



A Web-based Expense Management System

Rachid Aboulachouaq (2237654) – BSc (*Hons*) Computing (Software Development)

2237654@student.uwtsd.ac.uk

University of Wales Trinity Saint David

1. Introduction/Background

The current university expense claim process relies heavily on email and manual administration.

This results in inconsistent and incomplete information, limited visibility of expense progress, and additional workload for staff to correct any errors.

2. Aim and Objectives

Aim:

To design and develop a web-based expense system to mitigate data errors, improve efficiency and transparency of the current expense process.

Objectives:

- To analyse the current expense process and identify key user requirements
- To design an accessible user interface for expense submission
- To develop a web-based system for submitting, reviewing, and approving expenses
- To test the system for usability, functionality, and reliability
- To evaluate the system against user and organisational requirements

3. Conclusion and Findings

This project delivered a functional prototype of a web-based management system, introducing a more effective alternative to email-based management of claims.

Usability testing produced a 90.5% task completion rate without prior instruction based on a small sample size of end users. The post evaluation questionnaire produced an overall mean of 4.0 out of 5.0.

In Summary, the system largely achieved its aim by supporting claim creation, submission, status tracking and role-based access control, and validating form data at input.

However, further improvements are required before deployment to a live environment, including institutional single-sign on, mileage calculation and receipt uploads.

Statement	Mean / 6
Claim creation was straightforward	4.0
Difference between Draft and Submitted was clear	3.8
Claim details were sufficient	4.0
Manager comments were clear	4.0
Approval and rejection were intuitive	4.0
Button outcomes were predictable	4.0
Error messages were helpful	3.7
System response was sufficiently quick	4.7

Figure 1: Summary of Usability Test. Source: Author's Evaluation

The system was designed around two main user role; manager and employees. Having a user centred-design helps ensure alignment with user tasks [1-2]. The design aimed to improve the underlying workflow of the expense process, with the interface separating the user roles to maintain role-based access and validation was used to improve consistency and accuracy of the provided information [1],[3].

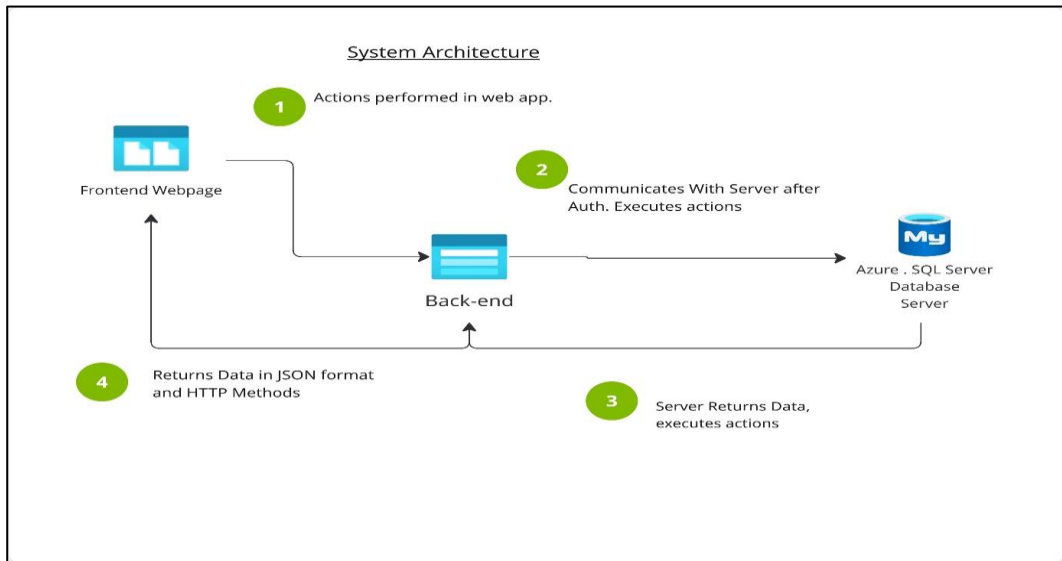


Figure 3: Proposed System Architecture Diagram. Source: Author's Own Work.

