Online Resort Management System

Sharanabasava M, H M Karibasavaswamy, Harish S Suryavanshi, Vikas V M

Sharanabasabasava M, Information Science & Engineering & RYMEC, Ballari
H M Karibasavaswamy, Information Science & Engineering & RYMEC, Ballari
Harish S Suryavashi, Information Science & Engineering & RYMEC, Ballari
Vikas V M, *Information Science & Engineering & RYMEC, Ballari*
Manjula Patil , Assistant professor Dept of ISE RYMEC, ballari

---

Abstract - An Online Resort Management System (ORMS) is a web-based application designed to streamline and automate the operations of a resort. It enhances efficiency by integrating various functionalities such as room reservations, guest management, billing, staff co-ordination, and service requests in to a single digital platform. This system allows guests to book rooms online, select packages, , and request additional services through a user-friendly interface. Resort administrators can manage ensuring smooth operations and better customer service. The system employs features like report generation, feedback management, and analytics assist in decision-making and business growth. to improve the overall experience for both guests and management.

INTRODUCTION

1.1 OBJECTIVES:

Designing an online resort management system for a mini project in a database management system (DBMS) involves various objectives aimed at efficiently managing resort operations and enhancing user experience.

Implement a secure system for users to register, log in, and manage their profiles. This includes storing user information securely and allowing for password recovery and change.

Maintain an updated inventory of available rooms, their types, rates, and amenities. Ensure efficient allocation of rooms based on user preferences and availability.

1.2 LIMITATIONS

When creating a report on the limitations of an online resort management system, it's essential to consider various factors that might hinder its effectiveness and efficiency. Here are some potential limitations:

1. Technical Constraints:

- Dependency on stable internet connectivity: An online system relies heavily on internet access. Poor or unreliable internet connections can disrupt operations.
- Compatibility issues: Compatibility problems may arise with different web browsers or devices, affecting user experience.
- Scalability challenges: As the resort grows or experiences fluctuating demand, the system might struggle to scale efficiently to accommodate increased load or user base.

2. Data Security and Privacy Concerns:

- Vulnerability to cyber threats: Online systems are susceptible tohacking, data breaches, and other cyber attacks, compromising sensitive guest information.
- Compliance with regulations: Ensuring compliance with data protection laws like GDPR or CCPA can be challenging, leading to legal and financial consequences if not properly addressed.

3. User Adoption and Training:

- Resistance to change: Staff members may resist transitioning from traditional manual methods to the online system, leading to slower adoption rates.
- Training requirements: Adequate training is necessary for staff to effectively utilize the system, which can be time-consuming and costly.

4. Dependency on External Service Providers:

- Reliance on third-party vendors: If the resort management system relies on external service providers for hosting, maintenance, or support, disruptions or issues with these providers can affect system functionality.
- Limited control: The resort may have limited control over updates, maintenance schedules, or customization options if reliant on external vendors.

5. Customer Support and Service Interruptions:

- Response time for support: Delays in resolving technical issues or providing customer support can lead to dissatisfaction among users.

- Downtime and service interruptions: Unforeseen technical glitches or maintenance activities can result in system downtime, affecting operations and user experience.

6. Cost Implications:

- Initial investment: Implementing an online resort management system involves significant upfront costs for development, customization, and training.
- Ongoing expenses: Continuous maintenance, updates, and subscription fees for software licenses or cloud services contribute to long-term expenses.

7. Limited Customization and Flexibility:

- Rigidity of features: Off-the-shelf solutions may lack flexibility to accommodate specific resort requirements or unique business processes.
- Cost of customization: Tailoring the system to meet specific needs may incur additional expenses and prolong implementation timelines.

8. Geographical and Cultural Considerations:

- Language and cultural barriers: The system may not fully support multi lingual or culturally diverse users, impacting accessibility and user satisfaction.
- Localization challenges: Adapting the system to comply with local regulations, currencies ,and business practices in different regions can be complex.

