A PROPOSED OFFERING OF A HOTEL RESERVATION MANAGEMENT SYSTEM FOR EUROTEL NORTH EDSA

A Technical Documentation Presented to the
Faculty of Datamex College of Saint Adeline, Inc.

In Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Information Technology

Presented by:
Adarayan, Jessie
Anacta, Francis Rico
Bernadas, Marjorie
Pedro, Pamela Anne

INTRODUCTION

This technical documentation provides a comprehensive overview of the development, design, and implementation of the Hotel Reservation Management System (HRMS) for Eurotel North Edsa. It serves as an essential guide for developers, administrators, and other stakeholders who will use, maintain, or enhance the system in the future. The primary purpose of this document is to record every technical aspect of the project, ensuring that the system can be properly understood, deployed, and maintained beyond its initial development. It also acts as an official reference that outlines how each component of the system operates, how they interact with one another, and how they contribute to the overall functionality of the HRMS.

The Hotel Reservation Management System is designed to automate and simplify the reservation process for Eurotel North Edsa. Traditionally, the hotel relied on manual booking methods that often led to inefficiencies, such as booking conflicts, misplaced guest information, and delayed check-ins or check-outs. The proposed system addresses these issues by providing a web-based platform that integrates all reservation-related processes into a single, accessible environment. Guests can easily book rooms online or through walk-in reservations, while hotel administrators can manage bookings, verify payments (in hotel), and monitor room availability. The system not only enhances convenience for guests but also improves operational accuracy and reduces administrative workload. Furthermore, it provides detailed reports that help the management to improve hotel operations and customer satisfaction.

The scope of this technical documentation encompasses all aspects necessary for the proper operation and maintenance of the HRMS. It covers the system architecture, installation and configuration guidelines, database structure, API endpoints, and security protocols used in the development process. Additionally, it includes user manuals, troubleshooting guides, performance testing results, and procedures for future updates or enhancements. The document also defines best practices for configuration, outlines common issues and their solutions, and explains maintenance routines to ensure long-term system reliability. By consolidating all these technical details, this document serves as a comprehensive resource that

promotes consistency, efficiency, and clarity throughout the lifecycle of the Hotel Reservation Management System.

SYSTEM OVERVIEW

The Hotel Reservation Management System (HRMS) is developed using a three-tier architecture composed of the client side, application layer, and database layer. This architectural structure ensures that system processes are well-organized, secure, and efficient, allowing smooth communication between the different components. The client side represents the user interface, where guests and administrators interact with the system through a web browser. The application layer handles the logic and processes behind every operation, such as managing reservations, processing payments, validating user input, and updating records. The database layer, on the other hand, stores all essential information including guest details, room data, reservation records, and transaction history, ensuring that all information is maintained securely and can be retrieved accurately when needed.

The main parts of the system are made to work together to give a smooth and easy experience for both guests and hotel staff. The Guest Module allows users to create accounts, log in, view available rooms, and make reservations online. The Admin Module allows hotel staff to access the system to manage bookings, update room availability, confirm payments, and generate operational reports, even without individual user accounts. The Database Module acts as the backbone of the system, storing and organizing all relevant data while ensuring that transactions are processed consistently and efficiently. These modules interact continuously, with data flowing from the client interface to the application layer and then to the database, allowing updates to room status and reservation details.

The HRMS uses a client–server deployment setup. Users can access the system through common web browsers on desktops and laptops. The web application is hosted on an Apache server, which handles user requests using PHP scripts. These scripts connect the frontend (what users see) with the MySQL database, making sure data is safely sent and stored. Communication between the client and server happens over HTTP or HTTPS using JSON-formatted data for better efficiency and security. In small-scale installations, the web server and database can run on the same machine. However, for larger operations, the system can be scaled by using separate servers to improve speed and stability.

INSTALLATION GUIDE

This section provides the necessary information for setting up and installing the **Hotel Reservation Management System (HRMS)**. It outlines the hardware and software requirements, the installation process, and configuration settings to ensure the system runs efficiently within its environment.

Category	Requirement	Description
Hardware	Processor	Intel Core i5 (10th Generation or higher)
	Memory	8 GB RAM minimum
	Storage	128 to 256 GB SSD or HDD
	Network	Stable internet or LAN connection
	Peripherals	Standard monitor, keyboard, and mouse
Software	Operating System	Windows
	Web Server	Apache (XAMPP)
	Database Server	MySQL
	IDE	Visual Studio Code or any PHP/HTML editor
	Web Browser	Google Chrome, Mozilla Firefox, or Microsoft Edge
	PHP Version	PHP 8.0
Dependencies	Data Exchange	JSON support for API communication

Table 1: Hardware, Software, and Dependency Requirements for the Hotel Reservation Management System

STEP-BY-STEP INSTALLATION PROCEDURE

1. **Install XAMPP:**

   Download and install the XAMPP package to set up Apache and MySQL servers.

