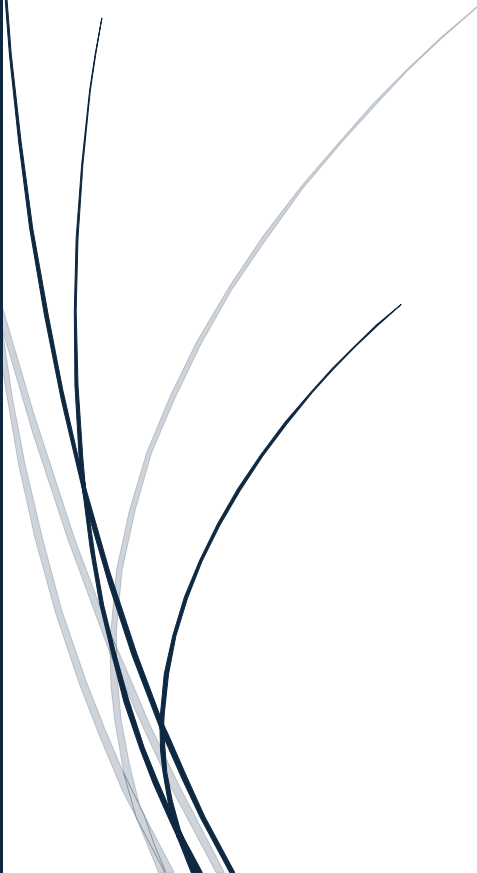


Team 25 15/02/2025

Workflow management system for non-crime related activity

Software Design Document



Contents

Introduction.....	2
1.1 Purpose	2
1.2 Scope	2
1.3 Reference Material.....	2
1.4 Definitions and Acronyms.....	2
Overview	3
2.1 System Overview	3
System Architecture	4
3.1 Architectural Design	4
3.2 Decomposition Description	5
Data Design	6
4.1 Data Description.....	6
4.2 Data Dictionary.....	6
4.3 Class Diagram	8
Component Design	9
5.1 Back End Component Design.....	10
5.2 UI Component Design	10
Human Interface Design	11
6.1 Overview of User Interface.....	11
6.2 UI Designs	12
Requirements Matrix	12

Introduction

1.1 Purpose

This document provides a detailed design of the Workflow Management System for Non-Crime Related Activity at Yorkshire and Humber Regional Organised Crime Unit (YHROCU). It outlines the system's architecture, components, modules and interactions to ensure the successful implementation of the system while showcasing the rationale behind the design.

1.2 Scope

The system is designed to improve task management, workflow tracking, and supervisorial reporting. It enables administrators, supervisors and users to manage tasks effectively and efficiently. This is achieved by allocating roles with unique levels of control creating a hierarchically structured and secure system. A user-friendly design, inclusive of neurodivergent staff, is also implemented using notifications to ensure tasks are never overlooked.

1.3 Reference Material

References will be provided as needed including system architecture frameworks and relevant documentation for easy understanding.

1.4 Definitions and Acronyms

- **YHROCU:** Yorkshire and Humber Regional Organised Crime Unit
- **WMS:** Workflow Management System
- **CSV:** Comma-Separated Value
- **PDF:** Portable Document Format
- **SQL:** Structured Query Language
- **PostgreSQL** – An open-source database management system.
- **HTML:** Hyper Text Markup Language – The standard markup language for creating web pages
- **JavaScript** – A coding language used to add dynamic interactions to web pages

Overview

2.1 System Overview

Login Page:

- OpenAuth implementation
- Roles are tied to logins.

