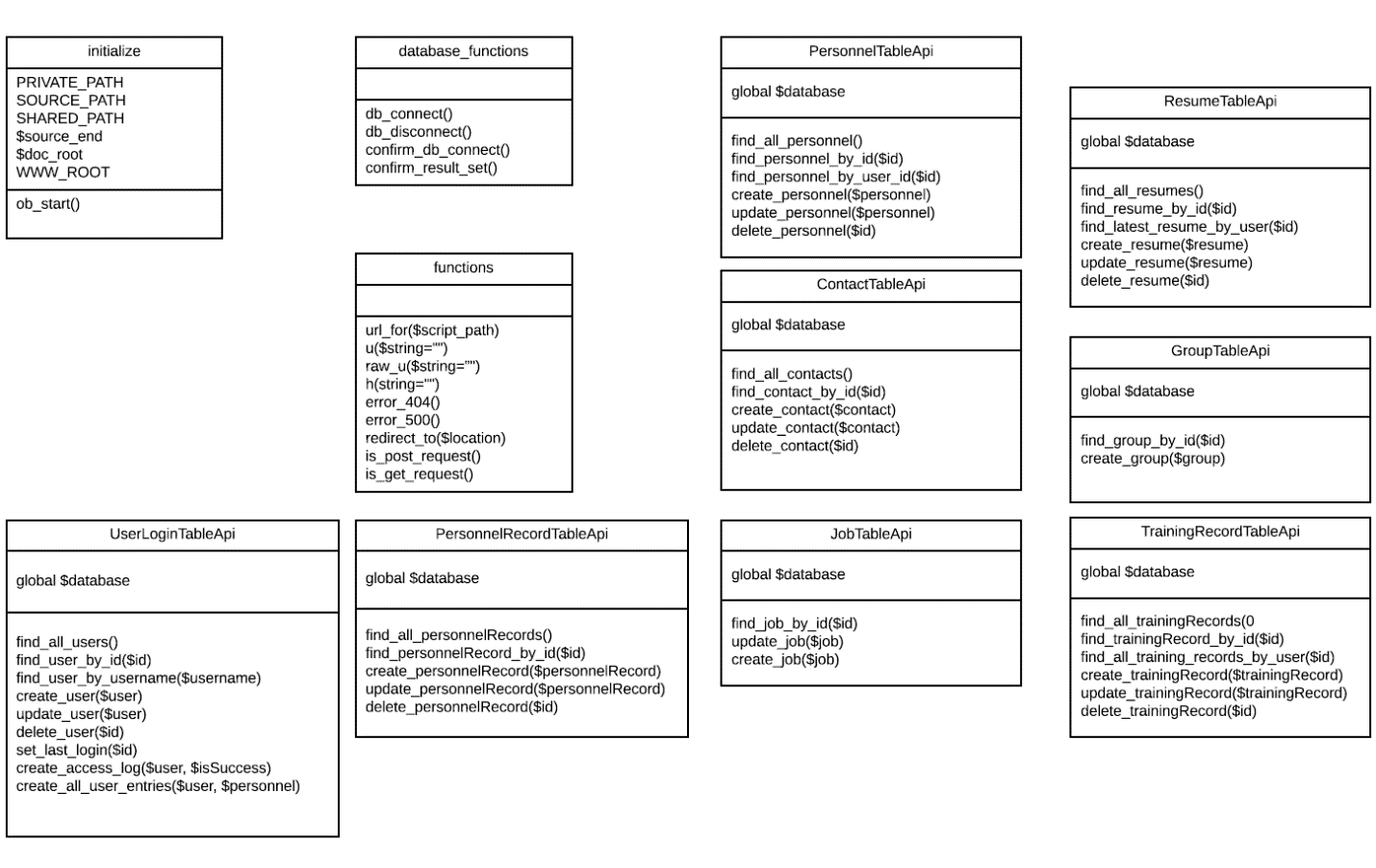
The system being designed will provide a username/password authenticated access to a system for the management of HR records. This will present in a simple interface for users to be able to view and modify some parts of their individual records. Privileged users will have additional accesses to manage more data for other users as a part of their role within the system. Management of the accounts associated with the system and their access levels will be controlled through a separated account administration interface which requires an account with that privilege level but no database level access, ensuring separation of access controls.

## Product Features

The delivered software product will provide API interfaces invisible to the end user for access of data stored in the system associated database. This will present to the end user via a simple to operate interface that presents all relevant data in an easy to read and well-structured format. For accounts with higher privilege levels, the current privilege level will be easy to identify within the interface and present in a cleanly structured format for ease of account and personnel records management. All deletions and modifications will present with a confirmation dialogue that clearly states the action about to be executed.

## User Classes and Characteristics

The design of the API uses a simple MVC pattern where the models use an Object-Oriented design and have standard attributes such as private properties that are accessible with getter and setter methods. Since the scope of the application is small only one controller will be used to handle all the database operations. Classes will be implemented to handle database connectivity, user login, and audit logging.



**Figure 1: The Class UML Diagram**

## Design and Implementation Constraints

Software will be designed and implemented to be platform agnostic and function to connect to the database through a standard SQL interface based on an external configuration file. This will allow for customer fielding of a database on whatever platform they choose, requiring only industry standard access methods.

## Assumptions and Dependencies

It is assumed that the underlying system will provide the required supports as defined in the System Requirements document. All software will be designed and implemented to ensure maximum clarity.

## User Interfaces

The User Interface for the will adopt a flat design. Not only would this provide the design with a modern aesthetic, but would also minimize the system resources consumed to render the web pages expanding the compatibile range of platforms. The design will be seamless between pages to ensure that there are no sudden shifts in design. Refer to the Appendix for page wireframes.

The User Interface presented after completion of the login process will depend on the level of access associated with the individual user account. The interface presented to a standard user will vary significantly from that presented to a system administrator as outlined in the wireframes presented in the appendix.

## Software Quality Attributes

All software design will adhere to industry standards for modularity, programming structure, algorithm efficiency, object-oriented design and clear and understandable in-source documentation (commenting). Known security vulnerabilities will be protected against and the software will be built in such a way as to ensure that future optimizations, security fixes, and expansions will be able to be implemented without additional effort to understand existing source code.

**Development History**

# INTRODUCTION

# Purpose

The purpose of this document is to detail the implementation process of the Uncommon Solutions HR Management System. The Uncommon Solutions HR Management System is being developed using an Agile SDLC framework. Any deviations from the planned Uncommon Solutions HR Management System will be reflected by updated changes to the associated design and system management documents. This document contains the necessary information required to effectively capture the development efforts of the team.

## Background

The Uncommon Solutions HR Management System will be designed in a way that makes it easy to support multiple platforms such as Windows, macOS, iOS and Android. This web-based tool provides a direct method for storing and providing access to individual personnel records, and for all processes required for HR tracking and data aggregation requirements. The HR system will be implemented using AWS Elastic Compute Cloud (EC2) and Amazon’s Relational Database Service (RDS) in order to allow for universal deployability and access.

## Scope

This document describes the development progress of the Uncommon Solutions HR Management System during each of the phases. This is a living document and will be updated as changes are made to the HR Management Systems design.

## Phases

The planned implementation of this project is broken down into three phases as follows:

1. Phase 1 consists of the creation of the database structure for information storage and the generation of the UI panels.
2. Phase 2 will consist of the functionality behind the login screen to include session management for the program. Additionally, user administration function to allow for management of user accounts will be implemented in this phase.
3. Phase 3 will consist of the data management functionality associated with this HR management system for the entry, modification, and management of the personnel information for the company.

## Schedule

The phased development schedule is a three-week process running from 18 November 2019 to 8 December 2019. There is an additional one-week flex time to allow for any schedule overruns and to allow for additional functionality to be added if time allows. This flex week runs from 9-15 December 2019.

## Design Considerations

The design for this program is as described in the Uncommon Solutions HR Management System Design Document. Any design variations will be validated by all members of the development team and incorporated into all design documentation to ensure that the entire development process is captured in documentation.

# PHASE 1

## Development Progress

In accordance with the planned steps for Phase 1 development, the initial process of creating the database structure independently from the creation of the initial user UI was executed. The individual talents of team members were employed at this phase in order to best make use of developmental strengths and deliver a solid foundation for later development process. The goals for Phase 1 were to create and implement the initial database structure and design and to build the initial program containing the User Interface for the system.

