**Software Design Document**

**Table of Contents**

1. Introduction
2. System Overview
3. Design Considerations
4. Architectural Design
5. Data Design
6. Component Design
7. Human Interface Design
8. Requirements Matrix
9. Appendices

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to provide a comprehensive design for a web-based workflow management system for non-crime-related activities at the YHROCU. The system will enhance task assignment, tracking, and management for individual and collaborative tasks within the department.

**1.2 Scope**

This document covers the design of the system, including data management, user interfaces, security, and functionality. The system will not manage crime-related activities, handle real-time task updates, or integrate with external law enforcement databases.

**1.3 Definitions, Acronyms, and Abbreviations**

* **YHROCU**: Yorkshire and Humber Regional Organised Crime Unit
* **CSV**: Comma-Separated Values
* **PDF**: Portable Document Format
* **OAuth**: Open Authorization

**1.4 References**

* OAuth 2.0 Authorization Framework: https://oauth.net/2/

**2. System Overview**

**2.1 System Context**

The workflow management system will operate within the YHROCU's internal network. It will interface with the existing email system and the department’s authentication infrastructure.

**2.2 System Functions**

* Task assignment
* Task status updates
* Task visibility control
* Task closure and deletion by supervisors
* Dashboard for task summary and categorization
* Data export (CSV, PDF)
* User authentication (OAuth)

**3. Design Considerations**

**3.1 Assumptions and Dependencies**

* The system will be accessed via modern web browsers.
* Users will have email accounts within the YHROCU domain.
* The existing authentication infrastructure will support OAuth.

**3.2 Constraints**

* The system must comply with internal security policies.
* The system must be scalable to handle increased task loads.

**4. Architectural Design**

**4.1 Overview**

The system will follow a client-server architecture. The client will be a web application, and the server will handle data processing, storage, and communication.

**4.2 Architectural Goals and Constraints**

* Ensure data integrity and security.
* Provide a user-friendly interface.
* Enable easy maintenance and updates.

**4.3 Architectural Patterns**

* Model-View-Controller (MVC) for the web application.
* RESTful services for server-client communication.

**4.4 Subsystem Architecture**

* **Client Subsystem**: Handles the user interface and user interactions.
* **Server Subsystem**: Manages data