User:

- View tasks assigned to that user
- View priority of their tasks
- Add and edit comments on their tasks
- Add files to tasks
- View and change status of tasks
- Change their password
- Update their profile

Supervisor:

- View all tasks
- Change priority of user tasks
- Review user tasks
- Create tasks for users
- Delete comments on tasks
- All user capabilities

Administrator:

- Add/Remove tasks
- Add/Remove Users
- Change the role of users
- Change password of users.
- All supervisor capabilities

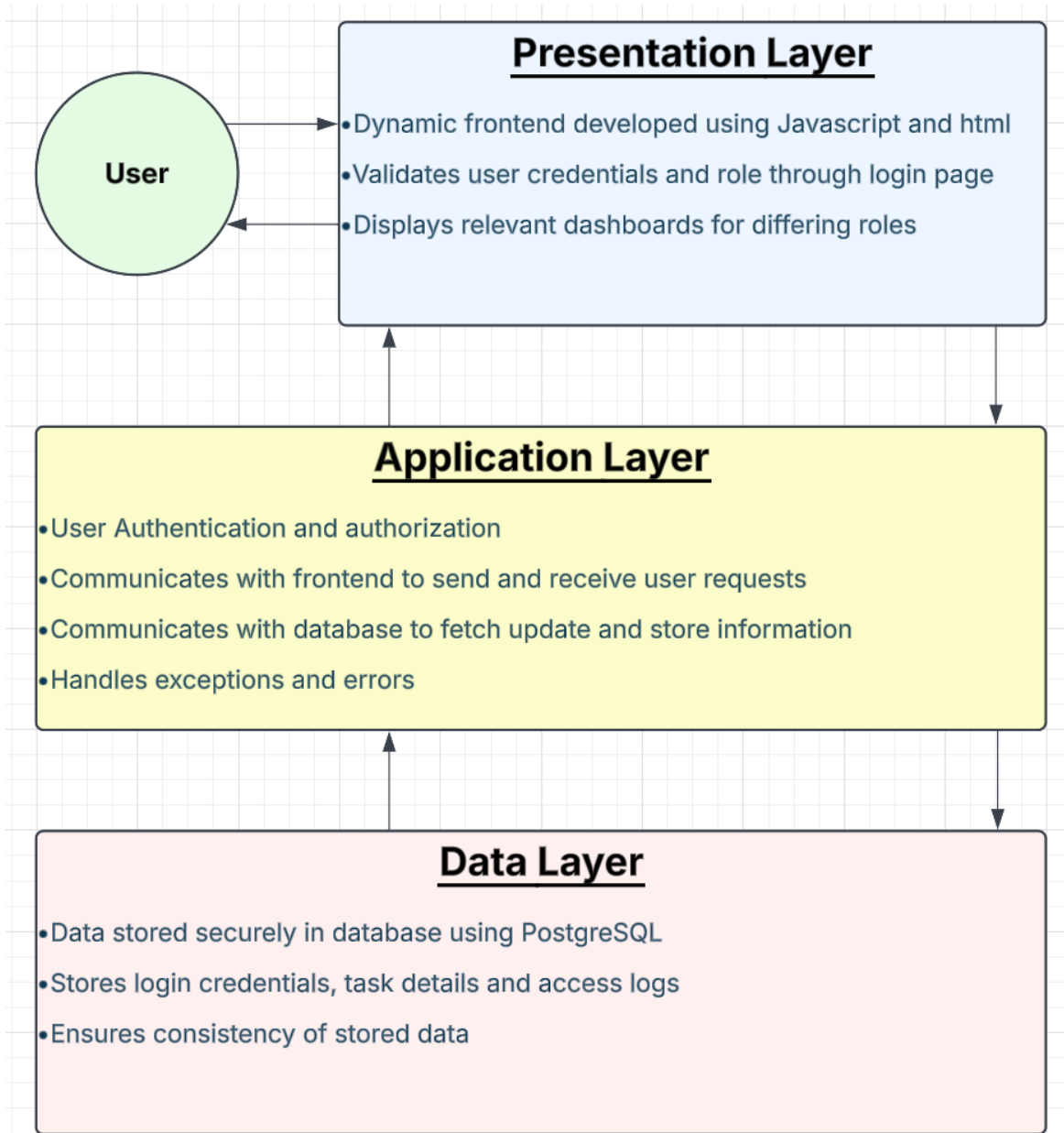
Non-role dependent features:

- Notify staff via email or popup when assigned a task
- Categorise and summarise tasks with filters (e.g., status, due date)
- Export tasks in CSV or PDF format
- Quickly find tasks via search function
- Progress logs on tasks

System Architecture

3.1 Architectural Design

The diagram below is a 3 Tier Architectural Design created using Lucidchart.



