

Project Summary Report

Hostel Room Allocation and Maintenance Management System

Course: Software Engineering

Date: December 10, 2025

Team: 9 Members

Executive Summary

This project presents a web-based prototype for managing hostel room allocation and maintenance requests in educational institutions. The system addresses the inefficiencies of manual processes by providing a streamlined, user-friendly interface for students and staff to manage room allocations and maintenance requests.

1. Project Overview

1.1 Problem Statement

Educational institutions face challenges in efficiently managing hostel room allocations and maintenance requests. Current manual processes using spreadsheets, emails, and paper forms lead to:

- Allocation errors and delays
- Lack of transparency for students
- Poor maintenance tracking and turnaround times
- No centralized system for communication

1.2 Solution

A web-based prototype system that provides:

- Streamlined room allocation workflow
- Transparent status tracking for students
- Efficient maintenance request management for staff
- Real-time status updates and notifications

1.3 Aim of the Project

Primary Aim:

To develop a web-based prototype that demonstrates streamlined workflows for hostel room allocation and maintenance management in educational institutions, providing a user-friendly interface that showcases how such a system can enhance efficiency, transparency, and user satisfaction for students and staff.

Specific Aims:

1. Demonstrate Room Allocation Workflow

- Provide an intuitive interface for students to submit room requests with preferences
- Enable staff to review and allocate rooms through a clear interface
- Show status tracking throughout the allocation process (simulated)
- Demonstrate transparent communication of allocation decisions

2. Demonstrate Maintenance Management Workflow

- Facilitate easy submission of maintenance requests by students
- Enable staff to view and update work order status systematically
- Show status updates to students regarding their maintenance requests
- Support categorization of maintenance issues

3. Improve User Experience

- Provide students with visibility into their request status
- Enable clear communication through the interface
- Display transparent allocation status and information
- Create a centralized platform for both students and staff

4. Validate System Concept

- Demonstrate the feasibility of the proposed solution
- Validate core workflows through user testing
- Gather feedback on interface design and usability
- Show value proposition to stakeholders

1.4 Objectives

Functional Objectives (Prototype):

1. Room Request Management

- Enable students to submit room requests with preferences (room type, floor, building, etc.)
- Allow students to view and track the status of their requests

- Support multiple request statuses: Pending, Allocated, Rejected
- Display allocation details when a decision is made

2. Room Allocation Management

- Enable staff to view pending room requests in an organized interface
- Allow staff to allocate rooms to students through the interface
- Support rejection of requests with reason display
- Show allocation status updates

3. Maintenance Request System

- Enable students to submit maintenance requests with descriptions and urgency levels
- Allow categorization of maintenance issues (plumbing, electrical, furniture, etc.)
- Display status updates on maintenance requests to students

4. Work Order Management

- Enable staff to view all maintenance requests and work orders
- Allow staff to update work order status (Open, In Progress, Completed)
- Display work order information and status changes

5. User Interface and Experience

- Provide role-based interfaces for students and staff
- Ensure intuitive navigation and clear workflows
- Implement responsive design for different screen sizes
- Maintain consistent design language throughout the prototype

Non-Functional Objectives (Prototype):

1. Usability

- Achieve >80% task completion rate in usability testing
- Ensure key tasks are completable by first-time users without guidance
- Maintain consistent components and predictable navigation
- Provide clear feedback for all user actions

2. Accessibility

- Ensure sufficient color contrast for readability
- Support keyboard navigation on main user paths
- Follow basic web accessibility practices

3. Performance

- Ensure quick response times for user interactions
- Optimize page load times for the prototype
- Provide smooth user experience

Prototype-Specific Objectives:

1. Validation Objectives

- Validate core workflows through functional prototype
- Gather user feedback on interface design and usability
- Test key user journeys with target users
- Identify areas for improvement

2. Demonstration Objectives

- Demonstrate feasibility of proposed solution
- Show value proposition to stakeholders
- Provide visual representation of system capabilities
- Support stakeholder presentations and approvals

2. Project Scope

2.1 Current Phase (Prototype)

- **Type:** Frontend-only prototype (HTML/CSS/JavaScript)
- **Purpose:** Validate user workflows and gather feedback
- **Limitations:** No backend, no database, simulated interactions

2.2 Features Implemented

1. Student Portal:

- Room request submission
- Status tracking (Pending/Allocated/Rejected)
- Maintenance request submission

2. Staff Portal:

- Room allocation interface
- Work order management
- Status updates for maintenance requests

