A coat of arms with white birds and a book

Description automatically generated

**University Of Bradford**

**Department of Computer Science**

**Requirements Specification Document and Prototype Implementation**

*Workflow management system for non-crime*

*related activity for Yorkshire and Humber Regional Organised Crime Unit (YHROCU)*



**Team 1**

**Team 1**

17003493 - Sania Bibi

22031400 - Javairia Shahid

20010596 - Amal Abeso Ela

22006776 - Imara Ali

21045121 - Oluwadamilare Falade

22036203 - Shafeeq Shuaib

Contents

1. Requirements Specificaton.............................................................................................
2. Project brief
3. Introduction of team expertise
4. Rationale of topic choice
5. Work plan
6. Peer review
7. Functional and Non-functional requirements.....................................................
8. Actors
9. Functional requirements table
10. Non-functional requirements table

**Project Brief**

|  |  |
| --- | --- |
| **Project Name** | Workflow management system for non-crime related activity for Yorkshire and Humber Regional Organised Crime Unit (YHROCU) |
| **Project Overview** | YHROCU requires an enhanced workflow management system to streamline non-crime related activities within its department. The system aims to efficiently manage support tasks assigned to staff, facilitating both individual and collaborative work. |
| **Project Objectives** | * Develop a web-based system for task assignment and management. * Improve communication by sending email notifications to staff when tasks are assigned. * Enable staff to update task status, due dates, review dates, and input progress updates into a rolling log. * Prevent deletion of tasks or any previous updates. * Provide supervisory access to view all tasks, with options to restrict visibility as needed. * Allow supervisors to close or delete tasks. * Implement a dashboard for task categorization, summarization, and filtering based on status or due date. * Include export functionality (CSV or PDF) for data analysis and reporting. * Ensure flexibility by allowing addition of new data fields to tasks for future adjustments. * Incorporate a search function for efficient task retrieval. * Implement user authentication using OpenAuth or similar for compatibility with existing infrastructure. |
| **Timeline** | Depicted in the attached Gantt chart. |
| **Stakeholders** | * YHROCU Management Team * YHROCU Staff Members * External Development Team (if applicable) * University of Bradford * Team 1 |
| **Risks** | * Potential delays due to unexpected technical challenges. * User adoption challenges. * Compatibility issues with existing infrastructure. * Project not being completed to a good standard for 15th March * Insufficient security measures leading to GDPR violations |
| **Success criteria** | * Successful implementation and adoption of the workflow management system by YHROCU staff. * Improved efficiency in task management and communication within the department. * Positive feedback from users regarding system usability and effectiveness. * A good mark in the Enterprise pro module |

**Introduction of team expertise**

**Sania Bibi 17003493** – Quick learner and is able to grasp concepts very quickly. Proficient in fundamentals of computer science as demonstrated in first year results (90+ in building a website, fundamentals of programming, software design and development). Comfortable using JAVA, HTML, CSS, JavaScript and has developed an iOS app using Swift programming language.

**Oluwadamilare Falade 21045121** – I bring a unique blend of strengths to team dynamics, embodying qualities of a 'plant' and a 'shaper' in Belbin team roles. Proficient in Java, Python, HTML, CSS, JavaScript, PHP, and MySQL. Experienced in developing a sports website. Achieved commendable grades in relevant courses such as Data Structures and Algorithms (76%) and Computer Architecture and Systems Software (95%). Demonstrated problem-solving abilities in individual projects. Committed to impactful contributions within collaborative settings, anchored by technical proficiency and proactive problem-solving.

**Shafeeq Shuaib 22036203** – Possesses expertise in crafting dynamic websites using JavaScript, CSS, PHP, SQL, and HTML. They're adept with Apache NetBeans and IntelliJ, having worked extensively with Java. Python proficiency is also part of their skill set. Their experience extends to phpMyAdmin, MySQL, and LAMP servers in conjunction with SQL programming. Familiarity with Microsoft products like Word, Excel, and PowerPoint complements their technical capabilities. Academic achievements, including high grades in Mathematics, Computer Architecture, Software Design, and Database Systems, reflect their commitment. Noteworthy qualities include robust teamwork, leadership, independence, and adeptness in lab settings.

**Amal Abeso Ela 20010596** – Extremely comfortable with teamwork, knows how to adapt herself in any situation. Possess abilities in JavaScript, CSS, SQL and HTML. Having worked previously with Java and Python. Has a familiarity with Microsoft products like Word and PowerPoint. Spends her time learning how to improve her coding skills either in Visual Pro or Apache NetBeans.