2. **Create the Database:**
   - Launch **phpMyAdmin** from the XAMPP control panel.
   - Create a new database named **hrms_db**.
   - Import the provided SQL file to create all necessary tables.
3. **Copy Project Files:**
   - Move the HRMS project folder to the htdocs directory of your XAMPP installation.
   - Verify that all system folders such as config, includes, and assets are intact.
4. **Configure Database Connection:**
   - Open the **config.php** file inside the HRMS directory.
   - Update the following values according to your local setup:
     - $host = "localhost"
     - $user = "root"
     - $password = ""
     - $database = "hrms_db"
   - Save the file after editing.
5. **Install and Set Up Visual Studio Code (VS Code):**
   - Download and install Visual Studio Code from the official website.
   - Recommended extensions:
     - PHP Intelephense (for PHP language support and IntelliSense).
     - PHP Debug (for debugging with Xdebug).
     - Open VS Code and select **File → Open Folder** then choose HRMS project folder.
     - Use the Integrated Terminal in VS Code (Ctrl+`) to run command-line operations such as database imports.
6. **Run the System (If XAMPP):**
   - Start Apache and MySQL using the XAMPP control panel.
   - Open your web browser and visit: http://localhost:3000/
   - The HRMS login page should appear, indicating a successful setup

Configuration Settings and Options

After installation, the administrator may configure several system settings to match the hotel's operations. The config.php file contains key parameters such as database credentials, base URL, and general site information like the hotel name, address, and contact details. These settings ensure that the system connects properly to the local MySQL database and displays accurate organizational information within the interface.

In addition, PHP settings can be modified in the php.ini file to improve system performance, such as increasing file upload limits, adjusting maximum execution time, and enabling error reporting during development. Properly configuring these options ensures that the Hotel Reservation Management System operates smoothly, maintains stable performance, and provides a secure environment for both administrators and users.

CONFIGURATION GUIDE

This section provides detailed instructions on how to configure the Hotel Reservation Management System (HRMS) after installation. Proper configuration ensures that the system functions smoothly within the local environment and that each component, such as the database, base URL, and user access, operates as intended. The configuration process mainly involves editing the config.php file and adjusting a few PHP settings to optimize performance and ensure compatibility with the local server setup.

Detailed Configuration Instructions

- Open the config.php file located in the HRMS project folder.
- Find the database connection settings and modify them according to local MySQL setup:
    - $host = "localhost" // Database host
    - $user = "root" // MySQL username
    - $password ="" // MySQL password
    - $database = "hrms_db" // Database name
- Save the file, then restart **Apache** and **MySQL** from the XAMPP control panel.
- Open your web browser and go to http://localhost/HRMS to verify that the system loads without a database connection error.

Base URL Configuration

- The base URL determines how the system is accessed through the web browser.
- Inside config.php, verify or update this value if your project folder name changes or if you deploy it online:
    - $baseURL = http://localhost:3000/

User Account Configuration

- Administrative and staff accounts are stored in the database under the user's table.
- To edit or add new accounts, open phpMyAdmin from XAMPP, select the hrms_db database, and navigate to the user's table.
- User roles (Admin, Staff) can be modified manually in this table, giving specific permission depending on responsibilities.

System Settings

- The HRMS includes a section in the Admin Panel where basic information such as the hotel name, address, and contact number can be updated.
- These values may also be stored in a configuration file named settings.php or directly within the database.
- Updating this information ensures that all reports, receipts, and user interfaces display the correct details.

PHP Environment Settings

The default PHP environment in XAMPP is already suitable for running the **Hotel Reservation Management System (HRMS)**. No additional configuration changes are required to ensure system performance and stability. However, administrators should verify that the PHP and MySQL modules are enabled in XAMPP before starting the Apache server. This ensures that all PHP scripts and database operations function properly. If the system is transferred to another server or computer, make sure that the PHP version remains compatible with the project to avoid runtime issues. Once confirmed, the HRMS will operate smoothly under standard XAMPP settings without further adjustments.