## Database Development

The database has been instantiated based on the design outlined in the original Design Document. There were some minor changes to the table and field names based on requirements unearthed during Phase 1 development. The updated Data Design Diagram is below, and has been updated in the Design Document.

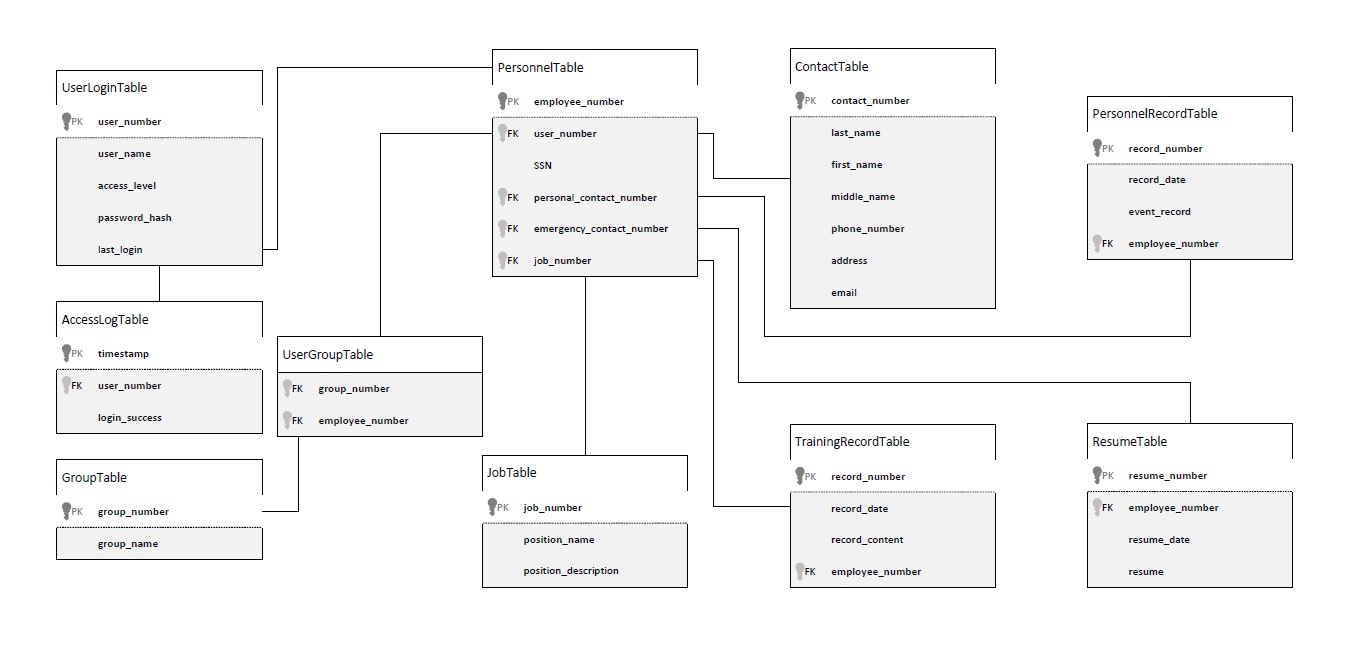


Figure 1: Data Design Diagram

The described tables had SQL statements built to create them and the foreign key dependencies defined by the Data Design Diagram. This SQL was executed in the instantiated Amazon RDS instance creating the baseline database structure for the Uncommon Solutions HR Management System to utilize. The below image shows the connected database after the SQL has been run and the tables are instantiated.

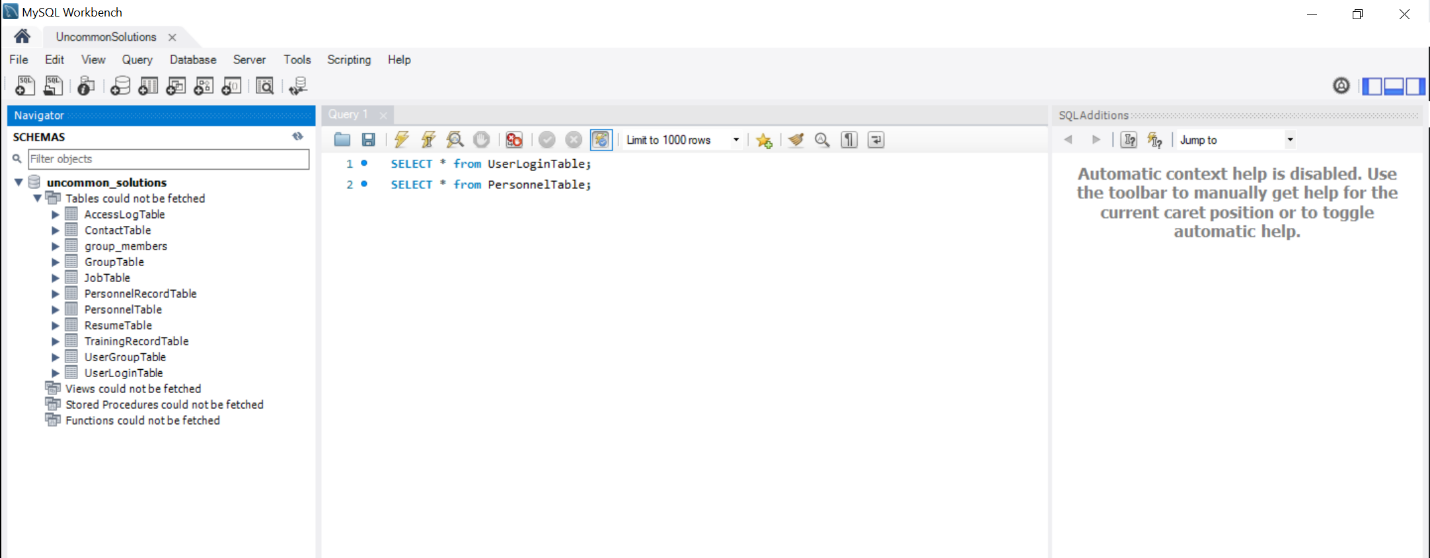


Figure 2: Database Instantiation

## Program Development

Initial development of the user interface systems was executed within PHP as a best option for the execution of a web-based interface system. With that in mind the initial interface design was created to show the login screen, user management console, and user data screen. In Phase 2, the first two of these will have their database connection instantiated and functionality created.