**Imara Ali 22006776** – Possess a diverse set of programming languages including JavaScript, CSS, PHP, SQL, and HTML to develop dynamic websites as part of various projects. Particularly adept at Java programming language due to extensive coursework. Additionally, capable in Python programming language. Demonstrated competence with phpMyAdmin and MySQL while working extensively with SQL for database management. Experienced in Microsoft Office suite including Word, Excel, and PowerPoint. Achieved exceptional grades in key modules, averaging 75% in year 1. Notable strengths include strong team collaboration, unwavering commitment, occasional leadership roles, self-reliance, and proficient lab skills.

**Javairia Shahid 22031400** – Highly skilled team player, allowing to work well within a team as well as lead the team. Proficient in using different programming languages such as JavaScript, Python and MySQL. Able to use different Microsoft applications such excel and word to execute tasks i.e. Gantt Charts, and able to adapt well and learn skills that will help with the duration of the project, for example GitHub. An active listener and communicates well with the group and external client to help deliver the assigned task.

**Rationale of topic choice**

The list below outlines some of the reasons why we chose to develop a workflow management system for YHROCU.

1. The project was a good fit in that aligned with our personal views. YHROCU is an organisation which aims to protect the public and make communities safer from the threat and harm of serious and organised crime. Our contribution to the organisation through this project allows us to serve the public.
2. The development of the management system was a good balance between application of our existing programming skills whilst also allowing us to learn new concepts. For example, some of our members are proficient in coding however completing tasks such as 2FA and allowing the user to export data are new concepts which would help the members build knowledge and skill base.
3. The chosen project would utilise *everyone’s* skills on team e.g., two members are proficient in coding whilst others are proficient in planning/organization and others at scoping out exact functional and non-functional requirements. This does not mean each member only works on areas they are good at but that they can contribute a baseline of work comfortably.
4. The chosen project seemed interesting as it required concepts, we had learned in first year therefore allowing us to put theory into practice.

**Work plan**

(gantt chart)

**Use Case Diagram**

The system consists of three actors. The Administrator, the supervisor and normal staff members.

* The normal staff members can sign up, log in, view tasks, add updates to tasks. They can also export tasks data in a PDF format if need be.
* The supervisors can do everything a normal staff member can do but also; declare tasks as completed. They also can unlock an account in case a staff member gets locked out.
* The administrator can do everything the supervisor can in addition to exclusively being able to delete tasks and promoting staff members to a supervisor.

A screenshot of a computer

Description automatically generated

**Functional Requirements**

|  |  |
| --- | --- |
| **Login page** | Every user would be able to log into their account with their staff number and a password. Once the information is validated, the access to their respective accounts would be granted, otherwise, access would not be granted, and the login info will have to be retyped. |
| **Logout** | User will be signed out and directed back to the login page. |
| **Unblocking an account** | Each staff member has 3 attempts to log in, once these attempts are exhausted the account will be blocked and will have to be unlocked by a supervisor. |
| **Sign up** | Every new user will have the ability to create they own account using their last and first name, their team, staff number and email address. |
| **Create task** | This page is used so users could create or add new tasks. |
| **View Task** | This page is created so every user could see all the tasks that are either accomplished either to do |
| **Add updates** | Each task will have an update button next to it, when the user clicks update tasks, they will be able to add notes about the task which will get added to the database. |
| **Mark Task as complete** | The supervisor actor will have the option to mark a task as complete. When the task is complete, no user will be able to add updates to it. |
| **Delete task** | Only the admin will have the power to delete a task, it will also ask for a confirmation pin before deleting to ensure it wasn’t a click error. |
| **Search page** | Users will be able to search for tasks by their task name, if the task exists in the database, it will be displayed on the screen. |
| **Exporting task** | Users will be able to export task data into a downloadable PDF file. |
| **Assign tasks** | The supervisor can assign tasks to specific staff members, this automatically triggers an email to their inbox. |
| **Update due date** | The supervisor can update due date of tasks |