Configuration File Formats and Parameters

File Name	Parameter	Description
config.php	$host, $user, $password, $database	MySQL connection details
config.php	$baseURL	Defines the project's access path in the browser
settings.php	$hotelName, $contactNumber, $address	Stores general hotel information

Table 2: Configuration Files, Parameters, and Their Descriptions

Best Practices for Customization

- Back Up Before Changes: Always back up configuration files before making any edits.
- Use Secure Credentials: Replace default MySQL usernames and passwords with unique credentials.
- Restrict Access: Configuration files should only be editable by administrators or authorized developers.
- Organize Config Files: Keep all configuration files inside a /config folder for easy maintenance.
- Document Modifications: Maintain a short changelog every time a configuration is updated.
- Restart Servers After Edits: Restart Apache and MySQL after making major changes to apply the new settings.

API DOCUMENTATION

This section describes the internal API structure of the Hotel Reservation Management System (HRMS). Although the system operates as a traditional web application, several PHP scripts act as functional endpoints that handle requests between the client interface and the database. These scripts perform essential operations such as login authentication, room retrieval, reservation management, and data updates.

All communication between the client and server is done through HTTP requests (using either the GET or POST method), and responses are returned in either HTML or JSON format depending on the operation.

Endpoint / File	**Description**	**Request Method**	**Access Level**
login.php	Validates user credentials and starts a session for Admin or Staff users.	POST	Admin
logout.php	Ends the current user session and redirects to the login page.	GET	Admin
register.php	Handles new account creation	POST	Admin
fetchRooms.php	Retrieves a list of available rooms from the database.	GET	Public / Guest
bookRoom.php	Saves a new booking request including customer details, room ID, and check-in/out dates.	POST	Guest

getReservationDetails.php	Displays or returns the details of a specific reservation.	GET	Admin
updateReservationStatus.php	Updates the status of a booking (confirmed, checked-in, checked-out, cancelled).	POST	Admin
deleteReservation.php	Removes a reservation record from the system.	POST	Admin
rooms_add.php	Adds new room information to the database.	POST	Admin
rooms_update.php	Updates existing room details.	POST	Admin
rooms_delete.php	Deletes a room entry from the system.	POST	Admin
viewReports.php	Retrieves reservation and payment reports for viewing or export.	GET	Admin

Table 3: Summary of API Endpoints and Their Functionalities

Endpoint URLs, Request Formats, and Responses

All endpoints are accessed locally through the following base URL: http://localhost:3000/

1. **Login Authentication**

   Endpoint: http://localhost:3000/login.php
   Method: POST
   Parameters

Parameter	Type	Description
username	String	Guest or Admin
password	String	Account password

Sample Response (on success)

```
{
  "status": "success",
  "role": "admin",
  "message": "Login successful"
}
```

Sample Response (on failure)

```
{
  "status": "error",
  "message": "Invalid username or password"
}
```

2. **RoomAvailability Retrieval**

   Endpoint: http://localhost:3000/fetchRooms.php

   Method: GET

   Sample Response

```
[
  {
    "room_id": 101,
    "room_type": "Deluxe",
    "rate": 2000,
    "availability": "Available"
  },
  {
    "room_id": 102,
    "room_type": "Suite",
    "rate": 3500,
    "availability": "Occupied"
  }
]
```

3. **Booking a Room**

   Endpoint: http://localhost:3000/bookRoom.php

Method: POST

Parameters

Parameter	Type	Description
customer_name	String	Guest's full name
contact_number	String	Guest's contact number
room_id	Integer	ID of the room being booked
check_in	Date	Date of arrival
check_out	Date	Date of departure

Sample Response:

```
{
  "status": "success",
  "message": "Room successfully booked",
  "reservation_id": 1
}
```

4. **Updating Reservation Status**

Endpoint: http://localhost:3000/updateReservationStatus.php

Method: POST

Parameters

Parameter	Type	Description
reservation_id	Integer	ID of the reservation
status	String	New reservation status (Confirmed, Checked-in, Cancelled)

Sample Response

```
{
  "status": "success",
  "message": "Reservation status updated successfully"
}
```

Authentication and Authorization

- **Session-Based Authentication**
  The system uses PHP sessions to verify user identity after login. Each authorized user (Admin) must log in through the system interface before accessing restricted endpoints.
- **Role-Based Access**
  User permissions are defined within the database under the user's table. Admins can access and modify all modules, while staff accounts have limited access (managing reservations but not deleting records).
- **Guest Access**
  Guests may interact only with public endpoints such as room availability and booking submission. They do not require login credentials.

DATABASE DOCUMENTATION

The Hotel Reservation Management System (HRMS) utilizes a relational database designed in MySQL to store, organize, and manage all hotel-related information, including guest records, room details, and reservation transactions. The database ensures data consistency, reduces redundancy, and maintains secure access to sensitive information.

