



HotelOps

Development of a Multi-User Web Application for Hotel Departmental Communication



Mariana Lupu

BSc (Hons) — Level 6 Independent Project
University of Wales Trinity Saint David

BACKGROUND

Hospitality technology has developed unevenly: front-of-house systems are mature, while housekeeping, maintenance, and other operational staff are often left with paper logs, radios, and verbal handovers [1]. This gap is well documented in the wider literature, but it is also something I encountered directly across ten years working in hotel housekeeping and reception.

HotelOps was built to close that gap for one 125-room hotel: a single, role-based platform covering every department, not just the front desk.

AIM & OBJECTIVES

- To design, develop and evaluate a multi-user, role-based web platform that improves departmental communication and operational efficiency in a hotel environment.
- 1 Identify and analyse user requirements across seven hotel roles
 - 2 Design a secure, role-based application [2]
 - 3 Implement with React, Flask and SQLite
 - 4 Test through unit, integration, system and acceptance testing
 - 5 Evaluate the system and reflect on the process

METHODOLOGY

An Agile-inspired iterative approach was adopted [3] rather than a fixed Waterfall plan, since requirements kept changing as real operational needs became clearer through building and testing each feature.

7

user roles

10

pages

125

rooms managed

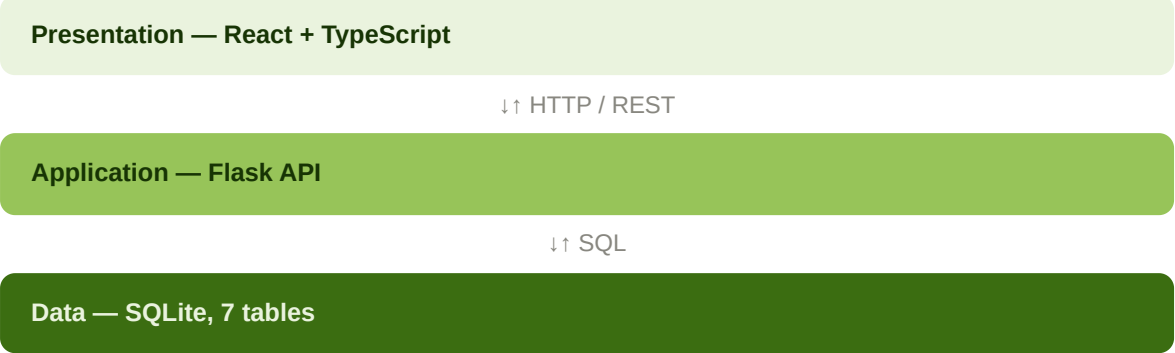
SEVEN ROLES

General Manager	Housekeeping Lead
Reception	Room Attendant
Maintenance Tech.	Linen Porter
Administrator	