SYSTEM DESIGN

The data base-planning phase begins when a customer requests to develop a database project. It is set of tasks or activities, which decide the resources required in the database development and time limits of different activities.

2.1 SOFTWARE REQUIREMENTS PECIFICATION (SRS)

2.1.1 REQUIREMENT COLLECTION AND ANALYSIS

During this step, the database designers interview prospective database users to understand and document their **data requirements**. The result of this step is a concisely written set of users requirements. These requirements should be specified in as detailed and complete a form as possible. In parallel with specifying the data requirements, it is useful to specify the known **functional requirements** of the application. These consist of the user defined **operations** (or **transactions**) that will be applied to the database, including both retrievals and updates.

DATA REQUIREMENTS:

The essential data elements and requirements:

1. User Data:

- Guest information: Name, contact details, address, nationality, ID/passport number.
- Staff information: Name, position, contact details, employee ID, schedule.
- Administrator credentials: Username, password, access levels.

2. Reservation and Booking Data:

- -Reservation details: Reservation ID, check-in/outdates, room type, number of guests, special requests.
- Booking history: Record of past reservations, including dates, room types, and payment status.
- Availability calendar: Tracks room availability by date and room type.
- Payment information: Payment method, transaction ID, amount paid, outstanding balance.

3. Room and Accommodation Data:

- Room inventory: Room number, type (e.g, single, double, suite), capacity, amenities.
- Room status: Clean, dirty, under maintenance, occupied.
- Room rates: Daily rates for different room types, seasonal variations, discounts.
- Housekeeping schedule: Cleaning schedule for each room, last cleaned time stamp.

4. Service and Amenities Data:

- Dining services: Menu items, prices, availability, dietary restrictions.
- Spa services: Treatment options, duration, therapist availability.
- Recreational activities: List of activities available, schedules, equipment rental.

5. Billing and Financial Data:

- Invoice details: Invoice ID, items/services availed, quantities, prices.

- Transaction records: Records of all financial transactions, including payments, refunds, and adjustments.
- Revenue reports: Daily, weekly, monthly revenue summaries, broken down by source (accommodation, dining, spa, etc.).
- Tax information: Tax rates, calculations, compliance records.

6. Feed back and Review Data:

- Guest feedback: Ratings, comments, suggestions submitted by guests.
- Review history: Record of all guest reviews, along with timestamps and responses from management.
- Performance metrics: Aggregate ratings, sentiment analysis, trends overtime.

7. Staff Management Data:

- Staff schedules: Work shifts, assigned tasks, dayso ff.
- Performance evaluations: Performance ratings, feedback from supervisors, training records.
- Staff communications: Internal messaging system, notifications, announcements.

8. Inventory and Supplies Data:

- Inventory items: Consumables (e.g., toiletries, linens), equipment, supplies.
- Stock levels: Current quantities on hand, reorder thresholds, supplier information.
- Purchase orders: Records of orders placed, delivery dates, quantities received.

9. Security and Access Logs:

- User access logs: Records of user logins, access times, IP addresses.
- Security incidents: Reports of unauthorized access attempts, breaches, or suspicious activities.
- Audit trails: Detailed logs of database transactions, including CRUD operations and user IDs.

10 Localization Data:

- Language preferences: Preferred language for communication and interface.
- Currency settings: Default currency for transactions, currency conversion rates.

2.1.2 FUNCTIONAL REQUIREMENTS

Functional requirements outline the specific actions and capabilities that an online resort management system must perform to meet user needs and achieve its objectives. Here are the functional requirements for such a system in a DBMS mini project:

1. User Management:

- User registration: Allow guests to register accounts with necessary information.
- User authentication: Enable login functionality with username and password.
- User roles: Differentiate between guests, staff, and administrators with appropriate permissions.

2. Reservation Management:

- Room reservation: Allow guests to search for available rooms based on dates, room types, and preferences.
- Booking confirmation: Provide instant confirmation of reservations and send email notifications to guests.
- Reservation modification and cancellation: Allow guests to modify booking details or cancel reservations within specified policies.