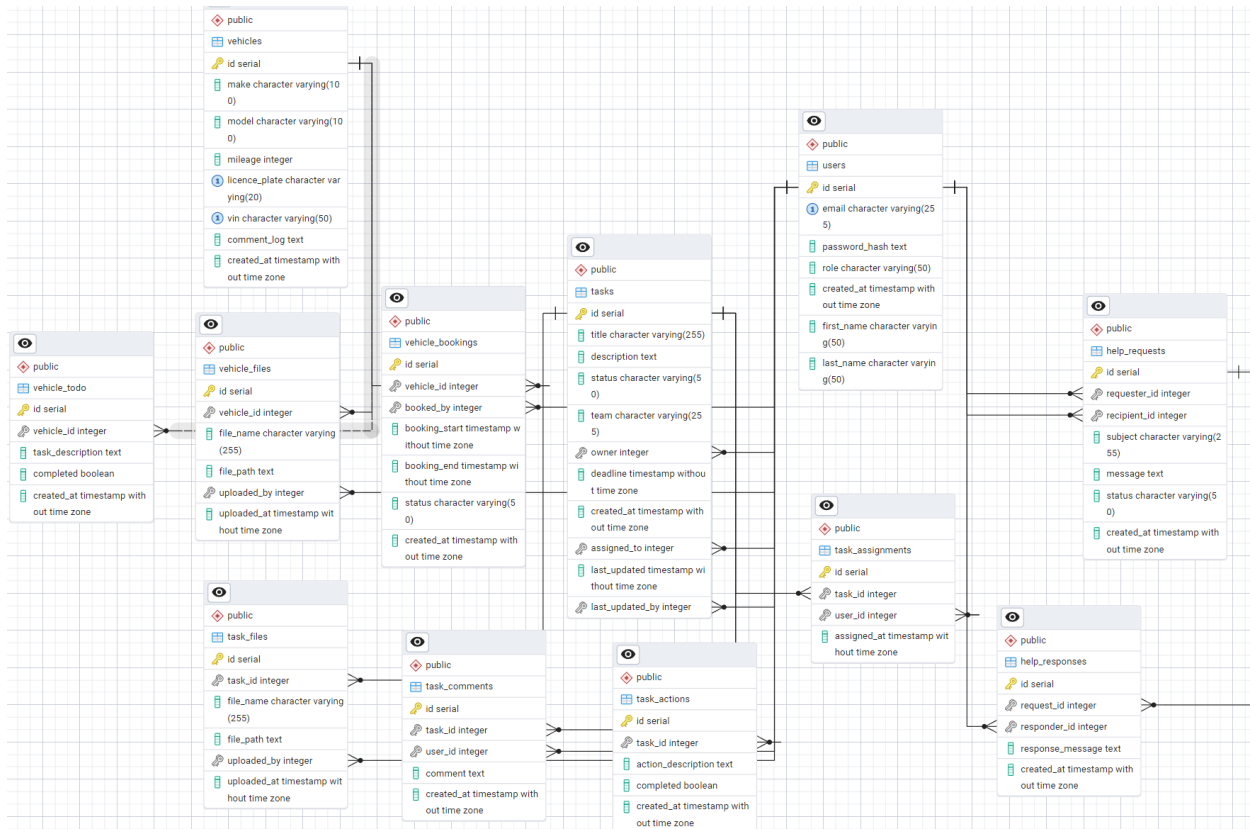
3.2 Decomposition Description

- **Authentication Module:** Manages user login, session handling, and role-based access
- **Task Management Module:** Handles task creation, assignment, and tracking
- **User Management Module:** Allows for creation/deletion of users, editing roles
- **User Dashboard:** Displays task lists and updates for users
- **Supervisor Dashboard:** Provides task oversight and reassignment options
- **Administrator Dashboard:** Allows for user management

Data Design

4.1 Data Description

The data of the WMS is stored into structured relational tables with clear relationships. Using primary and foreign keys to maintain referential integrity, the system ensures that all stored data is structured and normalised. A diagram showcasing the entity relationships is displayed below.