Entity–Relationship Diagram (ERD)

The following ERD illustrates the logical structure of the HRMS database. It shows the entities, their attributes, and relationships that define how data flows within the system.

Main Entities

1. **Users** – Stores login credentials and role-based access (Admin or Staff).
2. **Rooms** – Contains information about available rooms, including their type, rate, and availability status.
3. **Reservations** – Manages guest booking details, including room, check-in/check-out dates, and reservation status.
4. **Payments** – Records transaction details for each reservation.
5. **Customers** – Contains personal information of guests making reservations

Description of Database Tables

Table Name	Description
Users	Stores system user accounts, including username, password (hashed), and user role (Admin).
Rooms	Contains room details such as room number, type, rate, and availability.

Reservations	Holds booking information, including customer details, assigned room, check-in/check-out dates, and reservation status.
Payments	Records payment transactions for completed reservations, including payment amount, method, and date.
Customers	Stores guest information such as name, contact number, and email address.

Table Structures and Fields

1. Users

Field Name	Type	Description
user_id	INT (PK)	Unique identifier for each user
username	VARCHAR (50)	Username used for login
password	VARCHAR (255)	Hashed password for security
role	ENUM ('Admin')	Defines user access level

2. Rooms

Field Name	Type	Description
room_id	INT (PK)	Unique identifier for each room
room_number	VARCHAR (20)	Room number or code
room_type	VARCHAR (50)	Category of room (e.g., Deluxe, Suite)

rate	DECIMAL (10,2)	Price per night
availability	ENUM ('Available', 'Occupied', 'Maintenance')	Room status

3. Customers

Field Name	**Type**	**Description**
customer_id	INT (PK)	Unique identifier for each guest
name	VARCHAR (100)	Guest's full name
email	VARCHAR (100)	Guest's email address
contact_number	VARCHAR (20)	Guest's contact number

4. Reservations

Field Name	**Type**	**Description**
reservation_id	INT (PK)	Unique reservation identifier
customer_id	INT (FK)	References to the customer who made the reservation
room_id	INT (FK)	References for the booked room
check_in	DATE	Check-in date
check_out	DATE	Check-out date
status	ENUM ('Pending', 'Confirmed', 'Checked-in', 'Checked-out', 'Cancelled')	Booking status

date_reserved	DATETIME	Timestamp when the reservation was created

5. Payment

Field Name	**Type**	**Description**
payment_id	INT (PK)	Unique payment identifier
reservation_id	INT (FK)	References for the corresponding reservation
amount	DECIMAL (10,2)	Amount paid
method	VARCHAR (30)	Payment method (Cash, Card, Online)
payment_date	DATETIME	Date and time of payment

Data Migration and Backup Procedures

1. **Data Migration**
    - Export existing data from previous hotel management records.
    - Import the data into the HRMS MySQL database using phpMyAdmin's Import feature or MySQL Workbench.
    - Validate imported data by checking relationships between tables (especially room and reservation references).
    - Test all modules (booking, reports) to ensure that migrated data functions correctly within the system.
2. **Database Backup**
    - Regularly back up the database using phpMyAdmin → Export → SQL format.

- Save the .sql backup files in a secure, dated folder
- For additional security, store copies on external drives or cloud storage.
- To restore a backup, import the .sql file back into the MySQL database using phpMyAdmin’s Import feature.

USER MANUAL

Instructions for Using the Software

The **Hotel Reservation Management System (HRMS)** is a web-based platform developed to simplify and automate the reservation process for **Eurotel North Edsa**. It eliminates the need for manual booking methods by allowing administrators to manage hotel operations and guests to reserve rooms conveniently online.

The system can be accessed through a web browser using the following local URL:**http://localhost:3000/**

There are two main types of users

1. **Administrator** – Has full access to all features, including managing rooms, customers, reservations, and payments.
2. **Guest/User** – Can browse rooms, check availability, and make online reservations.

To use the system

1. Open your preferred browser (Google Chrome/Microsoft Edge is recommended).
2. Enter the system address (http://localhost:3000/) in the browser's address bar.
3. Log in using your assigned account (for Admin) or create a new booking (for Guests).
4. After login or booking, navigate through the available pages.
5. Always log out after use to maintain security.

