System Architecture Document

Hostel Room Allocation and Maintenance Management System

Course: Software Engineering
Date: December 10, 2025

---

1. System Overview

1.1 Purpose

This document describes the architecture of the Hostel Room Allocation and Maintenance Management System prototype. The system is designed to streamline room allocation and maintenance request processes for educational institutions.

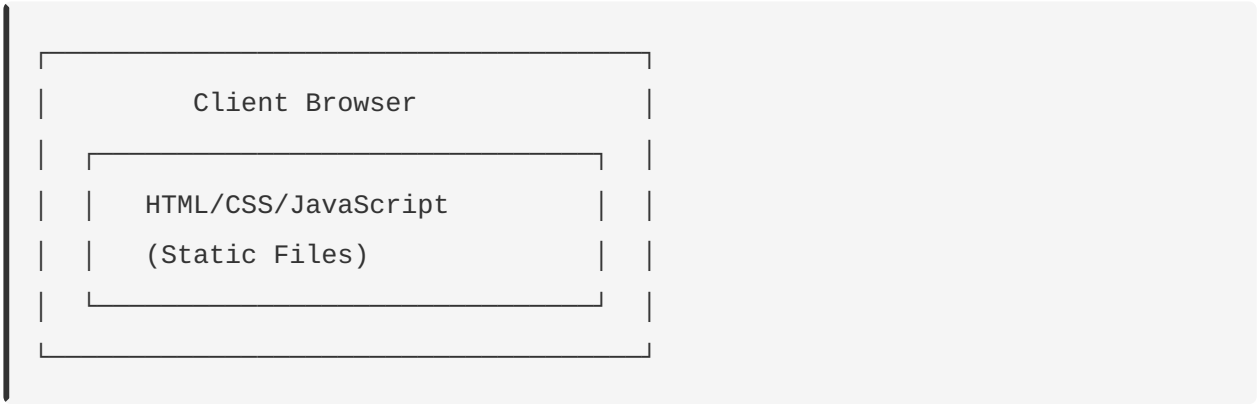
1.2 System Type

- **Current Phase**: Frontend Prototype (Static HTML/CSS/JavaScript)
- **Purpose**: Validate user workflows and demonstrate system concept

---

2. Architecture Overview

2.1 Current Architecture (Prototype)



Key Characteristics:

- **Frontend Only**: HTML5, CSS3, JavaScript
- **No Backend**: All interactions are simulated
- **No Database**: Data is hardcoded in HTML/JavaScript
- **Static Files**: All files are served directly to the browser

---

3. System Components

3.1 Frontend Components

Entry Point:

- `index.html` - Role selection (Student/Staff simulation)

Student Module:

- `student/dashboard.html` - Overview and navigation
- `student/request-room.html` - Room allocation request form
- `student/request-status.html` - View allocation status
- `student/maintenance.html` - Submit maintenance requests

Staff Module:

- `staff/dashboard.html` - Overview of pending items
- `staff/allocate-room.html` - Room allocation interface
- `staff/work-orders.html` - Maintenance request management

Shared Components:

- `styles.css` - Global styling and theme
- Navigation components - Consistent navigation across all pages
- Modal components - Custom modal notifications

3.2 Design Patterns

- **Component-Based Design**: Reusable UI components (buttons, cards, forms)
- **Consistent Styling**: CSS variables for theme management
- **Modular Structure**: Separation of concerns (HTML, CSS, JS)

---

4. Data Flow

4.1 Room Allocation Flow

```
Student → Request Form → Submit → Status Page (Pending)
Staff → View Requests → Allocate → Status Updated (Allocated)
```

4.2 Maintenance Flow

```
Student → Maintenance Form → Submit → Ticket Created
Staff → Work Orders → Update Status → Ticket Updated
```

---

5. Technology Stack

Current Implementation:

- **HTML5**: Semantic markup
- **CSS3**: Styling, animations, responsive design
- **JavaScript**: Client-side interactions and simulations

---

6. Current Limitations

- Static files only - no server-side processing
- No data persistence - all data is simulated
- No concurrent user support
- Limited to demonstration and validation purposes

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