Problem Statement, Aim, and Objectives

Hostel Room Allocation and Maintenance Management System

Course: Software Engineering
Date: December 10, 2025
Team: Software Engineering Course - Group Project

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)

---

1. Problem Statement

1.1 Context

Educational institutions worldwide face significant challenges in managing hostel accommodation for their students. The current state of hostel room allocation and maintenance management relies heavily on manual, paper-based processes and fragmented digital tools such as spreadsheets, email communications, and physical forms. These traditional methods have proven to be inefficient, error-prone, and lack the transparency and responsiveness required in modern educational environments.

1.2 Core Problem

There is no streamlined, user-friendly, centralized system that enables:
- **Students** to easily request rooms, track their allocation status, and submit maintenance requests with clear visibility into the process

- **Staff** to efficiently allocate rooms, manage waitlists, and handle maintenance work orders with proper auditability and timely communication
- **Administration** to maintain accurate records, generate reports, and make data-driven decisions regarding accommodation management

1.3 Specific Challenges

The current manual processes lead to several critical issues:

For Students:
- **Allocation Errors and Delays**: Manual processing results in mistakes, duplicate allocations, and significant delays in room assignment
- **Lack of Transparency**: Students have no visibility into the status of their requests, available options, or allocation timelines
- **Poor Communication**: Inconsistent and delayed communication regarding allocation decisions and maintenance request status
- **Difficulty Tracking Requests**: No centralized way to track room requests or maintenance issues from submission to resolution

For Staff:
- **Time-Consuming Processes**: Manual allocation and tracking require excessive administrative time
- **Inefficient Workflow**: Difficulty managing multiple requests simultaneously without a structured system
- **Limited Organization**: No systematic approach to tracking work orders, priorities, and completion status
- **Challenges in Fair Allocation**: Difficulty maintaining consistent allocation policies and ensuring fairness without proper tools

For Administration:
- **No Centralized Data**: Information is scattered across multiple spreadsheets, emails, and paper records
- **Limited Reporting Capabilities**: Difficulty generating reports, analyzing trends, and making informed decisions
- **Compliance and Audit Challenges**: Lack of proper audit trails and documentation for allocation decisions
- **Resource Planning Difficulties**: Inability to effectively plan and allocate resources based on historical data

1.4 Impact

These challenges result in:
- Reduced student satisfaction and increased frustration
- Inefficient use of staff time and resources
- Potential for unfair allocation practices
- Poor maintenance response times affecting student living conditions
- Lack of accountability and transparency in decision-making processes
- Difficulty in scaling operations as institutions grow

---

2. Aim of the System

2.1 Primary Aim

To develop a comprehensive, web-based system that streamlines and automates the processes of hostel room allocation and maintenance management in educational institutions, providing a centralized platform that enhances efficiency, transparency, and user satisfaction for all stakeholders.

2.2 Specific Aims

2.2.1 Streamline Room Allocation Process
- Provide an intuitive interface for students to submit room requests with preferences
- Enable staff to efficiently review, process, and allocate rooms based on established policies
- Automate status tracking and notifications throughout the allocation lifecycle
- Maintain a clear audit trail of all allocation decisions

2.2.2 Enhance Maintenance Management
- Facilitate easy submission of maintenance requests by students
- Enable staff to create, track, and update work orders systematically
- Provide real-time status updates to students regarding their maintenance requests
- Support prioritization and categorization of maintenance issues

2.2.3 Improve Transparency and Communication
- Provide students with real-time visibility into their request status
- Enable clear, timely communication between students and staff
- Display transparent allocation policies and decision criteria
- Maintain comprehensive records accessible to authorized users

2.2.4 Support Administrative Decision-Making

- Generate reports and analytics on allocation patterns and maintenance trends
- Provide data for resource planning and capacity management
- Maintain audit trails for compliance and accountability
- Support evidence-based policy development and refinement

2.2.5 Create Scalable Foundation

- Design architecture that supports future enhancements and integrations
- Ensure system can scale to accommodate growing user bases and institutions
- Provide foundation for advanced features such as automated allocation algorithms
- Enable integration with existing institutional systems

---

3. Objectives

3.1 Primary Objectives

3.1.1 Functional Objectives

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 in real-time
   - Support multiple request statuses: Pending, Under Review, Allocated, Rejected
   - Provide students with allocation details once a decision is made

2. **Room Allocation Management**
   - Enable staff to view all pending room requests in an organized interface
   - Allow staff to allocate rooms to students based on availability and preferences
   - Support rejection of requests with reason documentation
   - Maintain allocation history and records

3. **Maintenance Request System**
   - Enable students to submit maintenance requests with descriptions and urgency levels
   - Allow categorization of maintenance issues (plumbing, electrical, furniture, etc.)
   - Support attachment of photos or documents (future enhancement)
   - Provide students with status updates on their maintenance requests

