**COMP 3059 – Capstone Project I**

**Software Requirements Analysis and Design Assignment**

This assignment is an overview to gather the software needs with requirements analysis and help to proceed with the design.

The requirements analysis helps to break down functional and non-functional requirements to a basic design view to provide a clear system development process framework. It involves various entities, including business, stakeholders and technology requirements.

The design is the activity following requirements specification and before programming. Software design usually involves problem solving and planning a software solution.

To work on this assignment you could use the references and a sample template given below. The sample template can be customised to suit the nature of your project.

Reference Readings/Example:

<http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf>

[www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc](http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc)

Source for this template:

[www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc](http://www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc)

**Requirements Analysis and Design**

**<Queen of Apostles Website>**

**<Partner Organization (if any)>**

|  |  |
| --- | --- |
| Industry Partner | Queen of Apostles |
| Primary Instructor | Anjana Shah |
| Team Member | Siam Hasan |
| Team Member | Jason Gunawan |
| Team Member | Jason Opoku |
| Team Member | Riyad Youssef |
| Team Member | Bilal Youssef |

**Table of Contents**

1.0 Introduction .............................................................................. 4

1.1 Purpose ................................................................................... 4

1.2 Scope ....................................................................................... 4

2.0 System Overview ...................................................................... 4

2.1 Project Perspective ................................................................... 4

2.2 System Context ....................................................................... 4-5

2.3 General Constraints ................................................................. 5

2.4 Assumptions and Dependencies ............................................ 5-6

3.0 Functional Requirements ........................................................... 6

3.1 Functional Requirements or Features ........................................ 6

3.2 Use Cases .................................................................................. 6

3.3 Data Modelling and Analysis .................................................... 7-9

3.4 Process Modelling ...................................................................... 10

4.0 Non-Functional Requirements .................................................... 10

4.1 Performance ........................................................................... 10-11

4.2 Reliability ................................................................................... 11

4.3 Availability .................................................................................. 11

4.4 Security ...................................................................................... 11

4.5 Maintainability ....................................................................... 11-12

4.6 Portability ................................................................................ 12

5.0 Logical Database Requirements .............................................. 12

5.1 Data Formats ........................................................................ 12-13

5.2 Storage Capabilities ................................................................. 13

5.3 Data Retention ......................................................................... 13

5.4 Data Integrity ............................................................................ 13

5.5 Security Measures .................................................................... 13

6.0 Other Requirements ................................................................. 14

7.0 Approval ................................................................................... 14

# 1.0 Introduction

The Introduction section provides an overview of the system requirements, utilizing software requirements analysis and design to define the scope of the system. This document outlines the high-level software requirements for the proposed system, detailing what the system will accomplish for stakeholders and the development team without specifying how it will be implemented.

## 1.1 Purpose

The document’s purpose can be stated as defining the high-level software requirements for the Queen of Apostles Renewal Centre’s system, focusing on streamlining operational processes and enhancing guest experience.

## 1.2 Scope

**In Scope**: Guest registration, room management, scheduling, departmental communication, and payment processing.

**Out of Scope**: Lobby display integration, third-party software compatibility, legacy data migration, and non-digital booking management.

# 2.0 System Overview

The software is being developed to improve operational efficiency and guest experience at the Queen of Apostles Renewal Centre.

## 2.1 Project Perspective

The system is a new, self-contained software solution specifically designed for the Queen of Apostles Renewal Centre. It is being developed to automate and modernize various operational processes, including guest registration, room assignment, scheduling, departmental communication, and payment processing.

## 2.2 System Context

* This system is intended to address inefficiencies in the current manual processes for managing reservations, billing, and inter-departmental communication.
* It is designed from the ground up with the unique needs of the Renewal Centre in mind, focusing on improving the overall guest experience and operational efficiency.
* While it does not replace an existing software system, it will replace paper-based or semi-automated workflows, creating a cohesive digital environment for the Centre’s management needs.