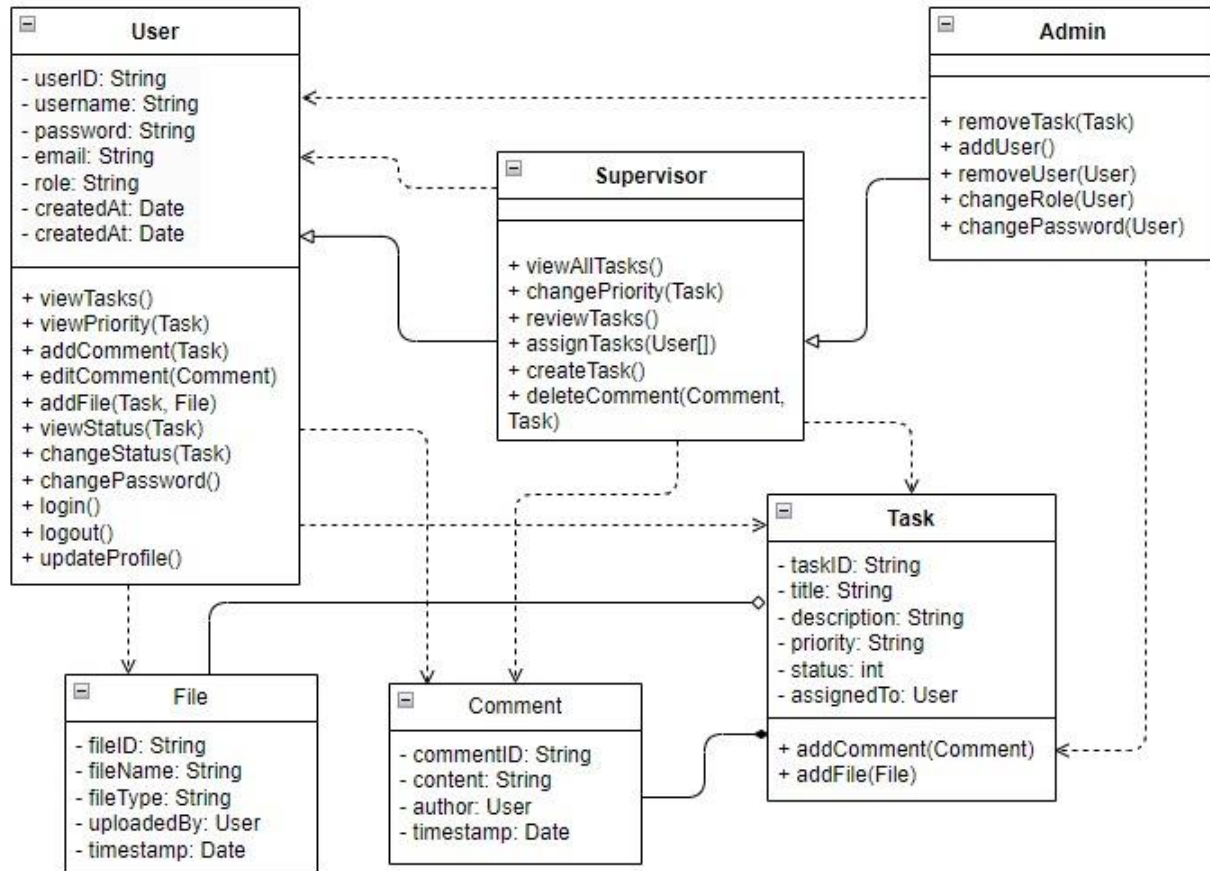


4.2 Data Dictionary

Entity	Attributes/Methods	Data Type	Description
Admin	removeTask(Task)	Method	Removes a task from the system.
	addUser()	Method	Adds a new user to the system.
	removeUser(User)	Method	Removes a user from the system.
	changeRole(User)	Method	Changes the role of a user.
	changePassword(User)	Method	Changes the password of a user.
Comment	commentID	String	Unique identifier for the comment.
	content	String	Text content of the comment.
	author	User	User who created the comment.
	timestamp	Date	Date and time when the comment was created.
	addComment(Comment)	Method	Adds a new comment to a task.
File	fileID	String	Unique identifier for the file.
	fileName	String	Name of the file.
	fileType	String	Type of the file (e.g., PDF, DOCX).
	uploadedBy	User	User who uploaded the file.
	timestamp	Date	Date and time when the file was uploaded.
	addFile(File)	Method	Adds a new file to a task.
Supervisor	viewAllTasks()	Method	Views all tasks in the system.
	changePriority(Task)	Method	Changes the priority of a task.
	reviewTasks()	Method	Reviews tasks for progress and accuracy.
	assignTasks(User)	Method	Assigns tasks to a specific user.
	createTask()	Method	Creates a new task.
	deleteComment(Comment, Task)	Method	Deletes a comment from a task.

Task	taskID	String	Unique identifier for the task.
	title	String	Title of the task.
	description	String	Brief description of the task.
	priority	String	Priority level of the task.
	status	int	Progress status of the task (0% to 100%).
	assignedTo	User	User to whom the task is assigned.
User	userID	String	Unique identifier for the user.
	username	String	Username used for logging in.
	password	String	User's password (hashed for security).
	email	String	User's email address.
	role	String	Role of the user (e.g., Admin, Supervisor, User).
	createdAt	Date	Date and time when the user account was created.
	updatedAt	Date	Date and time when the user account was last updated.

4.3 Class Diagram



Component Design

5.1 Back End Component Design

Authentication Module

- **Functions:**
 - User login/logout
 - Session management
 - Role assignment
 - Implement OpenAuth