3. Room Management:

- Room availability: Display real-time availability of rooms based on date range and occupancy.
- Room details: Provide detailed information about each room type, including amenities, rates, and images.
- Room allocation: Automatically assign rooms to guests upon reservation or allow manual allocation by staff.

4. Billing and Payment:

- Generate invoices: Automatically generate invoices for reservations, additional services, and amenities.
- Payment processing: Integrate with payment gateways to facilitate secure online payments via credit/debit cards or other methods.
- Payment tracking: Track payment status, including pending, paid, or refunded amounts.

5. Localization Data:

- Language preferences: Preferred language for communication and interface.
- Currency settings: Default currency for

transactions, currency conversion rates.

SYSTEM REQUIREMENTS

Functional requirements outline the specific actions and capabilities that an online resort management system must perform to meet user needs and achieve its objectives. Here are the functional requirements for such a system in a DBMS mini project:

1.User Management:

- User registration: Allow guests to register accounts with necessary information.
- User authentication: Enable login functionality with username and password.
- User roles: Differentiate between guests, staff, and administrators with appropriate permissions.

2.Reservation Management:

- Room reservation: Allow guests to search for available rooms based on dates, room types, and preferences.
- Booking confirmation: Provide instant confirmation of reservations and send email notifications to guests.
- Reservation modification and cancellation: Allow guests to modify booking details or cancel reservations within specified policies.

3.Room Management:

- Room availability: Display real-time availability of rooms based on date range and occupancy.
- Room details: Provide detailed information about each room type, including amenities, rates, and images.
- Room allocation: Automatically assign rooms to guests upon reservation or allow manual allocation by staff.

5. Billing and Payment:

- Generate invoices: Automatically generate invoices for reservations, additional services, and amenities.
- Payment processing: Integrate with payment gateways to facilitate secure online payments via credit/debit cards or other methods.
- Payment tracking: Track payment status, including pending, paid, or refunded amounts.
- Data encryption: Encrypt sensitive information such as user credentials, payment details, and personal data.
- Access control: Implement role-based access control (RBAC) to restrict access to sensitive functionalities and data.
- Compliance auditing: Maintain audit logs and conduct regular audits to ensure compliance with data protection regulations.

6. System Administration:

- Configuration settings: Allow administrators to configure system settings, such as room rates, tax rates, and email templates.
- Backup and recovery: Implement automated backups and recovery procedures to prevent data loss and ensure system reliability.
- System maintenance: Schedule regular maintenance tasks, updates, and patches to keep the system running smoothly.
- These functional requirements form the foundation of an online resort management system, enabling efficient operation and providing a seamless experience for guests, staff, and administrators.

2.1.3 SOFTWARE AND HARD WARE REQUIREMENTS

1. Intel core i3 2nd generation is used as processor because it is faster &provide reliable and stable working environment.
2. A RAM size of 1GB is used as it will provide fast reading &writing capabilities.
3. For the implementation of an online resort management system in a DBMS mini-project, various software tools and technologies can be utilized. Here's a **list of commonly used software:**

1. **Data base Management System (DBMS):**

- MySQL
- PostgreSQL
- Microsoft SQL Server
- Oracle Database

2. **Programming Languages:**

- Java
- Python
- PHP
- JavaScript (Node.js)

- C#

3. **Web Development Frame works:**

- Django (Python)
- Flask (Python)
- Spring Boot (Java)
- Laravel(PHP)
- Express.js (Node.js)

4. **Frontend Development:**

- HTML5
- CSS3
- JavaScript
- Bootstrap
- React.js
- Vue.js
- Angular

5. **Version Control:**

- Git
- SVN(Subversion)

6. **Integrated Development Environment (IDE):**

- Eclipse
- Intelli JIDEA
- PyCharm
- Visual Studio Code
- Net Beans
- Sublime Text