**Non-Functional Requirements**

|  |  |
| --- | --- |
| **System Performance** | The system must respond swiftly to user actions like logging in or creating tasks, ensuring a seamless experience. It should effortlessly manage 1000 concurrent users without noticeable slowdowns. |
| **Security Measures** | User passwords must be securely hashed and stored to safeguard against unauthorized access. Data transmissions must be encrypted with HTTPS, preventing any interception of sensitive information. Login pages should have defences against brute force attacks, like temporarily locking accounts after 3 unsuccessful attempts. Admin accounts should have additional security layers, such as two-factor authentication, for added protection. |
| **Scalability Plans** | The system's design should facilitate easy scaling to accommodate more users and tasks. The database architecture should allow for seamless scaling by adding resources or switching to larger database systems as needed. Including a common programming language that can be adopted and used by the clients inhouse IT department. |
| **Reliability Assurances** | A robust backup and recovery strategy should be in place to prevent data loss. Regular system backups must be performed to ensure data integrity. Monitoring tools should promptly detect and handle system failures or crashes. |
| **User-Friendly Interface** | The user interface should be intuitive and easy to navigate for all users. Clear error messages should guide users when they make mistakes, such as entering incorrect login credentials. Implement accessibility features to cater to users with disabilities. |
| **Compatibility Considerations** | The system should work seamlessly across major web browsers like Chrome, Firefox, Safari, and Edge. It should also be responsive and adapt well to various devices including desktops, tablets, and smartphones. |
| **Maintainability Focus** | Code documentation and organization should facilitate easy maintenance by future developers. Deploying updates and patches should be smooth without causing system downtime. Utilize proper version control to track changes and revert if needed. |
| **Legal and Compliance Guidelines** | Compliance with data protection laws such as GDPR should be ensured. Clearly visible privacy policies must be accessible to users. User data should only be utilized for its intended purposes and not shared with third parties without explicit consent. |
| **Compatibility with existing software** | The software should be compatible with existing software as outlined by the clients’ requirements (Open Auth) |
| **Cater to neurodivergent individuals** | The interface should cater to neurodivergent individuals, no harsh colours example yellow writing on a white background or no extremely small print. |

**Data Description: (will need amending for additional tables when more functionality is added to code)**

Data handled by admin are as follows:

Tasks: with fields identifying what the task is.

Tasks\_staff: what tasks are assigned to each staff member.

Staff: Details of each staff member and their log in details.

**Tasks:**

Task ID – Unique identifier for the task – Going to increment each time.

Task name – Max characters 8 – Suggests what the task is for.

Task Description – Max characters 250 – Goes into more detail about what specifically needs to be done for the task to be complete.

Due date – dd/mm/yyyy the date of the task being due.

Assignee – Max digits (6) as this will be linked to the staff number – Person the task is assigned to.

Requester – Max digits (6) as this will be linked to the staff number – Person that created and assigned the task.

**Tasks\_staff:**

Task ID – Max digits(250) – Unique identifier for the task – Going to increment each time.

Assignee – Max digits (6) as this will be linked to the staff number – Person the task is assigned to.

Requester – Max digits (6) as this will be linked to the staff number – Person that created and assigned the task.

**Staff:**

First name – Max character (50) – First name of staff member.

Last name – Max character (50) – Last name of staff member.

Team – Max character (50) – The team the staff member is a part of within the organisation.

Role – Max character (20) – The role of the customer could be Admin or Staff.

Email – Max character (100) – The staff members work email that they have been assigned.

Password – Max character (15) – The password the staff member created to log in.

Staff Number – Max digits (6) – Unique identifier for the task.

**Class diagram**

* Supervisor can change due date for tasks
* Supervisor can assign tasks to staff numbers and when that staff number logs in they can see their task and all tasks
* Pages should only been seen when logged in
* Generate a report for error handling (later)
* Users should get an email when a task is assigned
* 2FA
* Exporting PDF’s

**User Interface**

This is the login and signup page. To log on to the system, the user must provide their staff number and password and to sign up the user must fill in all of the required information. If a field is left blank, or incorrect details are provided it will prompt the user to complete or amend. There is a toggle button at the top which slides between the two pages.

Company Logo

First Name

Last Name

Signup

Staff Number

Team

Password

Login

Signup

Toggle between login and signup

Login

Signup

Company Logo

Staff Number

Password

Login

Error message if all fields are not filled in

Login failed if user does not exist in the database. Account blocked if 3 unsuccessful login attempts.

This is the ‘All tasks’ page. The tasks will be displayed dynamically from the database. Each task will also have 3 buttons next to it. Delete, open, mark as complete. Only the administrator will be able to delete a task. Only the supervisor will be able to mark a task as complete.

All Tasks

Create Task

Search Task

Logout

All Tasks

taskName, taskDesc, dueDate

taskName, taskDesc, dueDate

taskName, taskDesc, dueDate

taskName, taskDesc, dueDate

To delete or mark as complete the user a prompt will ask for supervisor/admin password or pin.

1. Delete
2. Mark as complete
3. Open

All Tasks

Create Task

Search Task

Logout

Open task

taskName, taskDesc, dueDate

task details and updates task details and updates

task details and updates task details and update task details and updates task details and updates

Update

Update

This is a popup that will appear when the user clicks open above. It will allow the user to add updates about the task. Here supervisors will be able to assign task to a staff number and update the due date. When a task is assigned to a staff number, that staff member will receive an email.

This is the search page. The user will be able to search for tasks by task name, if the task exists in the database it will appear in the box below.

This is the create task page. Here the user will be able to create tasks. When a task is created, it is added to the database and can automatically be seen in the “All tasks” tab.

Task Name, description and due date are all input fields with description being a text area and date being a calendar.