## 2.3 General Constraints

## Technical Constraints :Given the Centre’s potential future needs, the system should be designed with scalability in mind but should avoid unnecessary complexity at this stage.

* **Operational Constraints:**The system must be user-friendly and require minimal training for staff, as it will be used by individuals with varying levels of technical expertise.
* **Business Constraints:**The software must comply with Ontario’s data privacy regulations, especially for guest personal and payment information.

## 2.4 Assumptions and Dependencies

*2.4.1 Assumptions:*

* Staff will have basic technical skills and will receive training on using the software.
* Guests and staff will rely primarily on the software’s web interface for booking, managing reservations, and communicating needs.
* The Centre will provide access to necessary resources, like devices for on-site staff to access the system.

*2.4.2 Dependencies:*

* **Payment Gateway Integration**: Successful implementation depends on a reliable third-party payment gateway for secure transactions.
* **Data Security Compliance**: Compliance with data security and privacy laws is essential. The software will need to work within these guidelines to ensure the safe handling of guest data.
* **Wi-Fi/Internet Availability**: The Centre requires stable internet access for the software to function properly, as it will likely be hosted on the cloud or a centralized server.

## 3.0 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

### 3.1 <Functional Requirement or Feature #1>

* Guest registration and management.
* Scheduling for room assignments, meal planning, and conference space.
* Departmental communication to facilitate seamless interactions among staff.
* Payment processing, including deposit handling and invoicing.
* Reporting and database functionalities for guest records and reservation history.

## 3.2 Use Cases

**Use Case #1:** Guest Registration

* **Actors:** Guest, Reception Staff
* **Description:** Guest registers for a retreat, selects dates and preferences, confirms booking.

**Use Case #2:** Room Assignment

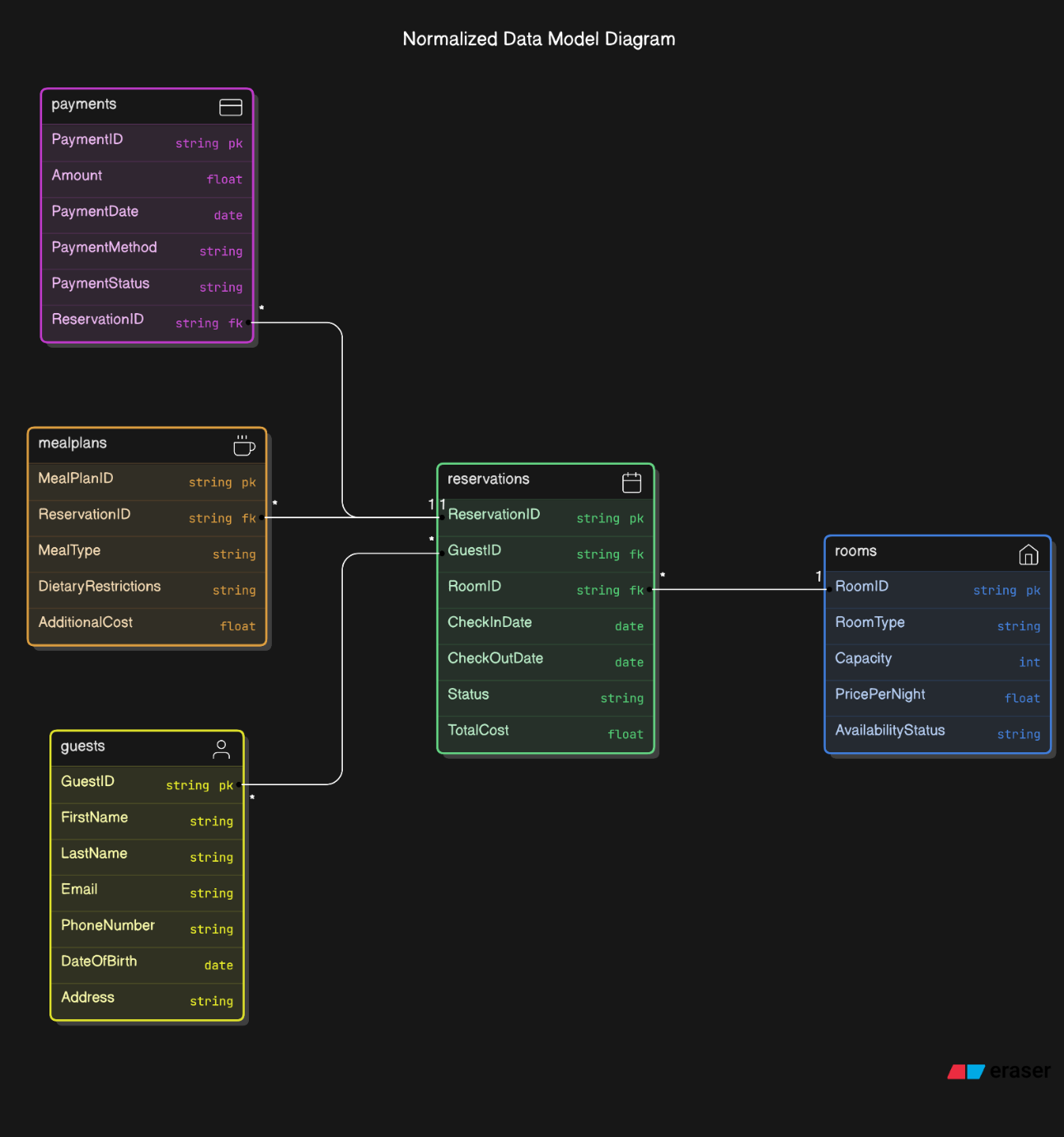
* **Actors:** Reception Staff, Housekeeping
* **Description:** System assigns rooms based on availability and updates cleaning schedules.