7. **Web Servers:**

- Apache HTTP Server
- Nginx
- MicrosoftInternetInformationServices(IIS)

8. **Operating Systems:**

- Linux (Ubuntu,CentOS,Debian,etc.)
- Windows Server
- macOS (for development)

9. **Payment Gateway Integration:**

- PayPal API
- Stripe API
- Square API
- Authorize.Net API

10. **Email Services:**

- SMTP (Simple Mail Transfer Protocol) servers
- SendGrid
- Mailgun
- Amazon SES (Simple Email Service)

11. **Data Visualization and Reporting:**

- Tableau
- Power BI
- Google Data Studio

12. **Project Management and Collaboration:**

- Jira
- Trello
- Asana
- Slack
- Microsoft Teams

13. **Documentation:**

- Microsoft Office (Word ,Excel, PowerPoint)
- Google Docs, Sheets, Slides
- Mark down editors

14. **Testing and Quality Assurance:**

- JUnit (for Java)
- Pytest (for Python)
- Selenium (for automated testing)
- Postman (for API testing)

15. **Security Tools:**

- SSL certificates
- OWASP ZAP (Zed Attack Proxy)
- Nessus
- Metasploit

16. **Backup and Recovery:**

- Database backup utilities (e.g.,mysql dump for MySQL)
- Cloud storage solutions (e.g., AWSS3,GoogleCloud Storage)

17. **Deployment and Hosting:**

- Amazon Web Services(AWS)
- Microsoft Azure
- Google Cloud Platform(GCP)
- Heroku
- Digital Ocean

2.2 CONCEPTUAL DESIGN

Once all the requirements have been collected and analyzed, the next step is to create a conceptual schema for the database, using a high-level conceptual data model. This step is called the conceptual design..The conceptual schema is a concise description of the data requirements of the users and includes detailed descriptions of the entity types, relationships, and constraints; these are expressed using the concepts provided by the high-level data model. Because these concepts do not include implementation details, they are usually easier to understand and can be used to communicate with nontechnical users. The high-level conceptual schema can also be used as a reference to ensure that all users' data requirements are met and that the requirements do not conflict. This approach enables database designers to concentrate on specifying the properties of the data, without being concerned with storage and implementation details, which makes it is easier to create a good conceptual database design. The result of this step is an Entity

2.3 LOGICAL DESIGN

The next step in database design is the actual implementation of the database, using a commercial DBMS. Most current commercial DBMSs use an implementation data model— such as the relational (SQL) model—so the conceptual schema is transformed from the high- level data model in to the implementation data model. This step is called **logical design** or **data model mapping**; its result is a database schema in the implementation data model of the DBMS. The ER to relation mapping algorithm is used to convert conceptual design into logical design. In this phase, the primary keys and foreign keys are defined.

Mapping an Entity-Relationship (ER) model to relations in a database involves several steps. For an online resort management system, the process generally follows these steps:

1. **Identify Entities:** Identify all the entities involved in the system. In an online resort management system, entities might include Customer , Booking , Price , Priviege, etc.
2. **Define Attributes:** For each entity, determine the attributes or properties that describe them. For example, a Customer entity might have attributes like PID, FName, LName, Email, Tel, etc.
3. **Establish Relationships:** Identify the relationships between entities. For instance, a Customer places Booking, so there's a one-to-many relationship between Customer and Booking.
4. **Determine Cardinality:** Determine the cardinality of each relationship, indicating how many instances of one entity are related to another. For example, one Customer can place many Booking, but each Booking is placed by only one Customer.
5. **Normalize Entities:** Ensure that each entity is in its most normalized form to avoid redundancy and ensure data integrity.
6. **Convert ER Diagram to Relations:** Once the ER model is finalized, convert it into relational schema by creating tables for each entity and including attributes as columns.
7. **Define Primary Keys:** Choose a primary key for each table, ensuring that it uniquely identifies each row in the table. This is typically based on one or more attributes that uniquely identify each entity instance.
8. **Establish Foreign Keys:** If there are relationships between tables, include foreign keys in the related tables to enforce referential integrity.
9. **Refine the Schema:** Review the relational schema for any potential improvements or optimizations, such as adding indexes for performance.
10. **Normalize Relations (Optional):** If necessary, further normalize relations to eliminate any anomalies or redundancies.
11. **Validate the Schema:** Ensure that the rlational schema accurately represents the requirements of the online resort management system and meets the desired functionality.

