

Software Design Specifications

for Hostel Management System (Python Django + SQLite)

Prepared by: Nandan Murari

Position: Software Developer

Document Information

Title: Hostel Management System Software Design Specification

Project Manager: Nandan Murari

1 INTRODUCTION

1.1 PURPOSE

This document outlines the Software Design Specification for the Hostel Management System implemented using Django and SQLite. It defines the software architecture, data models, interfaces, and design goals that guide the development process.

1.2 SCOPE

This SDS covers the complete design of a hostel management web application that supports room allocation, student registration, fee processing, complaints, and communication modules through a Django-based platform with SQLite as the underlying database.

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

HMS - Hostel Management System

MVC - Model View Controller

ORM - Object Relational Mapper

JWT - JSON Web Token

1.4 REFERENCES

1. Hostel Management System SRS v1.0
2. Django Documentation
3. SQLite Documentation

2 USE CASE VIEW

2.1 USE CASE

Use Case 1: Student Registration

Actors: Student

Description: Allows new students to create an account and login.

Use Case 2: Room Booking

Actors: Student, Warden

Description: Enables students to book available rooms and wardens to approve allocation.

3 DESIGN OVERVIEW

3.1 DESIGN GOALS AND CONSTRAINTS

Goals: Scalability, Modularity, Security, Maintainability

Constraints: Must use Django and SQLite; lightweight deployment for university intranet.

3.2 DESIGN ASSUMPTIONS

All users have access to the internet and a browser. The application will run on a university-managed server.

3.3 SIGNIFICANT DESIGN PACKAGES

1. users - authentication and profile management
2. rooms - room inventory, booking, and allocation
3. payments - invoicing and transactions
4. core - settings, logging

3.4 DEPENDENT EXTERNAL INTERFACES

- Email API for password resets and notifications
- Optional: Payment Gateway API for online transactions

3.5 IMPLEMENTED APPLICATION EXTERNAL INTERFACES

- REST API for user authentication, booking, and payment management
- Admin interface for managing students and rooms

4 LOGICAL VIEW

4.1 DESIGN MODEL

The system uses Django's MVC architecture. Key models include User, Student, Room, Booking, Payment, Complaint.

4.2 USE CASE REALIZATION

UC1: Registration -> AuthView -> User model (create user, email verification)

UC2: Room Booking -> RoomView -> Booking model

5 DATA VIEW

5.1 DOMAIN MODEL

Entities: User, Student, Room, Booking, Payment, Complaint, Notification

5.2 DATA MODEL (PERSISTENT DATA VIEW)

Tables: auth_user, students, rooms, bookings, payments, complaints

Relationships defined using Django's ORM

5.2.1 DATA DICTIONARY

Example: Room Table - Fields: room_id (PK), number, building, floor, capacity, type, status, fee

6 EXCEPTION HANDLING

Handled using Django's middleware and custom exception classes. Errors are logged and users see custom error pages.

7 CONFIGURABLE PARAMETERS

- DEBUG - enables debugging mode
- DATABASES - SQLite config in settings.py
- ALLOWED_HOSTS - server IP/domain list

8 QUALITY OF SERVICE

8.1 AVAILABILITY

System is expected to maintain 99.5% uptime. SQLite keeps minimal dependencies.

8.2 SECURITY AND AUTHORIZATION

RBAC using Django groups, form input validation, CSRF protection, encrypted passwords with PBKDF2.

8.3 LOAD AND PERFORMANCE IMPLICATIONS

Optimized database queries with Django ORM. Application scales for 300+ concurrent users on SQLite with vertical scaling option.

8.4 MONITORING AND CONTROL

Logging, error emails to admins, and access tracking using Django admin.