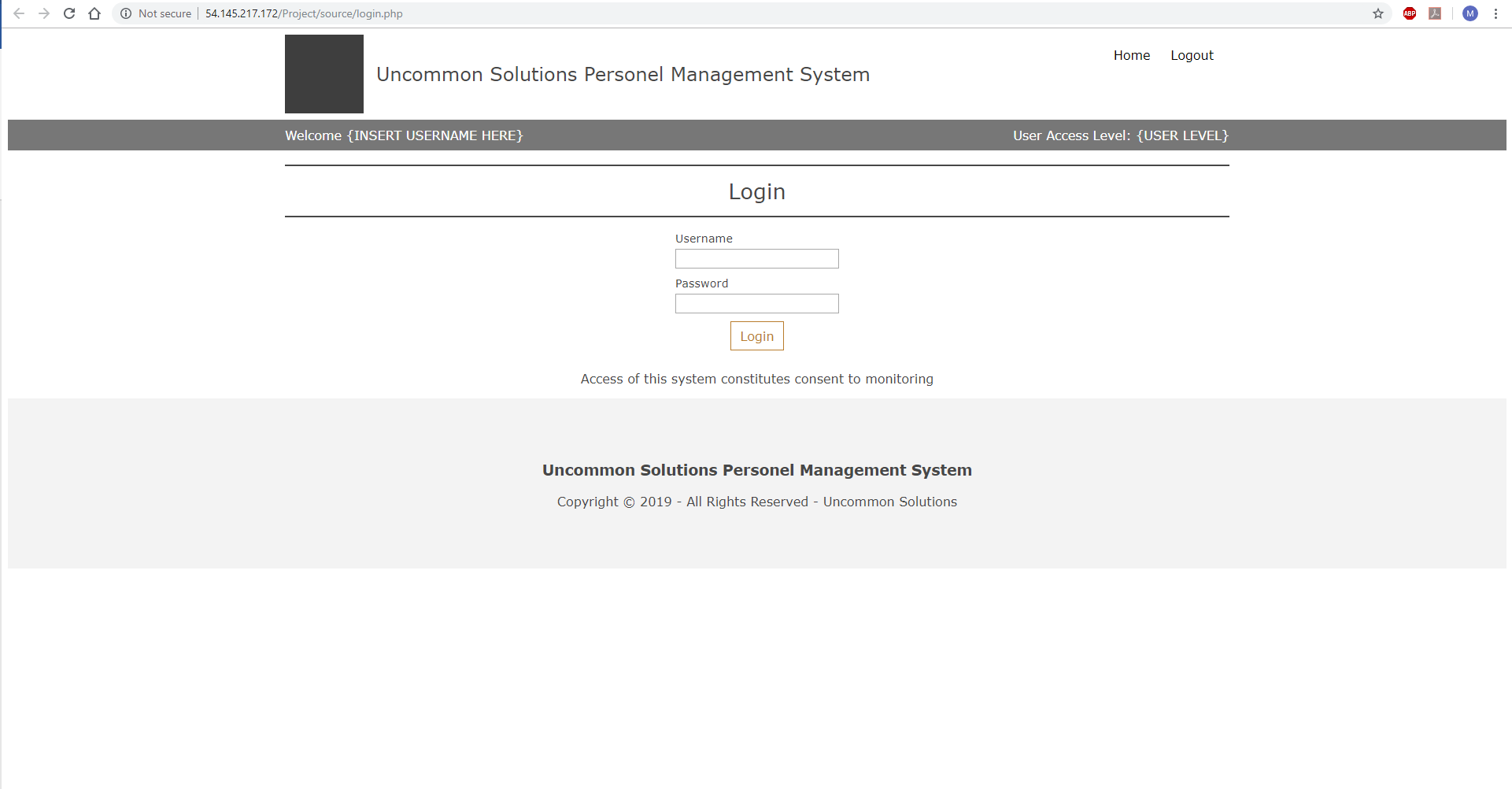


Figure 3: Login Screen

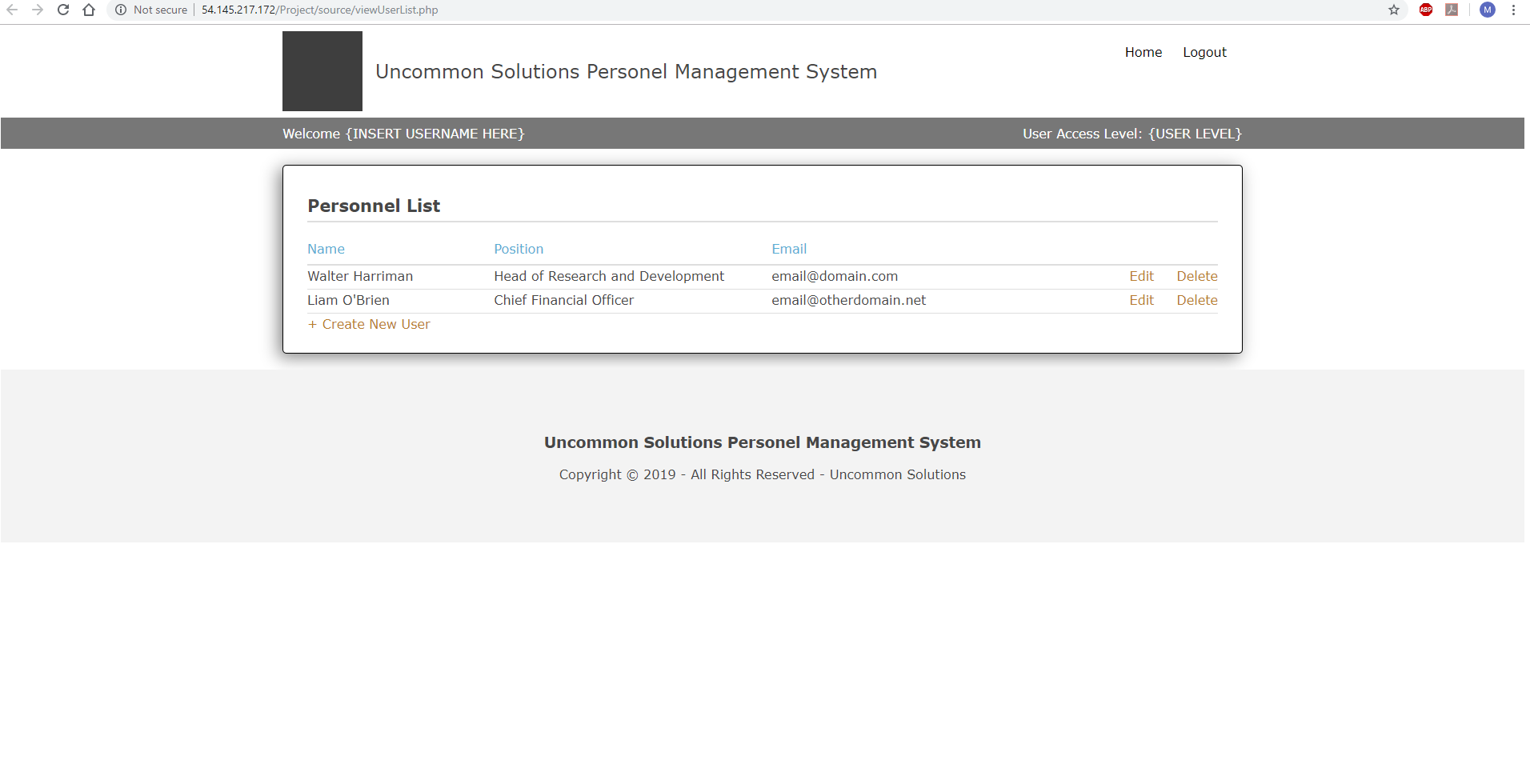


Figure 4: User Management Console

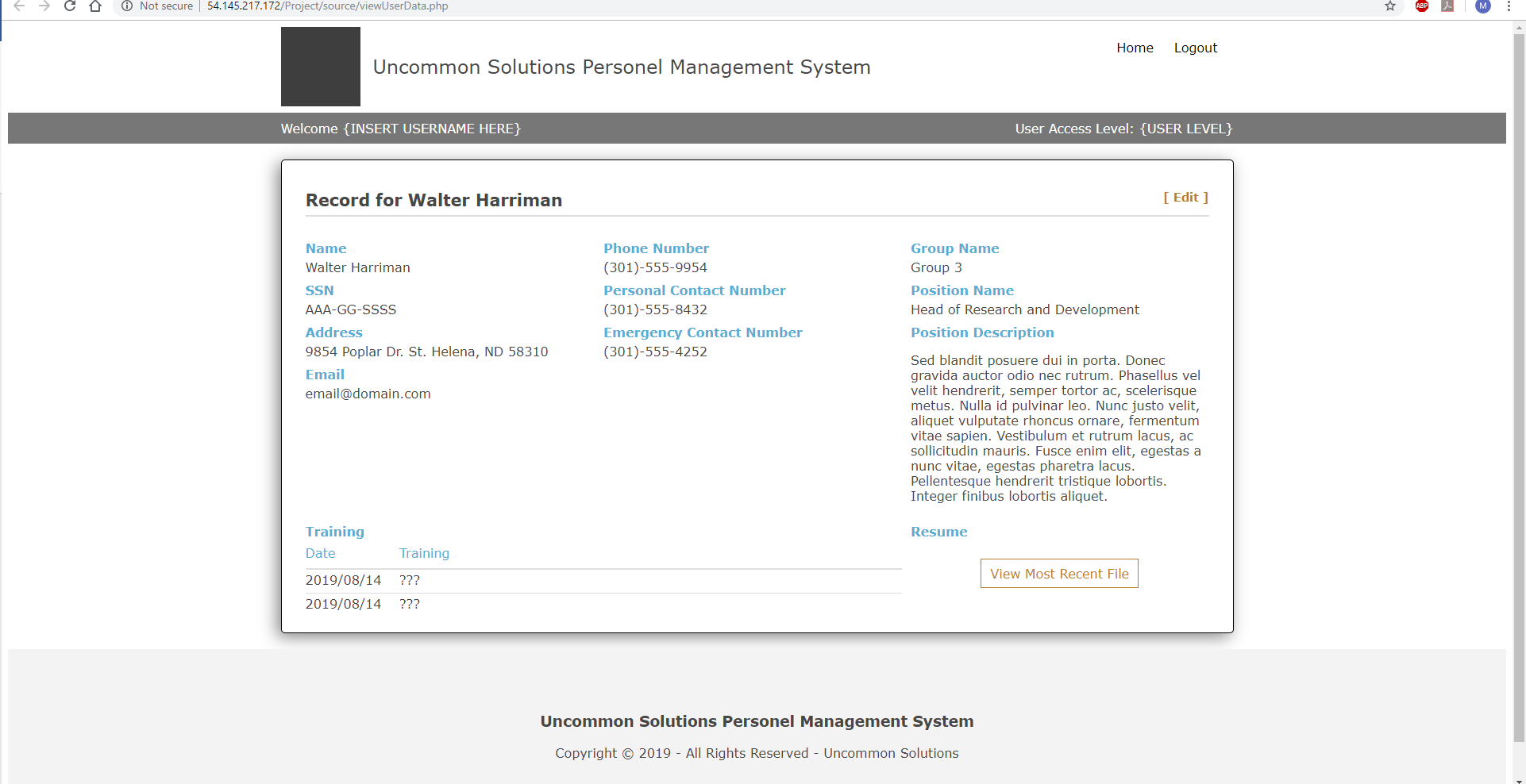


Figure 5: User Data Screen

## Milestone Evaluation

All milestone objectives for Phase 1 development were met by 23 November 2019. This was slightly ahead of projected scheduled development. As there is a major holiday occurring toward the end of Phase 2 development, Phase 2 was entered early to ensure that development would remain on track for the program.

**PHASE 2**

## Development Progress

In accordance with the planned steps for Phase 2 development, the interface for user login functionality was completed to include session management for the user interface, this includes database level logging of all login attempts. In addition, the functionality for user account administration was implemented and validated to function. The individual talents of team members were employed at this phase in order to best make use of developmental strengths and deliver a solid foundation for later development process. The goals for Phase 2 were to build and demonstrate the functionality behind the login screen to include session management for the program. Additionally, user administration functions to allow for management of user accounts were to be implemented in this phase.

## Database Development

Some adjustments to the database and the associated table creation scripts were executed during this phase. The creation script includes checking for table existence prior to attempting cleanup, in addition there were some minor corrections in database column naming to ensure proper reflection of the planned schema. Auto-increment was added to the primary keys for all tables in order to ensure clean records addition and automatic iteration of the numeric primary keys on row