Task Management Module

- **Functions:**
 - Task assignment
 - Priority setting
 - Status tracking
 - Comment editing
 - File attaching
 - Task deletion

User Management Module

- **Functions:**
 - User creation
 - User deletion
 - Role changing
 - Password Resets

5.2 UI Component Design

Admin Dashboard

- **Features:**
 - Allow for admin to see all users
 - Create or delete users
 - Can see password reset requests
 - Can edit roles of all staff.

Supervisor Dashboard

- **Features:**
 - View all tasks and their priorities
 - Can edit comments on tasks and see statuses
 - Can see progress and activity logs for tasks
 - Can create tasks and assign them to users

User Dashboard

- **Features:**
 - See all tasks assigned to user ordered by priority
 - Can attach comments and files to tasks
 - Search bar and filters to search for tasks
 - Can send password reset request to admin
 - Can update their profile
 - Will have notifications when assigned a new task
 - Can export tasks to CSV or PDF format.

Human Interface Design

6.1 Overview of User Interface

When the user first visits the site, they will be taken to a login page where they will be given the option to login. If the user chooses to log in the credentials will be checked to see if they match. They will then be sent to the corresponding page for their role.

If logged in as an admin, a list of all users and their info is shown. The admin will be able to delete and create users, edit their roles, and accept password reset requests. The admin can also switch to the supervisor dashboard.

If logged in as a supervisor, a list of ALL tasks will be shown including the priorities, comments, statuses and assigned users. Supervisors will also be able to create tasks to assign to users.

If logged in as a user, a list of all tasks assigned to the user is shown. Status, comments, priority and progress logs are all shown. A search bar with filters are included to help find tasks. Users can also update their profile and request for a password reset. There will also be a setting for dyslexia friendly font.

6.2 UI Designs

Concept Designs Are Stored On Github

Requirements Matrix