4. Design

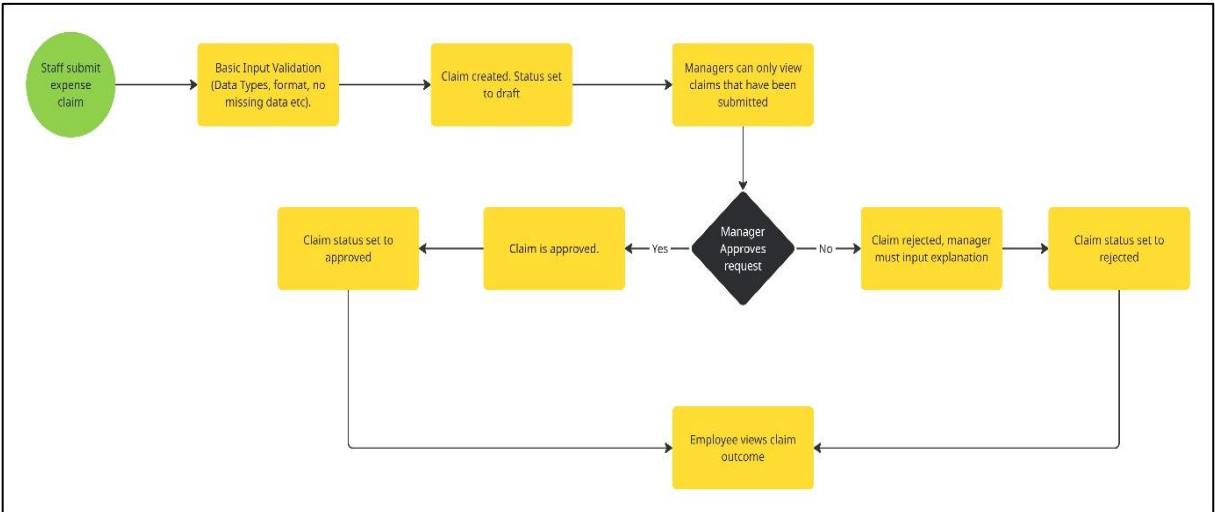


Figure 2: Proposed System Workflow. Source: Author's Own Work.

Created expenses follow defined states of: Draft, Submitted, Approved and Rejected. This provides users visibility of the status of their claim progress

5. Implementation & Proposed Solution

The solution implemented made use of a client-server architecture and a single page architecture design for the frontend. This was developed using industry recognised standards. The frontend provided a usable interface, while validating data at input, and the backend handled business logic and communication with the database

Frontend:

- Vite
- React
- TypeScript
- Shadcn UI

Backend:

- C#
- ASP.NET Core Web API
- Microsoft Entity Framework Core

JSON Web Token authentication was used to identify users and role-based authorisation was used to restrict managerial functions. Employees can view their own claims and managers are able to review before approving or rejecting claims [4].