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, On Hold)

- Support assignment of work orders to maintenance personnel
- Track completion times and resolution details

5. **User Interface and Experience**
   - Provide role-based interfaces for students and staff
   - Ensure intuitive navigation and clear workflows
   - Implement responsive design for mobile and desktop access
   - Maintain consistent design language throughout the system

3.1.2 Non-Functional Objectives

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
   - Design for screen reader compatibility (future implementation)
   - Follow web accessibility best practices

3. **Performance**
   - Ensure quick response times for user interactions
   - Optimize page load times
   - Design for efficient data handling (future backend implementation)

4. **Scalability**
   - Design architecture that supports future growth
   - Ensure database design can handle increasing data volumes
   - Plan for multi-institution support (future enhancement)

3.1.3 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 before full implementation

2. **Demonstration Objectives**
   - Demonstrate feasibility of proposed solution
   - Show value proposition to stakeholders

- Provide visual representation of system capabilities
- Support stakeholder presentations and approvals

3. **Foundation Objectives**
   - Establish technical foundation for future development
   - Create reusable UI components and design patterns
   - Document system architecture and design decisions
   - Prepare for backend integration and database implementation

3.2 Success Criteria

3.2.1 Prototype Phase Success Criteria

- Functional prototype covering all core workflows (room request, allocation, maintenance)
- Each workflow includes both success and error state handling
- Usability testing with target users achieves >80% task completion rate
- Positive stakeholder feedback on clarity and usability of flows
- Measurable reduction in steps/time compared to current manual processes
- Complete documentation of system design and architecture

3.2.2 User Satisfaction Criteria

- Students can successfully submit room requests without assistance
- Students can easily track their request status
- Staff can efficiently process allocations and work orders
- All users report improved experience compared to current processes
- System receives positive feedback on design and functionality

3.2.3 Technical Criteria

- Clean, maintainable code structure
- Responsive design works on multiple device sizes
- Consistent design system implemented
- Error handling and validation in place
- Documentation complete and comprehensive

3.3 Long-Term Objectives (Future Phases)

1. **Backend Implementation**
   - Implement secure user authentication and authorization

- Develop database schema and data persistence
- Create RESTful API for frontend-backend communication
- Implement data validation and business logic

2. **Advanced Features**
   - Email and SMS notification system
   - Advanced filtering and search capabilities
   - Analytics dashboard with reporting tools
   - Automated allocation algorithms based on preferences and policies

3. **Integration and Expansion**
   - Integration with student information systems
   - Mobile application development
   - Multi-institution support
   - API for third-party integrations

4. **Enhancement Features**
   - Room swap functionality
   - Waitlist management with priority queuing
   - Maintenance scheduling and resource allocation
   - Predictive maintenance capabilities

---

4. Scope and Limitations

4.1 Current Phase Scope (Prototype)

The current prototype focuses on:
- Frontend-only implementation (HTML/CSS/JavaScript)
- Simulated interactions without backend or database
- Core workflows: room request, allocation, and maintenance management
- User interface validation and usability testing

4.2 Current Limitations

- No persistent data storage
- No real authentication or authorization
- Simulated business logic and policies
- No email or notification system

- Limited to basic workflows
- No reporting or analytics capabilities

4.3 Out of Scope (Current Phase)

- Backend implementation
- Database integration
- Real authentication system
- Email notifications
- Advanced features (swaps, waitlists, analytics)
- Mobile application
- Multi-institution support

---

5. Stakeholders

5.1 Primary Stakeholders

1. **Students**
   - Need: Easy room request process, status visibility, maintenance request submission
   - Benefit: Reduced frustration, faster responses, transparency

2. **Hostel Staff/Wardens**
   - Need: Efficient allocation tools, organized work order management
   - Benefit: Time savings, better organization, improved service delivery

3. **Administration/Facilities Office**
   - Need: Data collection, reporting, audit trails
   - Benefit: Better decision-making, compliance, resource planning

5.2 Secondary Stakeholders

- Maintenance Personnel
- IT Department
- Institutional Management
- Parents/Guardians (indirect)

---

6. Expected Outcomes

6.1 Immediate Outcomes (Prototype Phase)

- Validated user workflows and interface design
- Stakeholder buy-in and approval for full development
- Clear requirements for Phase 2 implementation
- Foundation for future development

6.2 Long-Term Outcomes (Full Implementation)

- Reduced allocation processing time
- Improved student satisfaction
- Better maintenance response times
- Enhanced transparency and accountability
- Data-driven decision-making capabilities
- Scalable solution for institutional growth

Document Version: 1.0
Last Updated: December 10, 2025
Status: Complete