**Use Case #3:** Payment Processing

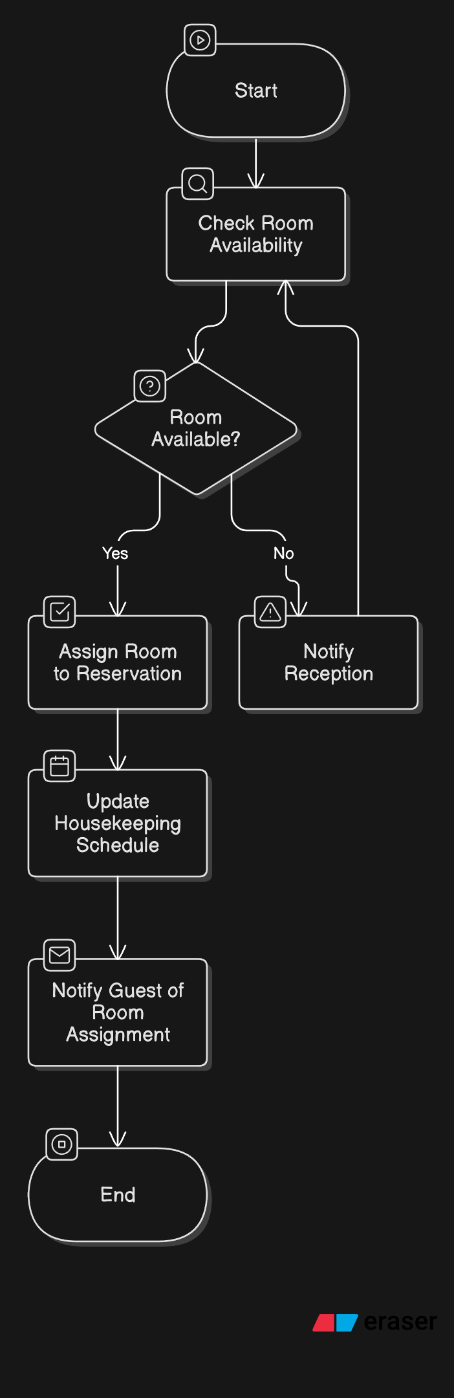
* **Actors:** Guest, Finance Department
* **Description:** Guest pays a deposit or final balance, and the system updates the payment status.

