**MALAWI COLLEGE OF ACCOUNTANCY**



**P.OBox30644**

**Chichiri**

**BLANTYRE3**

**SCHOOL OF COMMERCE**

**APRIL- SEPTEMBER 2025 SEMESTER ASSIGNMENTS (Cover Page)**

**Student Number: MA2815 Student Name: BEAUTY MALINGAMOYO**

**Student Phone Number: 0983701734 Student email Address: MA2815CA.AC.MW**

**Programme: BMIS Year of Study: 3 Semester: 2**

**Module Code: OOP321 Module Name: OBJECT ORIENTED PROGRAMMING**

**Mode of Study: WEEKDAY Attempt: (Normal or Repeat) NORMAL**

**ASSIGNMENT DUEDATE: 27TH JUNE**

**LECTURER’SNAME: LOVEMORE NGWIRA**

SOFTWARE REQUIREMENTS SPECIFICATION

PROJECT TTILE: KUMBALE LODGE MANAGEMENT SYSTEM

Table of Contents

[Software Requirements Specification for KUMBALE LODGE MANAGEMENT SYSTEM 2](#_Toc190684813)

[Introduction 2](#_Toc190684814)

[Scope 2](#_Toc190684815)

[Product Perspective 2](#_Toc190684816)

[Product Features 2](#_Toc190684817)

[Functional Requirements 2](#_Toc190684818)

[Non-Functional Requirements 3](#_Toc190684819)

[USE CASE DIAGRAM 4](#_Toc190684820)

[Actors 4](#_Toc190684821)

[Use Cases 4](#_Toc190684822)

[Class Diagram 6](#_Toc190684823)

[Classes and Relationships 6](#_Toc190684824)

[Sequence Diagram 8](#_Toc190684825)

[Sequence 8](#_Toc190684826)

[Entity Relationship Diagram (ERD) 9](#_Toc190684827)

[Entities and Relationships 9](#_Toc190684828)

[Activity Diagram 11](#_Toc190684829)

[Activities 11](#_Toc190684830)

INTRODUCTION

PURPOSE

This SRS outlines the functional and non-functional requirements for the KUMBALE Lodge Management System, which is being developed to automate and streamline operational processes at KUMBALE Lodge, located in Mzuzu. The goal is to improve booking, food ordering, check-in/check-out, guest record management, and report generation. (H.M, 2012)

SCOPE

The system will provide a web-based platform accessible to both lodge staff and customers. Customers will be able to (hostmann, 2012)

* Book accommodation
* Order food

STAFF DUTIES

* Generate analytical reports
* Check guest in and check out
* Handle transactions
* Manage bookings

INTENDED AUDIENCE

* System Developers
* Lodge Management
* Software Testers
* Project Stakeholders

1. Product Perspective (keogh, 2015)

This system replaces the current manual operations at KUMBALE Lodge. It is a self-contained application using the following langauges.

Python, HTML, CSS, JavaScript, and MySQL, hosted locally or online.

2. Product Features

* Online room reservation
* Online food ordering
* Guest check-in/check-out
* Invoice and transaction management
* Role-based access (Admin, Staff, Guest)
* Real-time reporting dashboard

3. User Classes

|  |  |
| --- | --- |
| USER CLASS | DESCRITPTION |
| GUEST | Book rooms, places food orders |
| STAFF | Manages bookings, check-in/out, guest data |
| ADMIN | Full access to all functionalities and reports |

4. OPERATING ENVIRONMENT (hostmann, 2012)

* HTML/CSS/JavaScript for frontend
* Python for backend logic
* MySQL database using XAMPP for hosting
* Google Chrome or Firefox recommended

5. CONSTRAINTS

* Runs only on web browsers
* Requires internet connection or local server
* Payment API not integrated in initial phase

Assumptions and Dependenciens

* Staff are trained to use web-based systems
* Lodge has working internet and server setup

3. SPECIFIC REQUIREMENT

.1 Functional Requirement

|  |
| --- |
| FR1 Users can register and log in securely |
| FR2 Guests can book rooms and view availability |
| FR3 Guests can place food orders |
| FR4 Staff can check guests in and out |
| FR5 System tracks transactions and generates invoices |
| FR6 Staff can generate weekly/monthly reports |
| FR7 System supports role-based access control |
|  |

3.2 Non-Functional Requirements (keogh, 2015)

* SECURITY

The system ensures the confidentiality, integrity and availability of guest information

* USABILITY

The system shall be user-friendly and easy to navigate for both guest and lodge staff

* PERFORMANCE

The system shall respond to user interactions within a reasonable time frame.

Code Requirement

|  |  |
| --- | --- |
| NFR1 | System should support 50+ concurrent users |
| NFR2 | Interface must be mobile-friendly |
| NFR3 | Passwords must be encrypted |
| NFR4 | System should maintain 99.5% uptime |
| NFR5 | Code should follow modular and maintainable practices |

4. INTERFACE REQUIREMENTS (H.M, 2012)

1 Tab-based navigation for staff/admin

* Booking and order forms for guests
* Dashboard widgets for reporting

4.2 Software Interface

* Frontend: HTML/CSS/JavaScript
* Backend: Python
* Database: MySQL (via XAMPP)

5. Use Case Diagram (hostmann, 2012)

Below is the generated use case diagram that represents the interaction between the system and its users:

GUEST

EQDDDWC LODGE STAF

SYSTEM

USE CASE EXPLANATION

Guest interacts with the system to:

* Book Room
* Order Food

Lodge Staff interacts with the system by

* Manage Bookings
* Perform Check-in/Check-out
* Manage Guest Services
* Generate Report

BOOKING.