2.4 IMPLEMENTATION

The various system tools that have been used in developing both the presentation layer,

Middle layer and data base layer of the project are being discussed in this chapter.

2.4.1 PRESENTAION LAYER (FRONTEND):

HTML, CSS, JAVASCRIPTS are utilized to implement the front end.

HTML (Hyper Text Markup Language): HTML is a syntax used to format a text document on the web. Hyper text Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and Java Script it forms a triad of Cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a

web server or from local storage and render them into multimedia web pages. HTML describes the structure of a webpage semantically and originally included cues for the appearance of the document. **CSS (Cascading Style Sheets):** CSS is a style sheet language used for describing the look and formatting of a document written in a markup language. Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. CSS is used to set the visual style of web pages and user interfaces written in HTML and XHTML. Along with HTML and Java Script, CSS is a cornerstone technology used by most web sites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications

Java Script: JS is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

2.4.2 MIDDLE LAYER:

Middle layer is implemented using PHP. In PHP development, the term "middle layer" typically refers to a middleware component or layer that sits between the frontend (user interface) and the backend (database, server-side logic). This layer often handles tasks such as authentication, authorization, input validation, and data transformation. In PHP, frame works like Laravel, Symfony, and Slim provide middleware functionality to facilitate these tasks effectively. You can create custom middleware or use built-in middleware provided by these frameworks to enhance your application's security, performance, and maintainability.

2.4.3 DATABASE SERVICES LAYER (BACKEND)

The back end is implemented using Oracle which is used to design the databases. Oracle is an open-source relational database management system (RDBMS). The SQL phrase stands for structured query.

An Oracle databaseis a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in multi user environments that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery**.**

CONCLUSION

In conclusion, the development of the online resort management system as part of this project has been a valuable learning experience. Our primary objective was to create a functional system that could efficient lymanage various aspects of resort operations using a robust database management system.

Throughout the project, we successfully designed and implemented key features such as reservation management, room allocation, guest services, and billing, all integrated within a centralized database architecture. This allowed for streamlined operations and improved data organization, accessibility, and security.

In conclusion, this mini-project has not only allowed us to apply theoretical knowledge ina practical setting but has also equipped us with valuable skills and experience that will be beneficial in our future endeavors in the field of software development and database management.

This conclusion succinctly summarizes the objectives, outcomes, areas of improvement, and the significance of the mini-project on an online resort management system built on a DBMS.

REFERENCES

[1] Elmasri and Navathe,“Fundamentals of Database Systems”, 3/e,Addison- Wesley,2001

[2] ASilberschaltz, H.F. Korth, and Ssudarshan,“Database System Concepts”,3/e,TataMcgraw Hill,1997

[3] Thomas M. Connolly, Carolyn E. Begg, “Database Systems & Practical Approach to Design Implementation and Management”, 4/e,Addison – Wesley, 2005

[4] Wikipedia for various information

BIOGRAPHIES

Mrs. Manjula Patil
Assistant Professor, Dept of ISE, RYMEC,Ballari.

SHARANABASAVA M
Final Year student of Information Science and Engineering (ISE) Department, RYMEC, Ballari.

H M KARIBASAVASWAMY
Final Year student of Information Science and Engineering (ISE) Department, RYMEC, Ballari.

HARISH S SURYAVANSHI
Final Year student of Information Science and Engineering (ISE) Department, RYMEC, Ballari.

VIKAS V M
Final Year student of Information Science and Engineering (ISE) Department, RYMEC, Ballari.