**3.3 Data Modelling and Analysis**

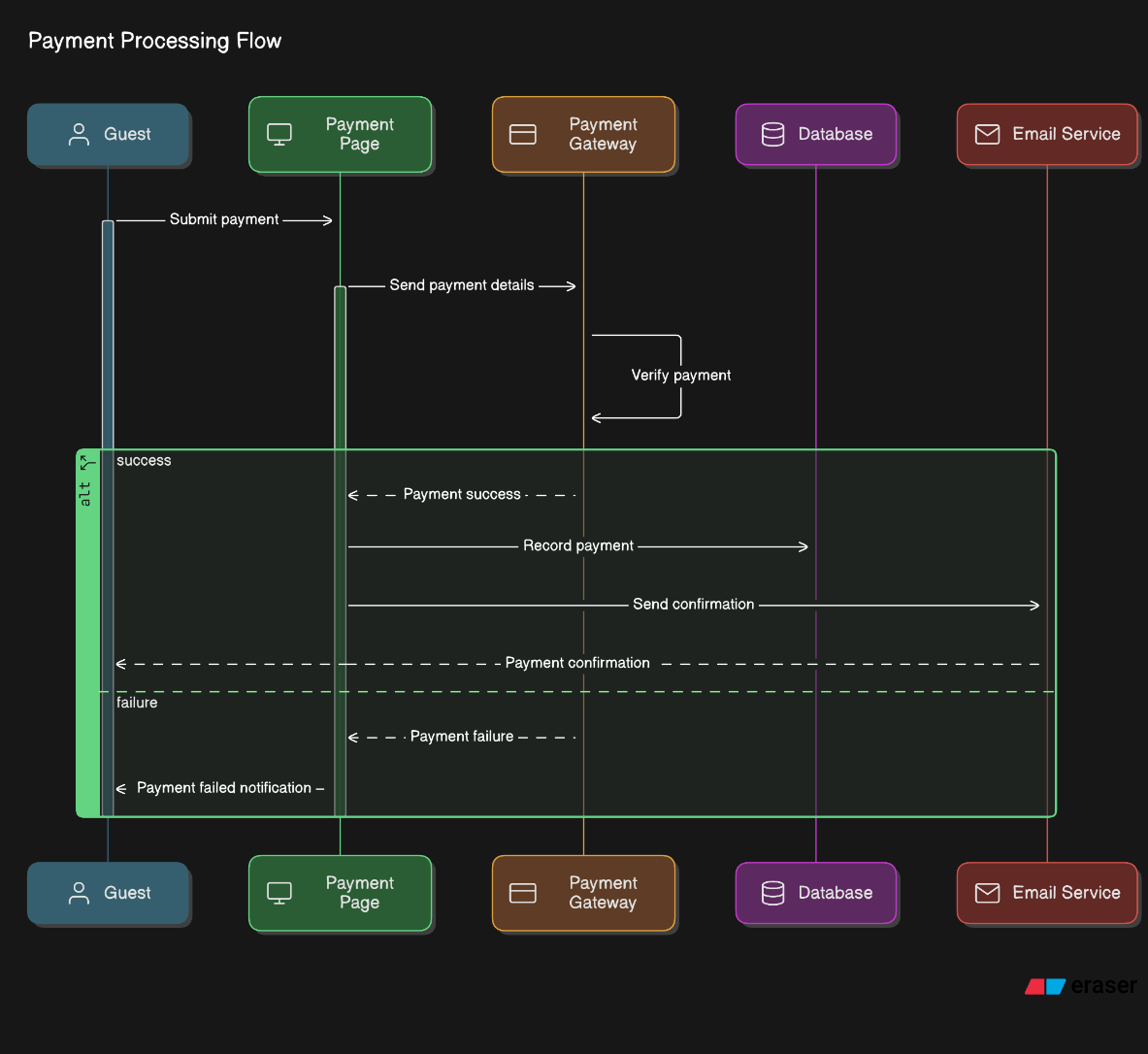
* Normalized Data Model Diagram:



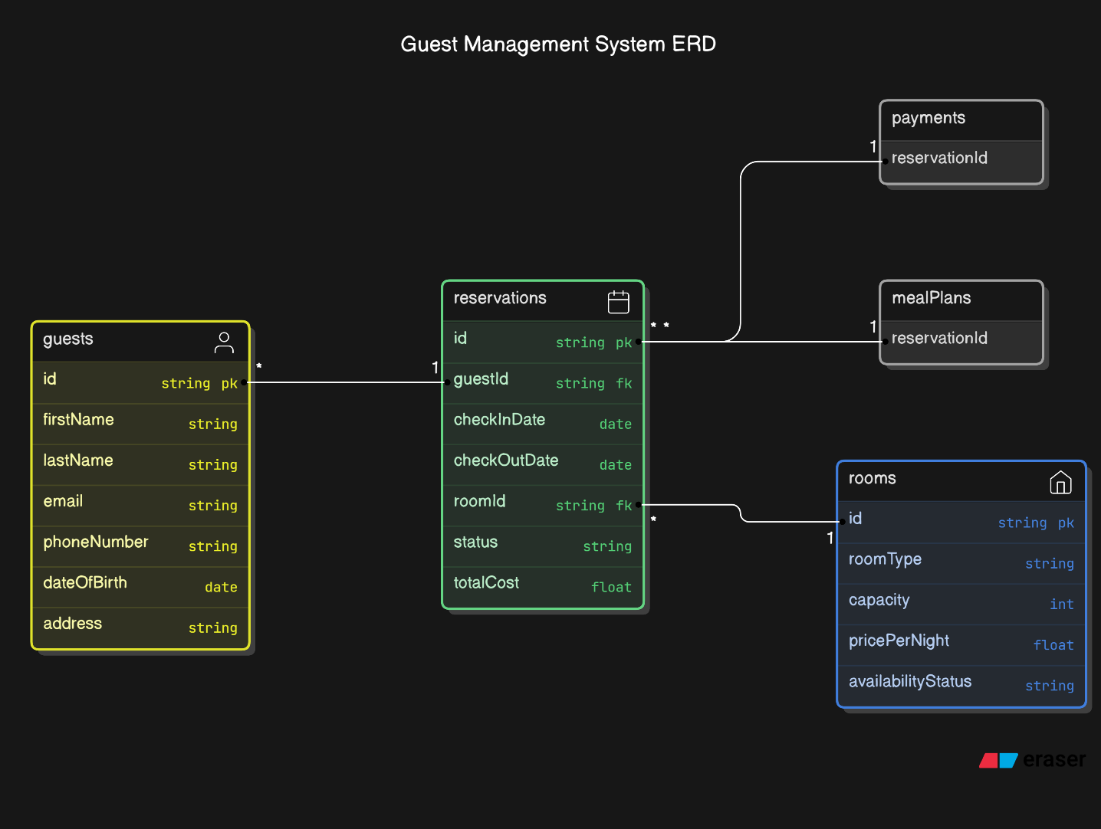
* Activity Diagrams



* Sequence Diagrams

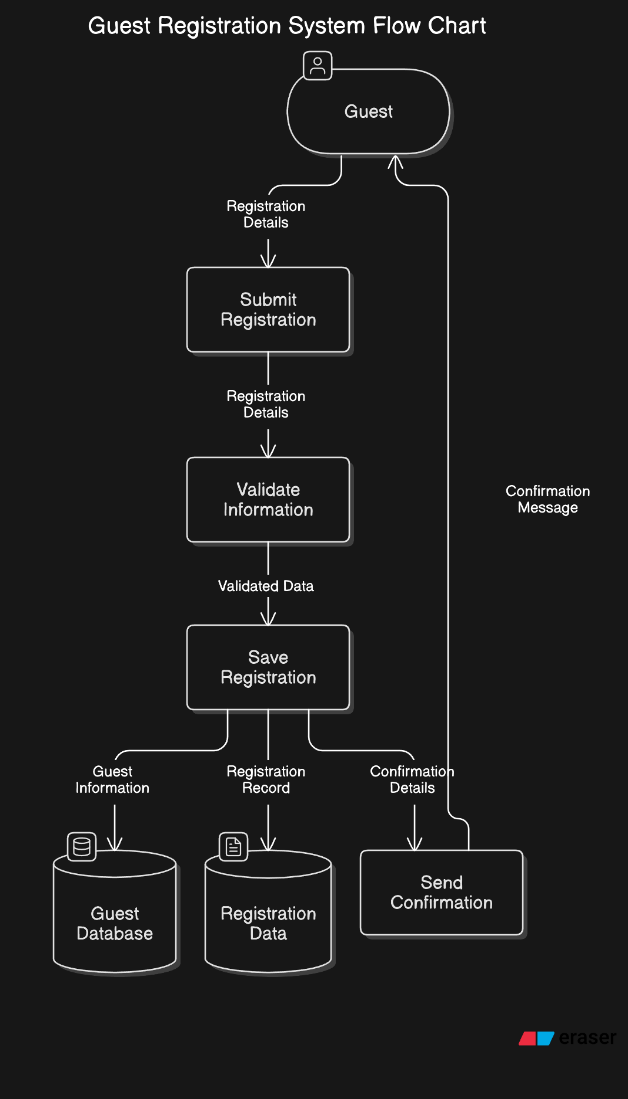


* UML Class Diagram:



**3.4 Process Modelling**

* Data Flow Diagram:



## 4.0 Non-Functional Requirements

The non-functional requirements for the Queen of Apostles Renewal Centre’s system outline constraints that dictate how the system will perform, ensuring it aligns with the center’s specific operational standards and enhances user experience. These requirements are crucial for maintaining overall system quality and meeting user expectations.

#### **4.1 Performance**

* **System Response Time**: The system shall process 95% of registration and booking transactions within 2 seconds to ensure a smooth user experience.
* **Data Processing Efficiency**: Reports and schedules, such as room assignments and meal attendance lists, should be generated within 5 seconds of a user request.
* **Scalability**: The system must support up to 500 simultaneous users without degradation in performance, accounting for potential peak times such as group event registrations.

#### **4.2 Reliability**

* **Error Rate**: The system shall have an error rate of no more than 0.1% for critical operations like registration and payment processing.
* **Data Consistency**: Data updates, such as room assignments or payment status, shall be reflected across the system within 1 second to maintain consistency.
* **Backup and Recovery**: The system shall perform automated backups every 12 hours and must be capable of recovering data within 15 minutes in the event of a system failure.

#### **4.3 Availability**

* **System Uptime**: The system shall maintain 99.9% availability, ensuring minimal disruptions for users registering for programs or accessing schedules.
* **Scheduled Maintenance**: Downtime for system maintenance shall not exceed 1 hour per month and shall be performed during non-peak times (e.g., late night) to reduce user impact.

#### **4.4 Security**

* **Data Protection**: All personal and financial data shall be encrypted using industry-standard encryption (e.g., AES-256).
* **Access Control**: The system shall implement role-based access control (RBAC) to ensure that only authorized users can access specific data or perform certain functions (e.g., approving payments or modifying schedules).
* **Authentication**: The system shall use two-factor authentication (2FA) for administrative users to add an extra layer of security.
* **Compliance**: The system shall comply with relevant data protection regulations such as GDPR to safeguard user data.

#### **4.5 Maintainability**

* **Modular Design**: The system shall be designed using modular architecture to simplify future updates and modifications. Each module (e.g., registration, room management) shall be independently updatable.
* **Documentation**: Comprehensive technical documentation shall be provided to facilitate maintenance tasks, including a user manual and a system architecture guide.
* **Bug Fixes**: Critical bugs shall be addressed and resolved within 48 hours of being reported.