3. User Interface:

- Modern dark theme with glassmorphism design
- Custom modal notifications
- Responsive layout
- Smooth animations and transitions

3. Technical Implementation

3.1 Technology Stack

- **Frontend:** HTML5, CSS3, JavaScript
- **Architecture:** Static files, component-based design
- **Design:** Responsive, mobile-friendly

3.2 System Structure

```
/  
└── index.html (Login/Role Selection)  
└── styles.css (Global Styles)  
└── student/  
    ├── dashboard.html  
    ├── request-room.html  
    ├── request-status.html  
    └── maintenance.html  
└── staff/  
    ├── dashboard.html  
    ├── allocate-room.html  
    └── work-orders.html
```

3.3 Project Repository and Access

GitHub Repository: <https://github.com/fabber04/HOSTEL-ROOM-ALLOCATION-MAINTENANCE>

Live Prototype (GitHub Pages): <https://fabber04.github.io/HOSTEL-ROOM-ALLOCATION-MAINTENANCE/>

The prototype is available online for demonstration and testing purposes. All source code, documentation, and project files are maintained in the GitHub repository.

4. Key Achievements

4.1 User Experience

- Intuitive navigation and clear workflows

- Professional, modern UI design
- Custom modal popups replacing browser alerts
- Consistent design language throughout

4.2 Functionality

- Complete room allocation workflow
- Maintenance request management
- Status tracking and updates
- Error and success state handling

4.3 Documentation

- Problem definition and requirements
- System architecture documentation
- Use cases and user stories
- Database design for future implementation
- Presentation materials

5. Project Methodology

5.1 Approach

- **Code-First Prototyping:** Built functional prototype before detailed design
- **Iterative Development:** Continuous refinement based on feedback
- **User-Centered Design:** Focus on usability and user experience

5.2 Team Roles

- **Project Manager:** Fabilous Lashidi
- **System Architect:** Zvinaishe Marume
- **UI Designer:** Promise Siafwiyoo
- **UX Designer:** Tapiwa Chigome
- **Researcher:** Keith Mutabvuri
- **Frontend Developer:** Carlton Kampota
- **Documentation Specialist:** Tivonge Kambarani

- **System Analyst:** Tinotenda Gozi
- **Quality Assurance:** Gufe Makomborero

6. Testing and Validation

6.1 Usability Testing

- Tested with target users (students and staff)
- Validated key workflows
- Gathered feedback on UI/UX
- Achieved >80% task completion rate

7. Challenges and Solutions

7.1 Challenges Faced

1. **Complex Workflows:** Multiple user roles and states
 - *Solution:* Clear separation of concerns, role-based views
2. **User Experience:** Making complex processes simple
 - *Solution:* Intuitive navigation, clear status indicators
3. **Design Consistency:** Maintaining visual coherence
 - *Solution:* CSS variables, reusable components

7.2 Lessons Learned

- User feedback is crucial for design decisions
- Simplicity beats complexity in UI design
- Prototyping helps identify issues early
- Clear documentation saves time in development

8. Future Enhancements

8.1 Phase 2 Features

- Backend implementation with database
- User authentication and authorization

- Email notifications
- Advanced filtering and search
- Analytics dashboard
- Mobile application

8.2 Scalability

- Multi-institution support
- Advanced allocation algorithms
- Integration with existing systems
- API for third-party integrations

9. Conclusion

This prototype successfully demonstrates the feasibility and value of a web-based system for hostel room allocation and maintenance management. The project validates the core workflows, provides a solid foundation for future development, and addresses real-world problems faced by educational institutions.

10. Deliverables

1. Functional Prototype (HTML/CSS/JavaScript)

- Available online via GitHub Pages
- Source code in GitHub repository

2. Documentation

- Problem Statement and Objectives
- Problem Definition document
- System Architecture documentation
- Use Cases document
- Database Design documentation
- Research Documentation
- Project Summary Report (this document)
- README and User Guide

3. Presentation Materials

- PowerPoint Presentation slide deck
- Live prototype demonstration

4. Project Repository

- GitHub repository with complete source code
 - Version control history
 - Issue tracking and project management
-

Project Status: Prototype Complete

Team: Software Engineering Course - Group Project

Repository: <https://github.com/fabber04/HOSTEL-ROOM-ALLOCATION-MAINTENANCE>

Live Demo: <https://fabber04.github.io/HOSTEL-ROOM-ALLOCATION-MAINTENANCE/>