

# HOSTEL ROOM ALLOCATION AND MAINTENANCE SYSTEM

## PROBLEM DEFINITION

### Context

Institutions face recurring challenges assigning hostel rooms fairly and efficiently, managing waitlists and swaps and tracking maintenance issues. Manual processes (spreadsheets, emails, paper forms) lead to allocation errors, delays, lack of transparency, and poor maintenance turnaround.

### Core Problem

There is no streamlined, user-friendly system that enables students to request rooms and maintenance, and enables staff to allocate rooms and manage work orders with auditability and timely communication.

### Goals (Prototype-Level)

- Validate a simple, clear workflow for room allocation and maintenance handling.
- Demonstrate transparency for students (request status, notifications).
- Ensure staff can perform allocations and update work orders quickly.

### Stakeholders

1. Students
2. Wardens/Hostel Staff

### 3. Administration (Facilities/Accommodation Office)

#### **Functional Requirements (Prototype):**

1. Students can: request rooms, view allocation status, submit maintenance requests.
2. Staff can: allocate rooms, manage waitlists/swaps (basic), update maintenance work orders and statuses.
3. System displays confirmations, errors, and current status for each action.

#### **Nonfunctional Requirements (Prototype):**

1. Usability: 3 key tasks completable by first-time users without guidance.
2. Learnability: Consistent components and predictable navigation.
3. Accessibility: Sufficient color contrast and keyboard navigation on main paths.

#### **Risks & Mitigations:**

1. Ambiguous allocation rules → Mitigation: limit to a simple, stated rule for prototype.
2. Misaligned expectations → Mitigation: early usability tests and written assumptions.

#### **Acceptance Criteria (Prototype):**

1. A clickable prototype covers: room request, allocation, maintenance handling.
2. Each flow includes at least one error and one success state.

3. Usability testing with 5 participants yields >80% task completion.

### **Success Measures:**

- Positive stakeholder feedback on clarity of flows.
- Measurable reduction in steps/time vs. current process (observed in tests).

---

Document Version: 1.0

Last Updated: December 10, 2025

Course: Software Engineering

Date: December 10, 2025

### **Team Members:**

- Fabilous Lashidi (Project Manager)
- Zvinaishe Marume (System Architect)
- Promise Siafwiyo (UI Designer)
- Tapiwa Chigome (UX Designer)
- Keith Mutabvuri (Researcher)
- Carlton Kampota (Frontend Developer)
- Tivonge Kambarani (Documentation Specialist)
- Tinotenda Gozi (System Analyst)
- Gufe Makomborero (Quality Assurance)

Author: Software Engineering Team