addition. Finally, the database was simplified slightly by the removal of the UserGroupTable from the schema, this restricts Personnel records to belong to a single group, but greatly simplifies the initial implementation of the API for Personnel record interface. This modified table creation script was executed on the database to ensure clean interaction with the implemented API.

## Program Development

Development in this phase was targeted more at the functionality behind the user interface developed on Phase 1. The initial focus was on a correct implementation of a secure login method to include session management and database level logging of all attempted accesses. Feedback in the form of system messages to the end user were implemented in order to show system functionality for invalid login attempts.

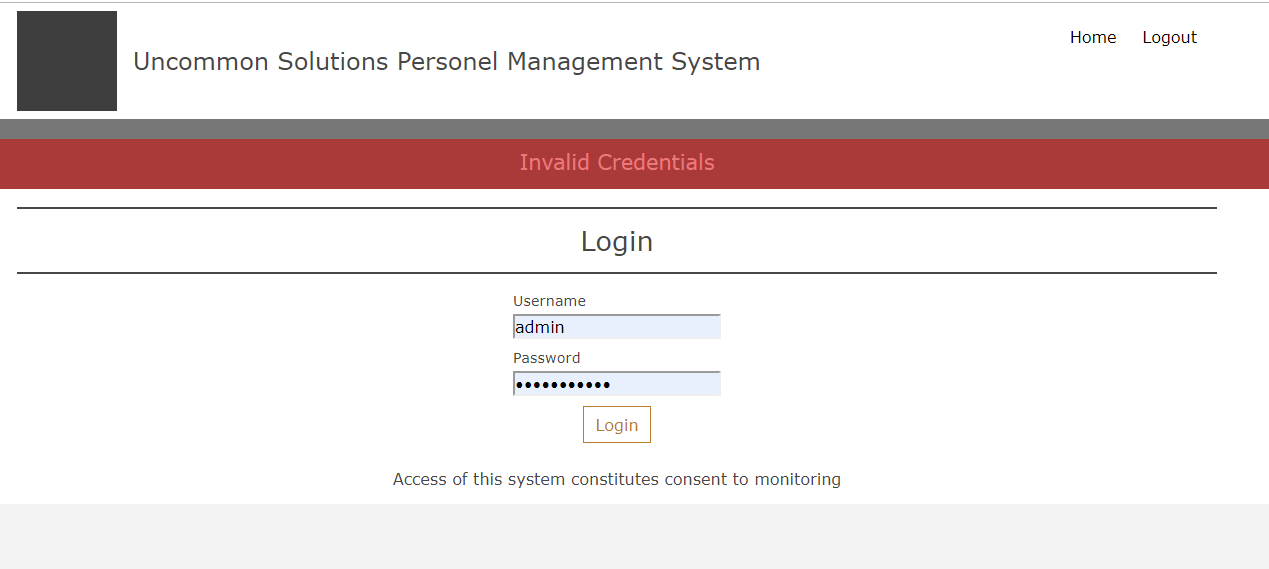
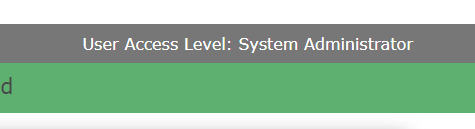
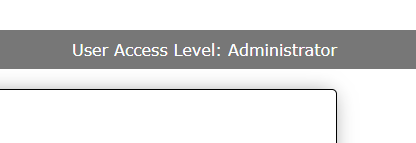


Figure 6: Invalid Login Credentials

As a part of the session management implementation, user-level session visibility on available functionality was added based on the defined access level of the logged in user account. This is reflected in what parts of the UI are displayed depending on access level, such that users will not even be aware of functionality that they do not have access to based on their assigned access level.







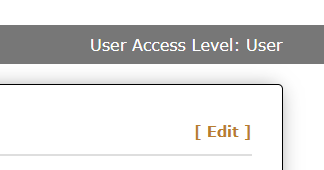


Figure 7: Access-Level Based Display

For the account administration panel, functionality for the creation, modification and deletion of user accounts was added for accounts with the system administration access level. This UI is not available or functional without an active session with system administration capabilities. This includes UI and functionality for the described functions and feedback messages on the success or failure of the attempted actions.