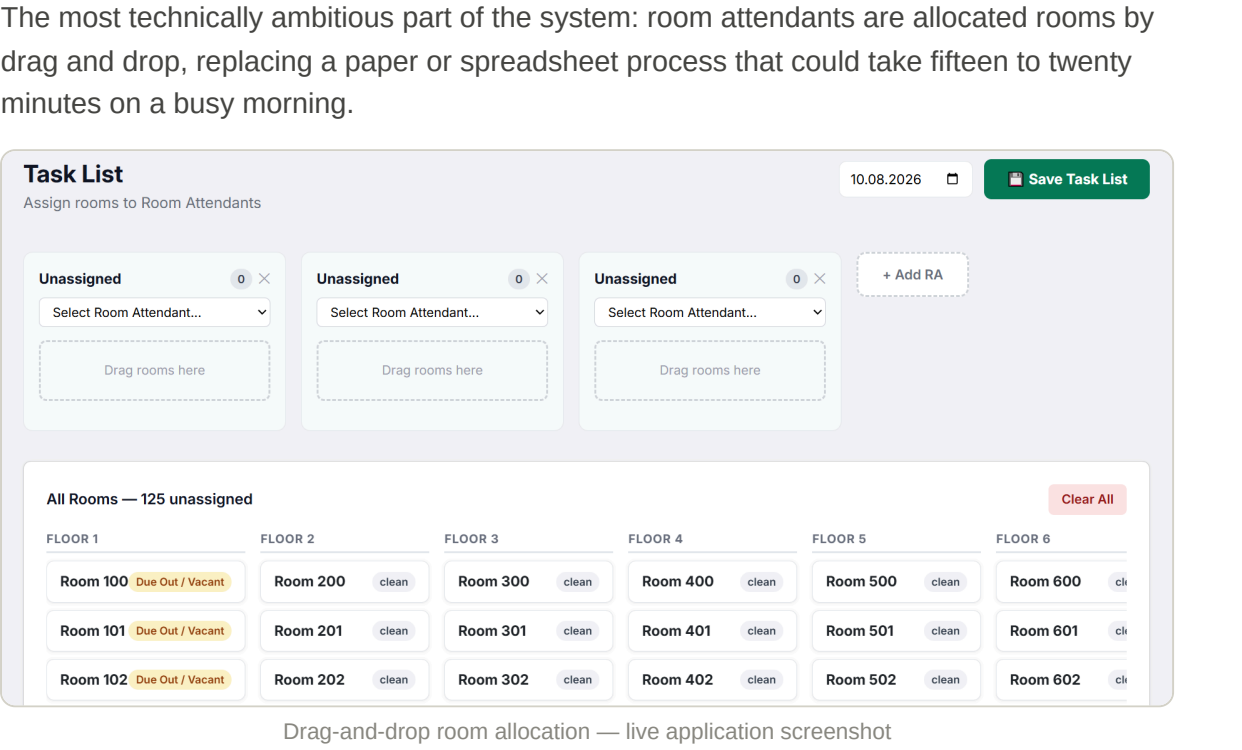
SYSTEM ARCHITECTURE

A three-layer client–server architecture keeps the interface, business logic, and data storage independently maintainable.

The frontend was built with React and TypeScript for a maintainable, component-based interface [4], and the backend with Flask, chosen for its simplicity and control over a growing codebase [5].