#### **4.6 Portability**

* **Cross-Platform Compatibility**: The system shall be accessible through modern web browsers (e.g., Chrome, Firefox, Edge) and optimized for mobile devices to ensure broad accessibility.
* **Deployment**: The system must be deployable on cloud infrastructure to support remote access by users and staff.
* **Data Transfer**: The system shall allow for easy data migration and integration with existing tools such as Yodeck Dashboard for lobby displays.

These non-functional requirements ensure that the system not only meets functional needs but also performs reliably, securely, and efficiently to enhance the experience for both guests and staff at the Queen of Apostles Renewal Centre.

### 5. Logical Database Requirements

The system for the Queen of Apostles Renewal Centre will require the use of a database to manage various operational data effectively. The following logical database requirements outline how data should be stored, formatted, and managed to meet the needs of the center’s activities and ensure system reliability and efficiency.

#### **5.1 Data Formats**

* **Standardized Structure**: All data shall be stored in standardized formats to facilitate easy data retrieval and integration. For example, dates will follow the ISO 8601 format (YYYY-MM-DD), and currency will be stored in decimal format with two decimal places.
* **Text Fields**: Names, addresses, and other text-based information shall use UTF-8 encoding to support a range of characters and ensure compatibility across various user inputs.

#### **5.2 Storage Capabilities**

* **Scalability**: The database must be capable of handling data growth without performance degradation. Initial storage should support up to 1TB, with the ability to scale based on increased data from registrations and guest records.
* **Data Segmentation**: Data will be segmented into appropriate tables (e.g., Guests, Reservations, Payments, Room Assignments) to optimize query performance and data organization.
* **Backups**: The database shall support automated backups every 12 hours to prevent data loss and ensure business continuity.

#### **5.3 Data Retention**

* **Retention Period**: Guest registration records and financial data shall be retained for a minimum of 7 years, in compliance with regulatory requirements and for audit purposes.
* **Archiving**: Data older than 7 years shall be archived securely, reducing the load on the active database while ensuring accessibility for historical reference.

#### **5.4 Data Integrity**

* **Referential Integrity**: The database shall enforce referential integrity through the use of foreign key constraints to maintain relationships between related tables (e.g., Guests and Reservations, Reservations and Payments).
* **Validation Rules**: Data input validation rules shall be applied to ensure data integrity (e.g., dates cannot be in the past for future bookings, email addresses must follow a proper format).
* **Transaction Management**: All database operations related to registration, payment processing, and room assignment shall be managed using ACID (Atomicity, Consistency, Isolation, Durability) principles to maintain data consistency and prevent errors during concurrent operations.

#### **5.5 Security Measures**

* **Access Control**: The database shall implement role-based access control (RBAC) to ensure only authorized users can perform specific operations (e.g., read, write, update).
* **Encryption**: Sensitive data, such as payment information and guest personal details, shall be encrypted both at rest and in transit using industry-standard encryption protocols.
* **Audit Logs**: The database shall maintain audit logs that track changes to data, including the user who made the change and timestamps, to enhance security and facilitate auditing.

These logical database requirements ensure the system’s database is robust, scalable, secure, and capable of supporting the retreat center’s data needs with high integrity and reliability.

## 6.0 Other Requirements

Additional requirements, if any.

**7.0 Approval**

The signatures below indicate their approval of the contents of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Role | Name | Signature | Date |
| Project Manager | Siam Hasan | Siam | 10/07/2024 |
| Frontend Developer | Bilal Youssef | Bilal | 10/07/2024 |
| Backend Developer | Deigo Tsukayama | Diego | 10/07/2024 |
| Developer | Jason Gunawan | Jason | 10/07/2024 |
| Team Leader | Riyad Youssef | Riyad | 10/07/2024 |