Create Task

All Tasks

Create Task

Search Task

Logout

Task Name

Task Description

Due date

Create Task

The logout button logs the user out and takes them back to the login page.

All Tasks

Create Task

Search Task

Logout

taskName, taskDesc, dueDate

Search bar

**Reflection on legal, social, ethical, security, professional issues of the project, risk and**

**economic aspects, environment awareness**

**Legal issues**

* Ensuring that the project complies with all relevant laws and regulations related to environmental protection, data privacy, and public safety.
* Determine the legal liabilities associated with the project, including any potential risks or damages to the environment or public.
* The system would need to have a privacy policy showing staff members transparency on how their data will be collected and stored.

**Social and ethical issues**

* Ethical considerations may include responsible AI use, avoiding bias, and maintaining transparency. For example, our project requires the supervisor user to assign tasks to staff members, if we adopted an AI algorithm to automate this, how would we eliminate bias when choosing staff members for task assignation.
* Another ethical issue which comes to mind, is that the project client initially asked that all tasks generated should be viewable by all staff members, this poses the ethical question about confidential information, or coveted tasks. Should everything still be viewable by everyone?

**Professional issues**

* The professional issues which may arise tie into some of the other LSEPI. One of the main professional issues is that the project is being developed by inexperienced students. If sufficient security measures were not put into place and after YHROCU adopted the programme who would be at fault.
* Although an NDA was signed, students often upload their projects on GitHub or other social avenues to showcase their experience and work to potential employers. This could lead to a possibility of "data breach". If the students waives the right to showcase the work, it takes away from their professional career especially if significant effort was put into it.

**Risk aspects**

* Legal and Compliance Risks: Assessing the risk of non-compliance with relevant laws, regulations, and policies, this could lead to legal liabilities or fines.
* Reputational Risks: How the project might impact YHROCU's reputation and public trust, especially if there are negative consequences or controversies associated with its implementation e.g., system failures etc.
* Security Risks: What happens the risk of data breaches, cyberattacks, or other security incidents that could compromise sensitive information or undermine the project's effectiveness.

**Economic aspects**

* Some economic points to consider are, if YHROCU was going to implement this workflow system over other already established products, what are the economic benefits? Will it save staff time? Save time as staff will require minimal training? Will it require significant work from their IT department to make the product robust and at an institutional standard? Will the product be sustainable long term?

**Environment awareness**

* The development of the project is for Yorkshire and Humber Regional Organised Crime Unit (YHROCU). YHROCU is an organisation which aims to protect the public and make communities safer from the threat and harm of serious and organised crime. If our project is adopted by YHROCU it will be directly impacting the environment/public.

**Some examples of risks that the project may encounter and strategies on how to mitigate**

|  |  |
| --- | --- |
| Risk | Mitigation strategy |
| Data Breach: Unauthorized access to sensitive data. | Implement robust encryption, regular security audits, and intrusion detection systems. |
| legal compliance: Non-compliance with data protection laws. | Stay informed about relevant regulations, implement privacy by design and conduct legal reviews. |
| Ethical concerns: Unintended bias in algorithms or unethical use of data. | Regularly review and audit algorithms, prioritize fairness, and establish ethical guidelines for system use. |
| Mitigating Intellectual Property Infringement: Potential risk involves unintentional infringement on intellectual property rights. | Conduct thorough IP searches, secure necessary permissions, and document the development process meticulously to prevent legal complications. |
| Data access control:  Unauthorized access to sensitive data. | Enforce strict access controls, limiting access based on job roles, and conduct regular audits to ensure compliance. |
| Communication security: Risk interception of sensitive data during transmission. | Use secure protocols (e.g., HTTPS), implement VPNs and employ encryption for data transmission |

**GitHub Link to:**

* **Meeting minutes**
* **NDA**
* **Prototype**
* **Requirements document signed by client**

**Peer Review**

All members contributed equally and achieved tasks to a good standard. All members have a score of 10.

17003493 - Sania Bibi

* Completed project brief, some of the LSEPI issues, user interface and some of the code for the prototype (create task, view task, update task).

22006776 - Imara Ali

* Completed the data description, and code for the login and signup pages for the prototype.

22031400 - Javairia Shahid

* Responsible for organising team meetings, documented all meetings and meeting minutes. Wrote the NDA and got it and the specification signed by the client. Organised project timeline and created Gantt chart. Helped with search page with Amal.

20010596 - Amal Abeso Ela

* Completed the LSEPI, functional and non-functional requirements and information about actors. Helped with search page with Javairia.

21045121 - Oluwadamilare Falade

* Liaised with other team members and completed the use case diagram and class diagram. Did designs and CSS of the website with Shafeeq.

22036203 - Shafeeq Shuaib

* Completed the LSEPI, functional and non-functional requirements and information about actors. Did designs and CSS of the website with Dami.