Booking Id

Guest ID

Room number

Dates

LODGESTAFF.

Staff ID

Name

Role

GUEST.

ID

Name

Contact information

ROOM.

Number

Type

status

SERVICES.

ID

Name

Description

The above diagram its class diagram that provide an overview of the systems classes, their attributes, method and relationships. It helps to understand the object- oriented design of the system. (keogh, 2015)

CLASS DIAGRAM DESCRIPTION (H.M, 2012)

* GUEST this has attributes like ID, name and contact information
* BOOKING this has attributes like booking ID, guest ID, room number and dates
* LODGESTAFF this attributes like staff ID, name and role
* ROOM this has attribute like room number, type and status
* SERVICES this has attributes like service ID, name and description

6.ENTITY RELATIONSHIP DIAGRAM

The ERD depicts the logical structure of the databases, showing entities their attributes and their relationship

ROOM

BOOKING

GUEST

SERVICES

STAFF

THE ENTITY RELATIONSHIP DIAGRAM

* GUEST AND BOOKING( One – many relationship)

A guest can have multiple bookings

* ROOM AND BOOKING(One – many relationship)

A room can be booked several times

* BOOKING AND SERVICES( Many –many relationship)

That is a booking can include multiple services and services can be part of multiple bookings

* STAFF AAND BOOKING( One – many relationship)

A staff member can manage multiple booking

STAFF AND GUEST ( many –many)

Staff members can be dealing with multiple guest (keogh, 2015)

SEQEUNCE DIAGRAM

GUEST

SYSTEM

STAFF

SEQUENCE DIAGRAM

* Guest request room availability
* System checks room availability and responds
* Guest submits booking request
* System creates booking and assign room
* System sends confirmation message to the guest
* Staff receives booking notification
* Guest makes payment
* System updates booking status and sends payment confirmation

ACTIVITY DIAGRAM

Guest logs in

System verifies

Display error when its no

Guest acces if its yes

Guest fill the details

Guest pays the room

Guest submits

System displays error message

System saves the payment

Error and resubmits

System notifies successful

ACTIVITY DIAGRAM (hostmann, 2012)

This will illustrate the workflow and activities involved in managing the lodge.

ACTIVITY DIAGRAM DESCRIPTION

* Guest request room availability
* System checks room availability
* Guest submits booking request
* System processes booking
* System sends confirmation massage to the guest
* Staff prepares for guest arrival and then the guest checks in
* Staff provides services like room cleaning as well as food
* Guest requests check out
* System update booking status

SYSTEM DESIGN DOCUMENT (keogh, 2015)

* This presents the system architecture, components, interfaces and data structures. It is used as a guide for developers during the system implementation as well as reference for future maintenance.

DESIGN OVERVIEW

* My system is designed as a web based system or optionally a desktop or mobile hybrid to manage bookings, guest check in check out, billing staff, rooms, services and reporting

SYSTEM ARCHITECTURE

The system has been divided into three tiers

* Presentation layer
* Business layer
* Data layer

TECHNOLOGIES

* Fronted
* Backend
* Database
* Frame works

COMPONENTS

* Reservation management
* Guest check in and out
* Room management
* Billing and payment
* Staff management
* Reports and analytics

DATA BASE DESIGN

This shows the schema overview of the system which include the entities which has primary keys, foreign keys and relationships

INTERFACE DESIGN

* User interface like dashboard, booking form room list
* Admin interface like staff management
* Optional interface like intended for mobile apps

SEQUENCE AND ACTIVITY DIAGRAM

This describes the visualized system interactions over time and decision path

SECURITY DESIGN

Tis include pass word hashing, role based access , input validation and session handling and time out

PERFORMANCE CONSIDERATION

This is where there is caching frequent data, optimized database queries and responsive design for database use

LIMITATION AND ASSUMPTIONS

* Internet connectivity requires full functionality
* Staff must be trained in basics and use
* Payment gateway integration depends on third party

FUTURE ENHANCEMENTS

* Mobile app integration
* Loyalty point module

CONCLUSION

KUMBAKE lodge management system provides a detailed design framework for translating the specified requirements into a fully functional and maintainable software solution. It describes the system architecture, components, data structures, interface design and interaction flows that will guide the system development

REFERENCE

Deitel P.J. & Deitel H.M. (2012), Java How to Program, 9th Edition, Prentice Hall, India

Hostmann C.S (2012) Core Java for the impatient, 9th Edition, Addison Wesley

Schildt H. (2014), Java: A beginner’s Guide, 8th Edition, McGraw Hill

Flanagan D, Farley J., Crawford W & Mangnusson K.( ), Java Enterprise in a nutshell, OReill

ISRD Group (2007) Introduction to Object Oriented Programming through Java, McGraw Hill

Keogh J(2015), J2EE: The Complete Reference, McGraw Hill. Schildt H. (2014), Java: The Complete Reference, 9th Edition, McGraw Hill