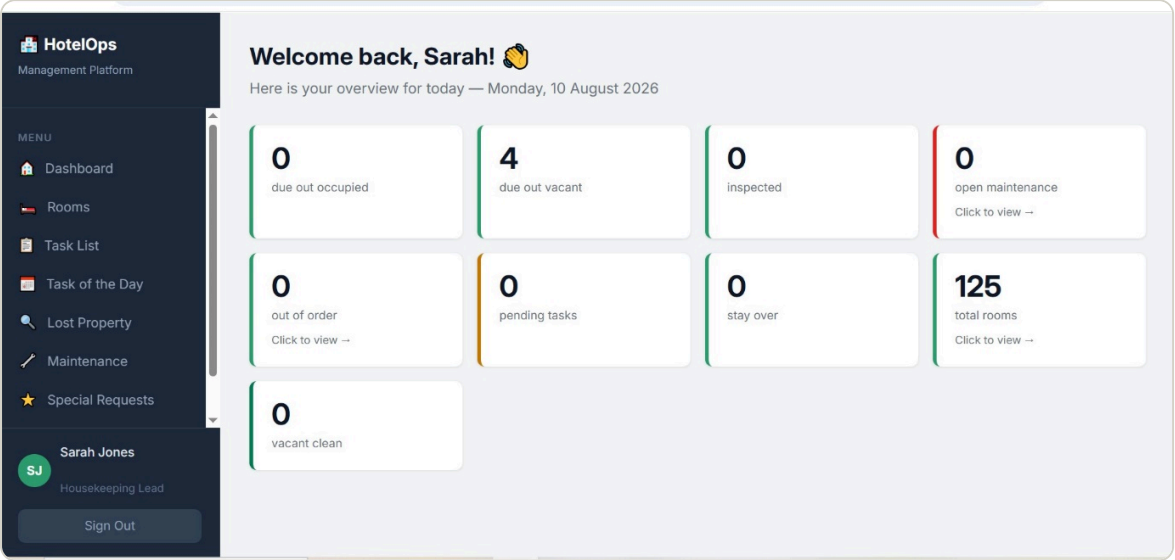


KEY FEATURE: THE TASK LIST



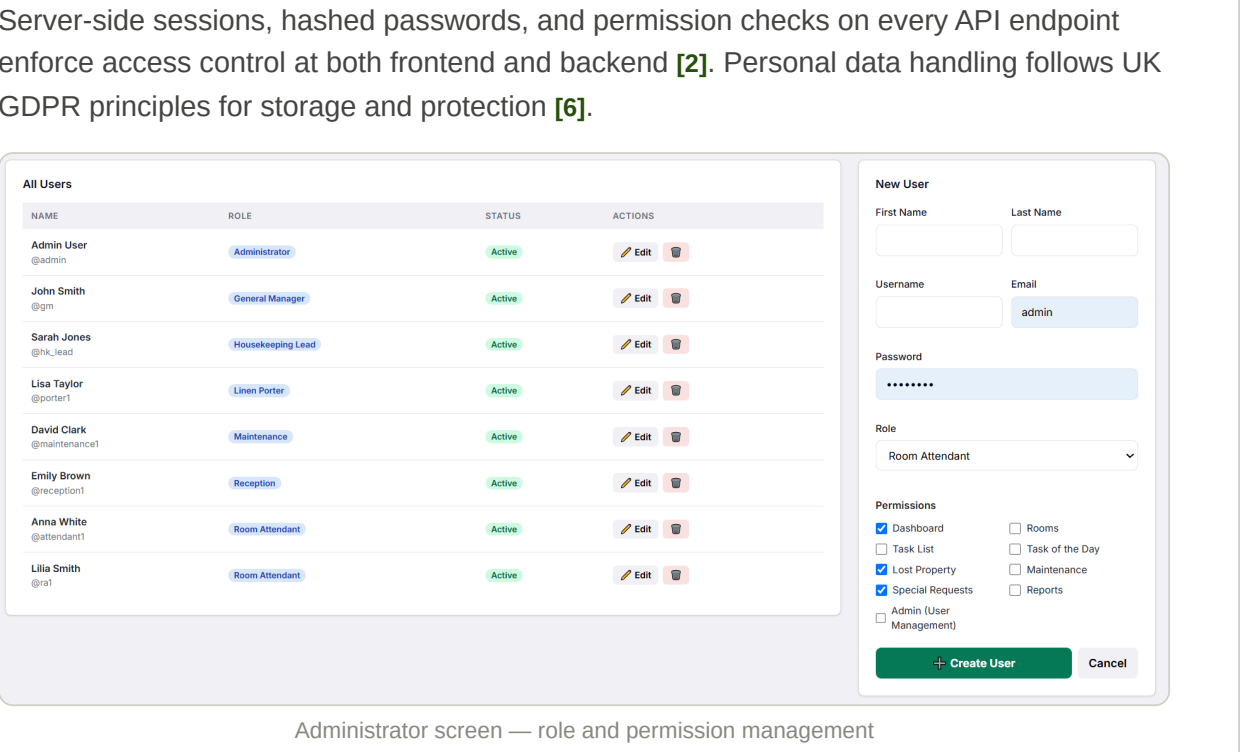
Drag-and-drop room allocation — live application screenshot

ROLE-BASED DASHBOARD



Housekeeping Lead dashboard, showing only role-relevant data

SECURITY & DATA PROTECTION



Administrator screen — role and permission management

TESTING

Five testing types were carried out throughout development, not left until the end. The interface was also evaluated against Nielsen and Molich's usability heuristics [7].

TYPE	CASES	RESULT
Unit	10	Pass
Integration	8	Pass
System	7	Pass
Acceptance	6	Pass
Usability	10 heuristics	Pass

EVALUATION

Strength — designed by someone who has done the jobs it supports; room statuses, task allocation and pending-request visibility reflect real hotel-floor workflow.

Limitation — no live PMS integration yet; SQLite and manual status updates suit this scale but not a multi-property deployment.

FURTHER WORK

PMS integration · real-time WebSocket notifications · mobile-first interface for room attendants · historical reporting · production database migration.

CONCLUSION

HotelOps demonstrates that the daily operational problems affecting hotel staff are not difficult to solve technically — what has been missing is a simple, role-based tool that brings existing information to the right person at the right time. Built with accessible, open-source technologies and grounded in a decade of direct hotel experience, it closes a real gap for small and mid-sized properties.

REFERENCES

[1] M. Sanli, "Digital transformations in the hotel industry," Long Island Univ., New York, NY, 2024.

[2] J. S. Park, R. Sandhu, and G.-J. Ahn, "Role-based access control on the Web," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 1, pp. 34–71, 2001.

[3] K. Beck *et al.*, "Manifesto for Agile Software Development," 2001. [Online]. Available: <https://agilemanifesto.org>

[4] A. Banks and E. Porcello, *Learning React*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2020.

[5] M. Grinberg, *Flask Web Development*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.

[6] Information Commissioner's Office, "Guide to the UK GDPR," ICO, Wilmslow, U.K., 2023.

[7] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proc. ACM CHI'90*, Seattle, WA, 1990.