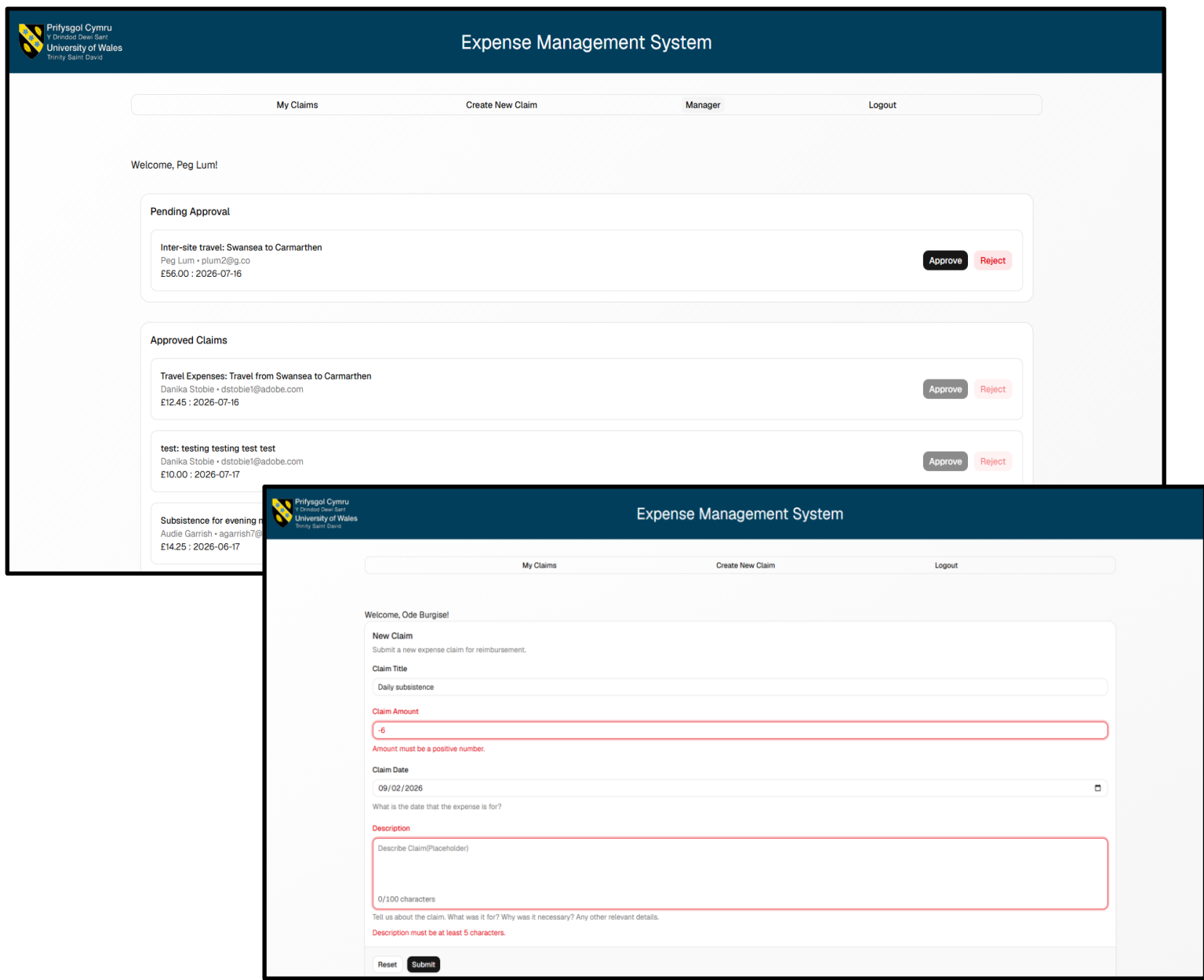


Figure 4: Manager Overview of Pending Claims and Actions. Source: Author's Own Work.
Figure 5: Validation of Input at Claim Creation. Source: Author's Own Work.

6. Methodology

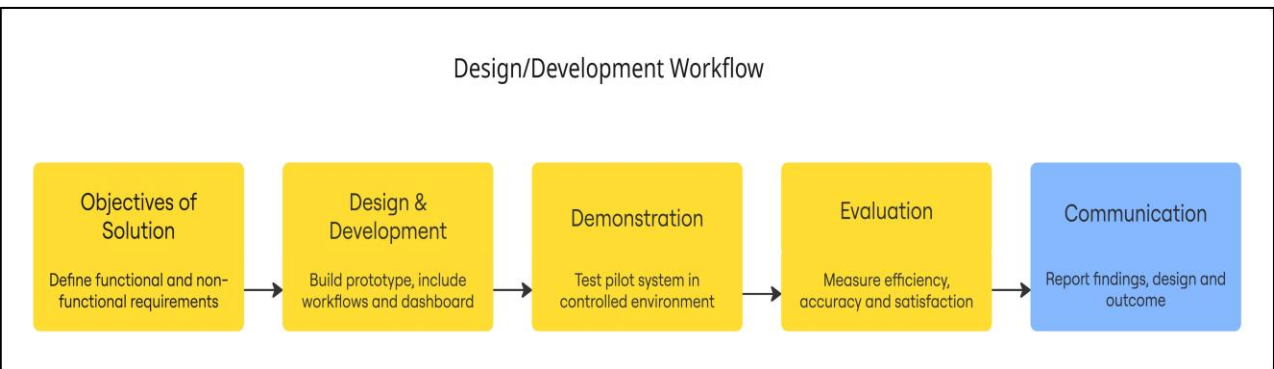


Figure 6: Design Flowchart. Source: Author's Own Work.

A pragmatic and iterative approach was adopted, and Requirements were identified through analysis of the existing process and consultation with stakeholders familiar with the process

Functional Requirements	Non-functional Requirements
Authentication	Usability, users should be able to intuitively use the application
Create and save claims	Form validation to reduce data errors
Submit Expense claims	Authentication and authorisation to protect financial information and adhere to organisation security policy
View Current Claim Status	
Managers to approve/reject claims	
Role-base restrictions	

Figure 7: Functional and Non-function Requirements. Source: Author's Own Work.

Quantitative and qualitative feedback was through eight 5-point Likert scale questions, while open-ended responses identified strengths of the user interface, as well as areas for improvement

7. References

- [1] M. Brhel, H. Meth, A. Maedche and K. Werder, "Exploring principles of user-centered agile software development: A literature review," vol. 61, pp.163–181, 2015, doi: 10.1016/j.infsof.2015.01.004.
- [2] F. L. Verdi, H. T. d. Oliveira, L. N. Sampaio and L. A. M. Zaina, "Usability Matters: A Human–Computer Interaction Study on Network Management Tools," vol. 17, no. 3, pp.1865–1878, 2020, doi: 10.1109/TNSM.2020.2987036.
- [3] T. Monteiro, S. Abrantes and M. Ratinho, "A Solution for Submitting Expenses," vol. 237, pp.20–27, 2024, doi: 10.1016/j.procs.2024.05.075.
- [4] M. Uddin, S. Islam and A. Al-Nemrat, "A Dynamic Access Control Model Using Authorising Workflow and Task-Role-Based Access Control," vol. 7, pp.166676–166689, 2019, doi: 10.1109/ACCESS.2019.2947377.