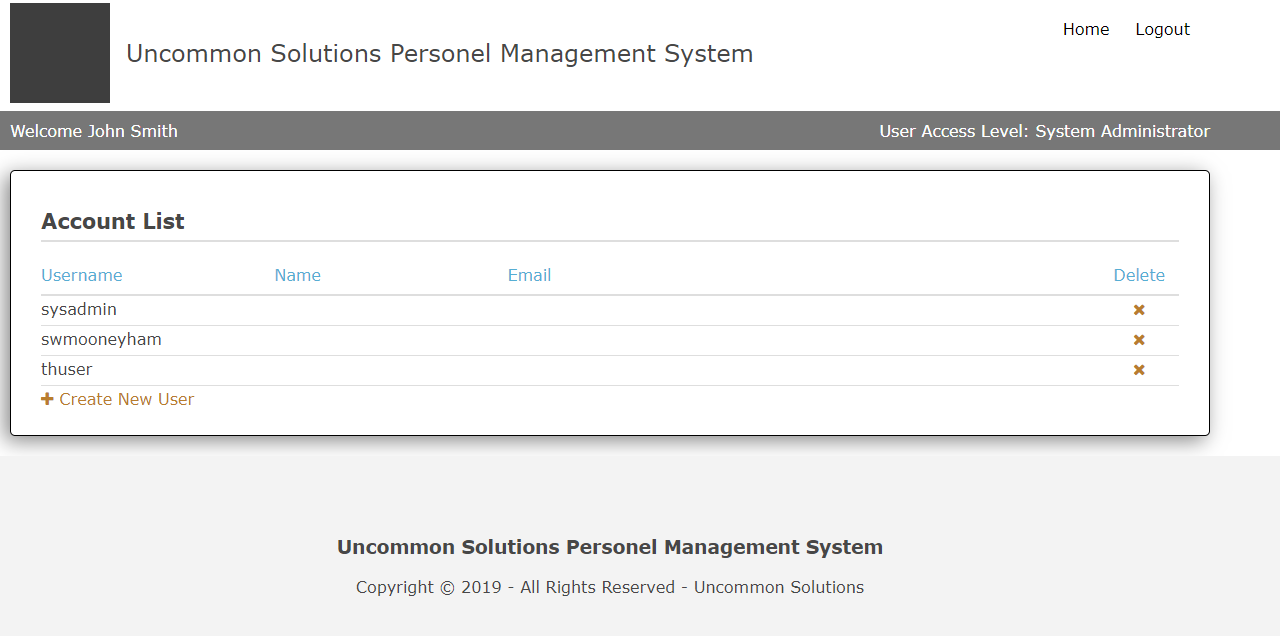


Figure 8: User Administration Panel

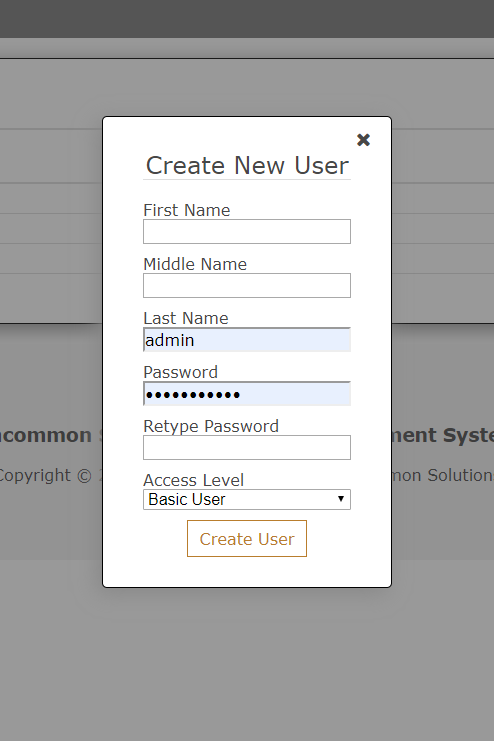


Figure 9: Account Creation

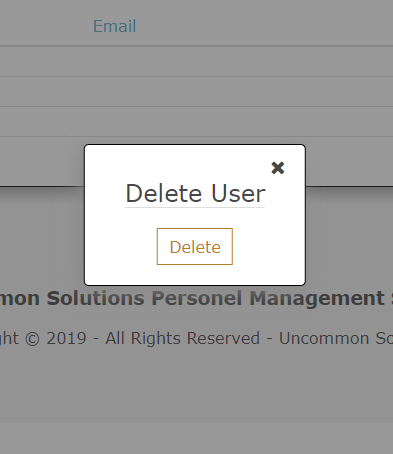


Figure 10: Account Deletion

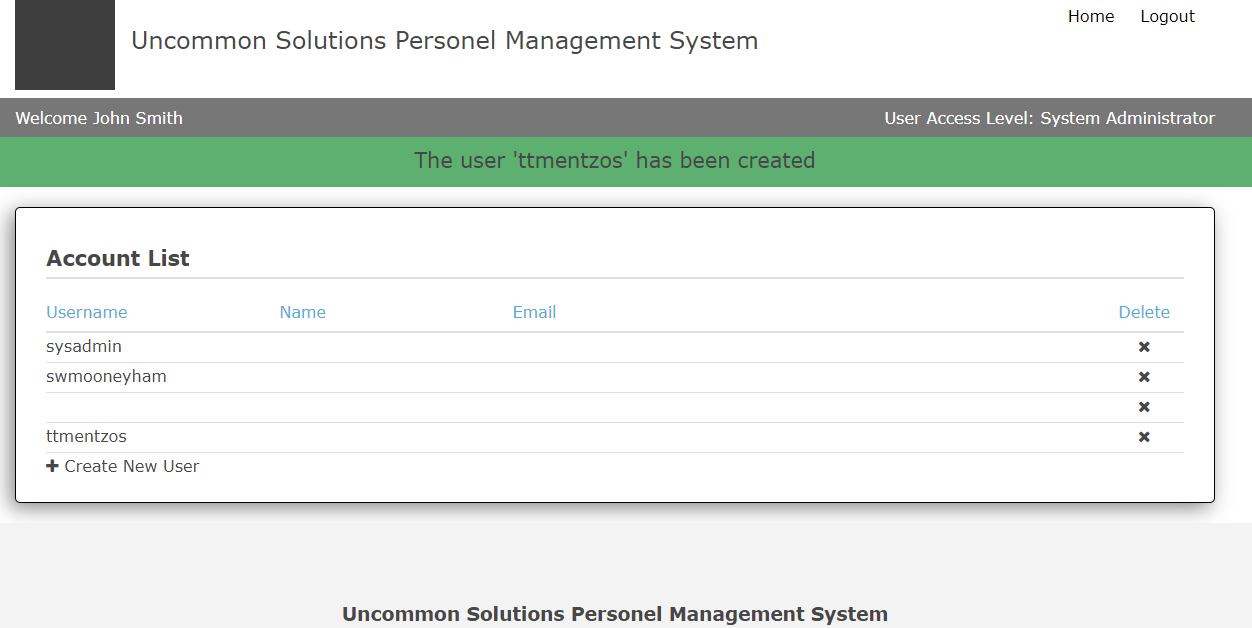


Figure 11: User Feedback Messaging

## Milestone Evaluation

All milestones for Phase 2 implementation have been validated as completed and met by 29 November 2019. At that time there was a break in implementation efforts to regroup, and development on Phase 3 will commence on 1 December 2019. The project remains on schedule for the planned implementation and should be able to enter into execution of the test plan on 8 December 2019 or sooner.

**PHASE 3**

## Development Progress

Development for Phase 3 was focused on the UI and APIs for interaction with the employee HR records. With the previous phases having delivered database functionality and the system controls required for our account management and login processes, this was a logical expansion toward the data management for the rest of the system. Our targeted development objectives for Phase 3 are the creation and implementation of database access APIs for HR records data followed by the utilization of those APIs for addition, modification, and deletion of employee HR records by accounts with the appropriate access levels. The test plan was expanded to include a validation matrix, which was applied after the completion of development.

## Database Development

No database modifications were required at this stage of development, the changes made in Phase 2 were the only changes needed for smooth implementation. There were additional user accounts and data provided to the HR database segments to allow for validation and demonstration of functionality as those APIs/UI were built and tested.

## Program Development

This phase was begun with the development of the APIs for access to the HR records data associated with the specific users of the system. There is little to illustrate with these APIs, but their success and utilization will be apparent later in this narrative in the look at integration with the designed UI.

At this stage of development, we were already complete with user management and the system administrator access for account management:

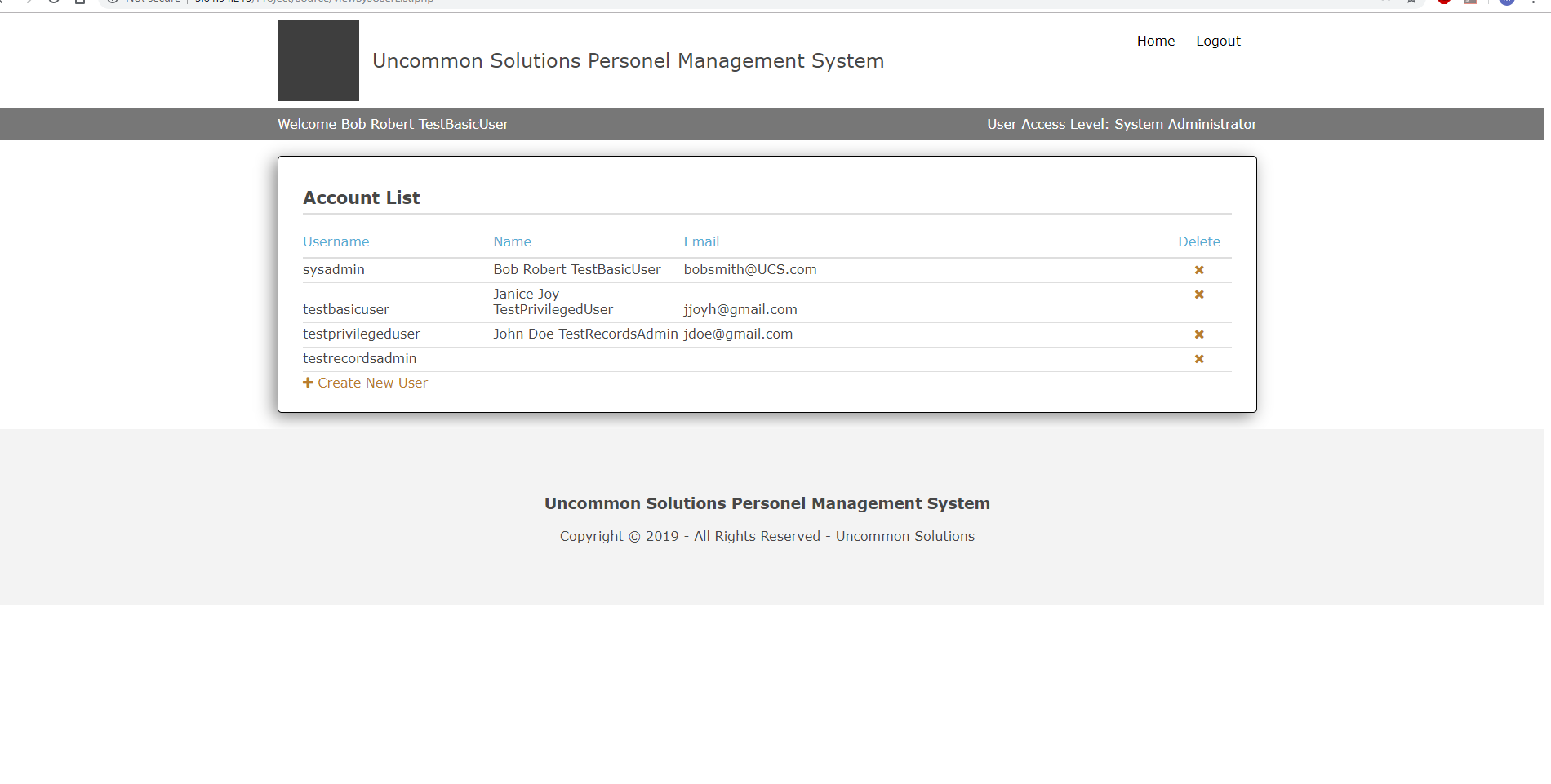


Figure 12: System Administrator Home Screen

Next for development was the integration of functionality for individual users, privileged users, and records administrators as described in the system design. The functional APIs were integrated to previously created framework, and additional framework and logic was created to display individual records as requested, depending on the access level of the individual utilizing the system. The displayed home screen was different depending on the access level of the individual user on the system, as illustrated below:

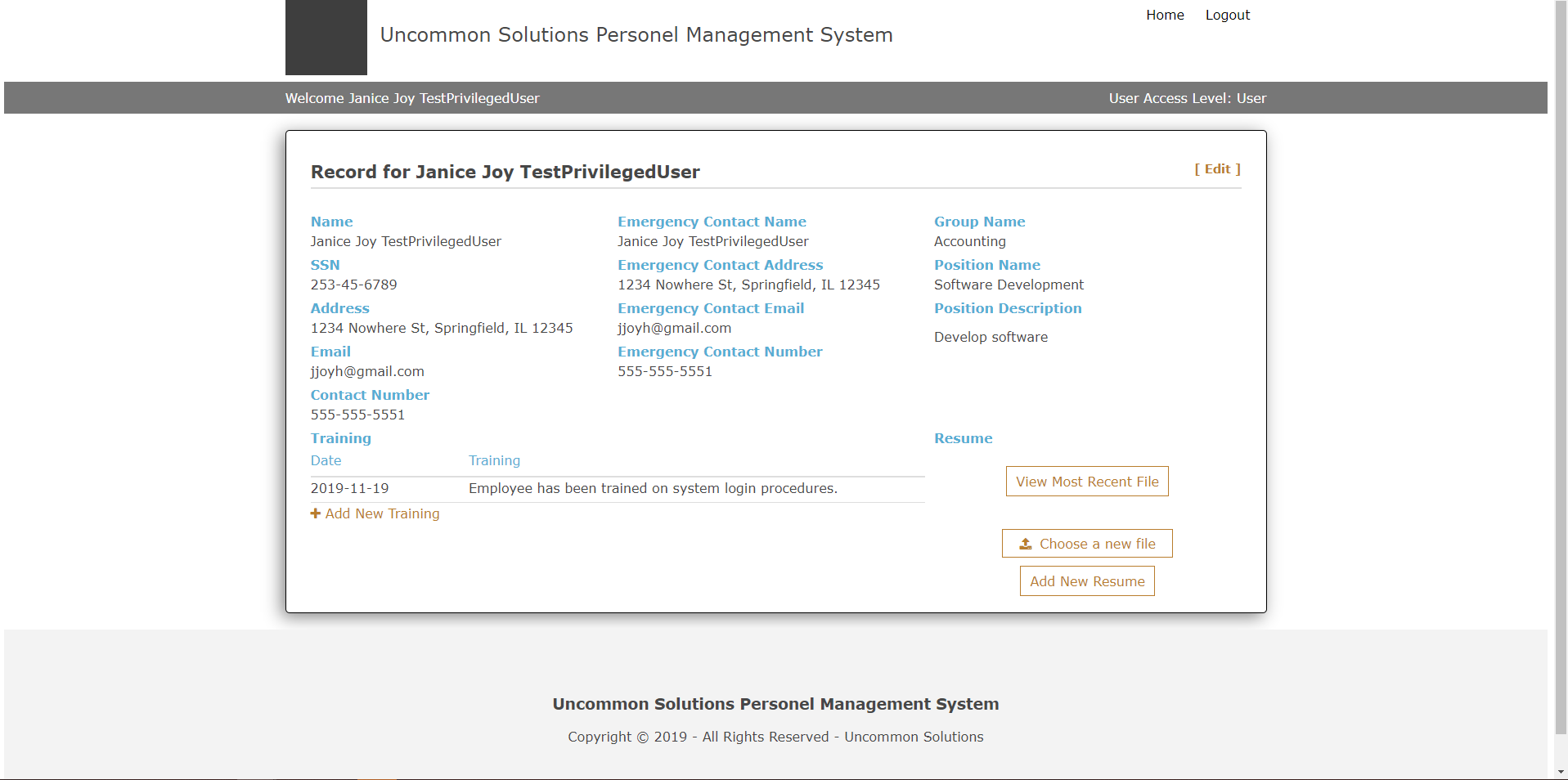


Figure 13: Standard User Home Screen

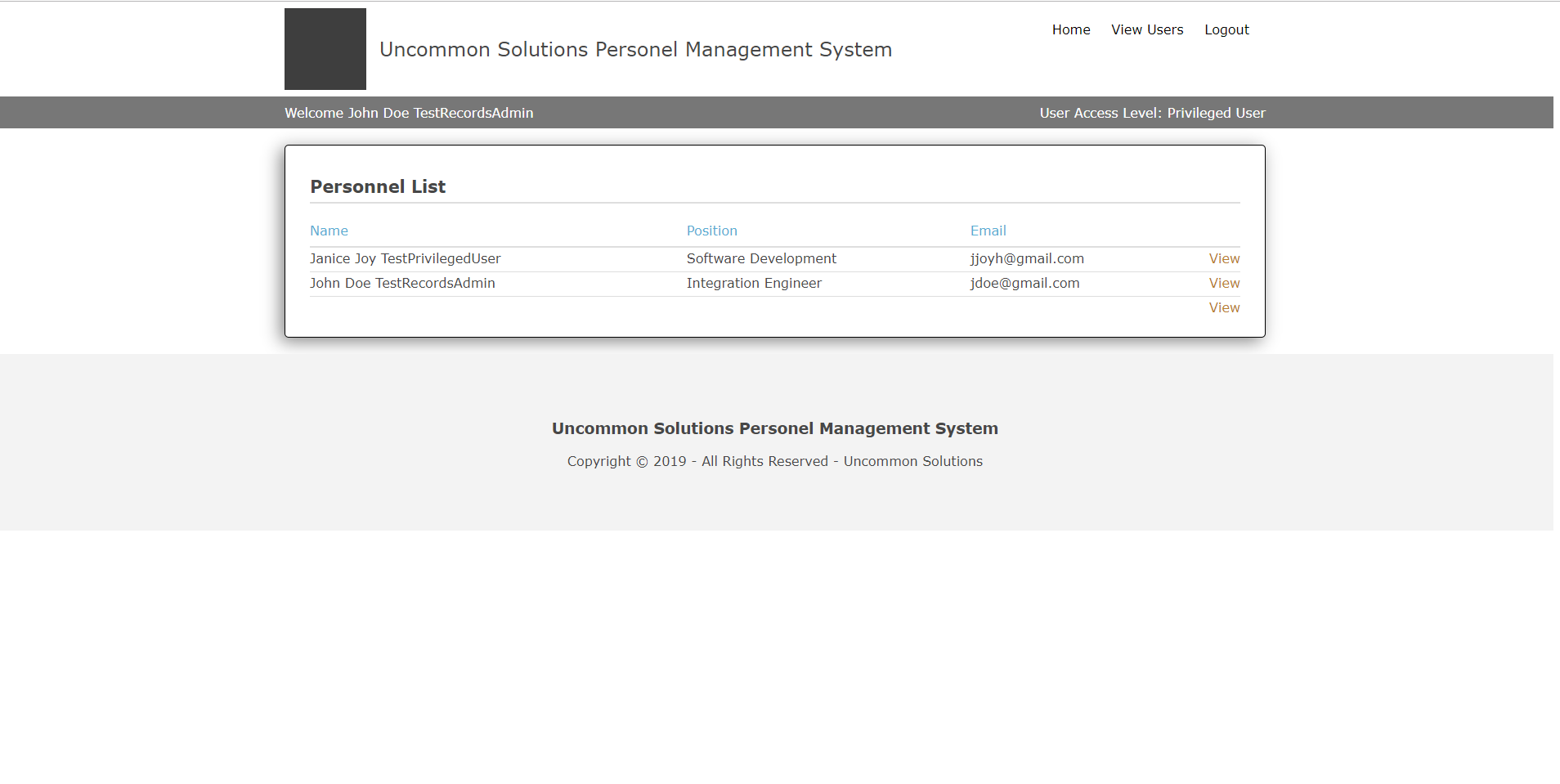


Figure 14: Privileged User Home Screen

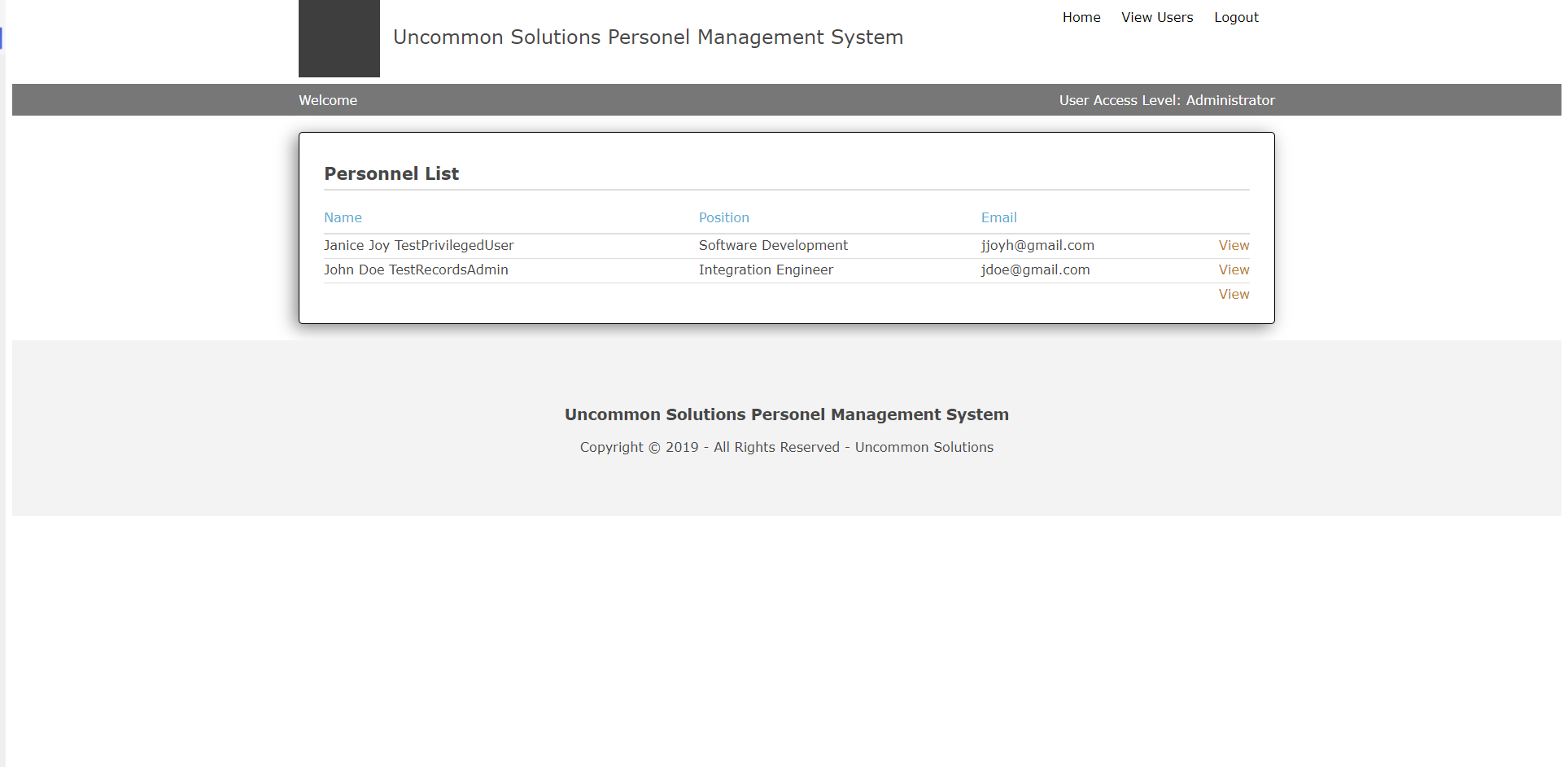


Figure 15: Records Administrator Home Screen

As can be seen, the functionality presented for our higher-level users allows access to more than just their own information, and by making use of the ability to view and modify records other than your own, the Records Administrator level has full access to do so:

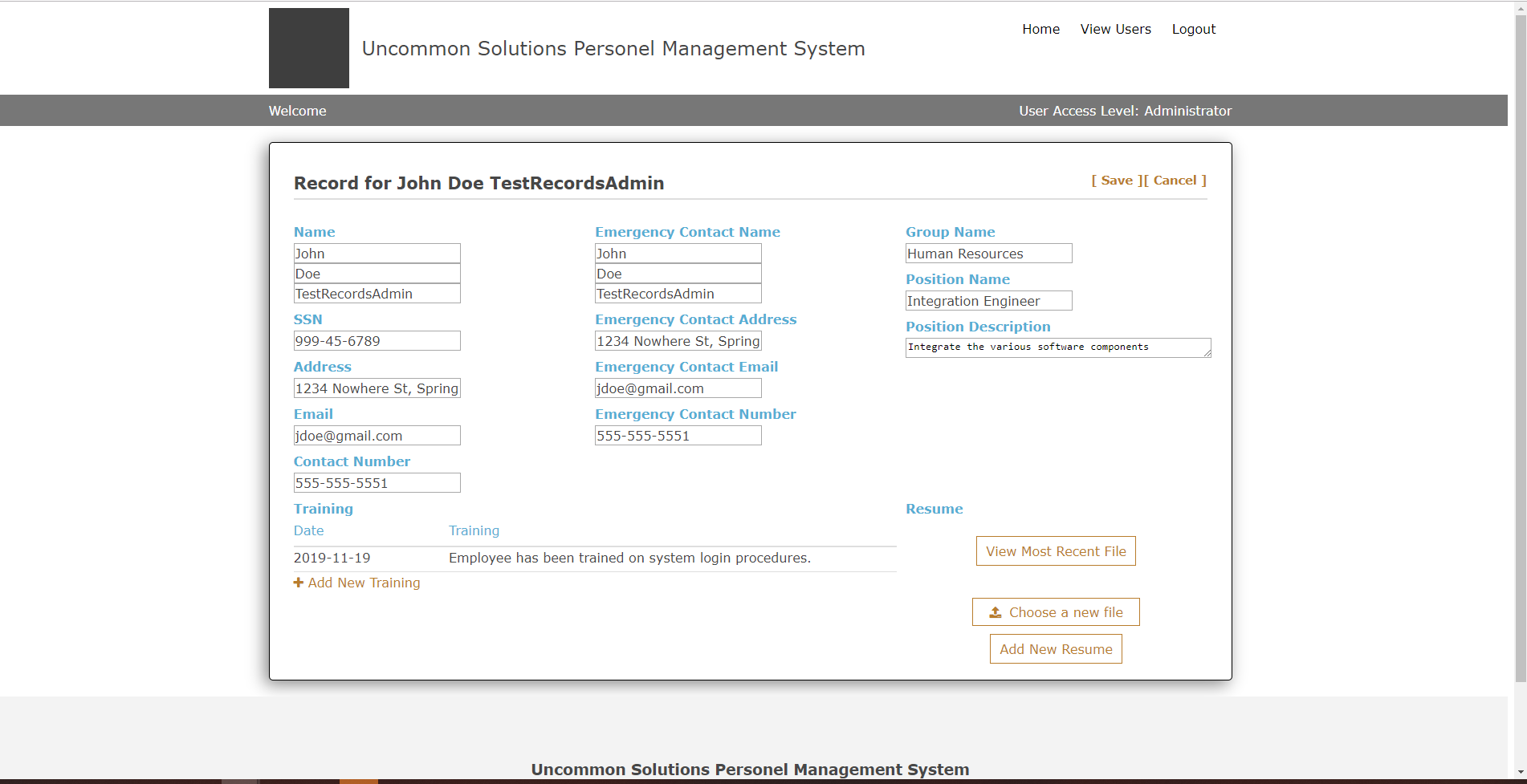


Figure 16: Records Administrator Other User Record

These looks at the logged-in home screens of each user level also serve to illustrate the previously implemented visual cue of current access level implemented in Phase 2 and show that the functionality has been carried across to all screens of the system.

After the completion of the development milestones required for Phase 3, systems level testing was executed by our test manager and documented appropriately. The updated Test Plan is included with the submission for Phase 3 to illustrate. On a final note, the testing by multiple parties has validated the correct function of audit logging as can be seen by the growing number of entries present in that function.

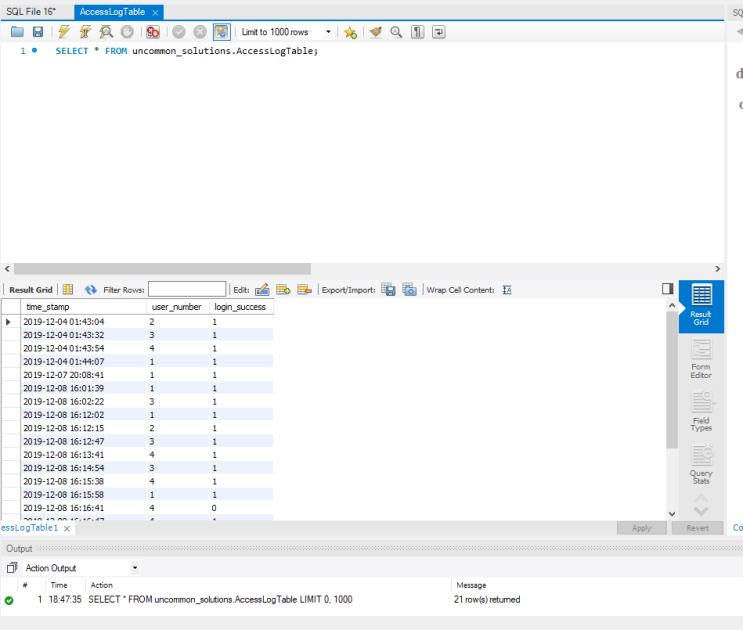


Figure 17: Audit Log

## Milestone Evaluation

Overall Phase 3 was executed successfully, however development timelines shifted slightly from the original plan and work was not completed until day six of the week cycle. This fell within the flex time built into each week of development, but slightly compressed the available time for documentation of development progress. Communication was a bit slower, but that was to be expected in the week following a holiday, however all members completed their assigned work for the Phase and it was completed. There are some minor technical issues in the current implementation which have been identified during group function validation and execution of the test plan, these will be resolved early next week inside of our “flex time” in the schedule prior to final delivery. Execution of the test plan revealed a need for individualized testing of specific segments of the site, which will be added to the final test plan to be re-evaluated after the completion of development fixes.

We expect the final version and associated documents to be ready for submission by Friday, 13 December in accordance with the current development schedule. As usual, this allows for additional flex of the remaining two days of the term, but as a number of us will be attending commencement on 14 December, the goal is to have all work complete prior to that date.

**Conclusions**

# INTRODUCTION

# Purpose

The purpose of this document is to reflect on the design and implementation process of the Uncommon Solutions HR Management System. The Uncommon Solutions HR Management System was developed using an Agile SDLC framework. This document contains the team’s perspective on the design and development process and thoughts for future efforts.

## Background

The Uncommon Solutions HR Management System will be designed in a way that makes it easy to support multiple platforms such as Windows, macOS, iOS and Android. This web-based tool provides a direct method for storing and providing access to individual personnel records, and for all processes required for HR tracking and data aggregation requirements. The HR system will be implemented using AWS Elastic Compute Cloud (EC2) and Amazon’s Relational Database Service (RDS) in order to allow for universal deployability and access.

## Scope

This document provides a retrospective look at the design and development process and shows the distinct results of the efforts of the Uncommon Solutions development team.

## Schedule

The phased development schedule was a three-week process running from 18 November 2019 to 8 December 2019. There was an additional one-week flex time to allow for any schedule overruns and to allow for additional functionality to be added if time allows. This flex week ran from 9-15 December 2019.

## Design Considerations

The design for this program is as described in the Uncommon Solutions HR Management System Design Document. Any design variations will be validated by all members of the development team and incorporated into all design documentation to ensure that the entire development process is captured in documentation.

# Conclusions

## Lessons Learned

The first lesson learned as a team was to make sure that we take stock of each individual team member’s strengths and weaknesses and look at positioning within the team take best advantage of those things, especially when working with a short, set timeline project.

Our second lesson learned seems like common sense, but it was important for us to take into account the variety of schedules and time zone representation within the group to ensure that all members were provided the opportunity to contribute to the overall design and development process.

The third lesson learned was the importance of scheduling and making sure that we stayed within the guidelines of that defined schedule. By doing this we were able to ensure that our geographically separated team was able to collaborate in producing a final project.

The fourth lesson learned was the importance of clear communications! All important team communications were flowing through multiple mediums before the end of the project in order to ensure that everything was accomplished on schedule and that all team members were working toward the same goal on the same timeline.

## Design Strengths

The initially produced design required very little modification during the implementation process, the team had a good overall concept for the system from the outset. The simplicity of the design was its biggest strength as we were able to draw on the abilities and knowledge of various team members to push strengths where appropriate for database design, UI design, API creation, and programming functionality. This was not restricted to the design for the system itself, but the documentation requirements led to some significant contributions from the team members assigned those tasks.

## Limitations

Our biggest limitations were captured in the lessons learned, the largest being the difficulty of having times the entire group could get together to collaborate with the variances in schedule and time zone between the group members. This was overcome as best as possible with discussion taking place between those group members who could be present with after the fact contributions from those who could not be present.

## Suggestions for Future Improvement

The delivered system is a solid baseline from which to build upon. Initial thoughts for future improvement to the system include a more robust look at system security to include password requirements. The ability for individuals to self-register with validation of accounts at an administrative level. Expansion from HR records to additional HR functionality could be implemented as well, but would require investigation with users of the system on how it could be improved to better meet their needs.