User Interface Descriptions and Navigation Guidelines

The HRMS interface is intuitive and designed for easy navigation. The main components of the user interface include:

1. **Home Page**
    - Displays the hotel's name, a welcoming banner, and an introduction to Eurotel North Edsa.
    - Features quick-access buttons such as **Reserve Now**, **View Rooms**, and **Promos & Deals**.
2. **Navigation Bar**
    - Located at the top of each page for consistent access.
    - Contains links to **Home**, **Rooms**, **Promos & Deals**, **Contact Us**, and **Guest Login**.
    - The **Guest Login** button allows guests to log in or manage their reservations.
3. **Admin Dashboard**
    - Accessible only to the Administrator after logging in.
    - Displays an overview of total rooms, active reservations, and room availability.
    - Contains management modules for:
        - **Rooms Management** – Add, edit, or remove rooms and set their availability.
        - **Customer Records** – View and manage guest information.
        - **Reservations** – View, confirm, cancel, or update reservations.

4. **Reservation Page**
    - Allows guests to make new reservations by selecting a room, choosing check-in and check-out dates, and submitting personal details.
    - Displays room availability and confirmation details after submission.
5. **Contact Page**
    - Shows the hotel's contact information, including address, phone number, and email.
    - Guests can send inquiries or requests directly from this page.

Common Tasks and Workflows

Task	**Description**	**User Role**
Make a Reservation	Click "**Reserve Now**," fill in guest details, select room type and stay duration, then submit.	Guest
Confirm Reservation	Review pending bookings in the **Reservations** panel and change status to "Confirmed."	Admin
Cancel Reservation	Select an existing booking in the **Reservations** panel and set status to "Cancelled."	Admin
Add or Edit Room Details	Go to **Rooms Management**, update room information such as rate, type, or availability, then save.	Admin
View Guest Information	Access the **Customer Records** section to see registered guest details.	Admin

Log Out	Click the account dropdown in the upper-right corner and select "**Log Out**."	Admin

Table 4: Common Tasks and User Workflows in the HRMS

TROUBLESHOOTING GUIDE

Common Issues and Error Messages

This section provides solutions for common technical issues that may arise while using the Hotel Reservation Management System (HRMS). The goal is to help users quickly identify and resolve problems without the need for extensive technical assistance.

Issue / Error Message	Possible Cause	Solution / Resolution
Database Connection Failed	Incorrect database credentials in the config.php file or the MySQL server are not running.	Verify that MySQL is installed and running. Check the config.php file to ensure the database name, username, and password are correct.
Page Not Found (404 Error)	The requested page or file does not exist in the system directory.	Make sure all system files are properly uploaded to the correct folder (/htdocs/hrms/). Double-check internal links and file names.
Access Denied	The user attempted to access admin pages without proper authentication.	Log in using the correct administrator credentials. If the problem persists, clear your browser cache and retry.
Unable to Submit Reservation	Missing required input fields or JavaScript validation error.	Ensure all required fields are filled in (check-in, check-out, name, contact details). Reload the page and try again.
Blank Page After Login	PHP error due to missing extensions or syntax issue in code.	Enable PHP error reporting (error_reporting(E_ALL)

		;) to identify the issue, or reinstall the required PHP extensions (mysqli, pdo).
Images Not Displaying	The image path is incorrect, or files were not uploaded properly.	Check the uploads/ or images/ folder path and re-upload missing images.
Cannot Connect to Server	Apache or MySQL service is stopped.	Open XAMPP Control Panel and start **Apache** and **MySQL**. Then reload the page.

Table 5: Troubleshooting Guide for Common Issues and Errors

Troubleshooting Steps and Resolutions

If an issue occurs while using the HRMS, follow these basic troubleshooting steps before seeking technical support.

1. **Check Server Status**
    - Ensure that Apache and MySQL are running in the XAMPP Control Panel.
    - Restart the services if necessary.
2. **Verify Configuration Files**
    - Open the config.php file and confirm that all parameters (host, username, password, and database name) are correct.
    - Make sure file paths (for images or includes) are accurate.
3. **Clear Browser Cache**
    - Sometimes cached versions of files cause outdated results. Clear cache and cookies, then reload the system**.**
4. **Check Database Tables**
    - Use phpMyAdmin to confirm that all tables (users, rooms, customers, reservations) exist and are populated correctly.
5. **Inspect Code Errors (For Admins)**
    - If pages are not loading, enable error reporting in PHP or check the Apache log file for more details.
    - Review syntax in recently modified PHP files.

6. **Test on a Different Browser**
    - Try accessing the system using another browser (Chrome, Edge) to rule out browser compatibility issues.

Contact Information for Technical Support

For further assistance, please contact the system developer or assign technical support personnel.

Technical Support Contact:
Name: Francis Galler Anacta
Email: anactafrancisrico@gmail.com
Phone: 09457528096

Name: Marjorie Bernadas
Email: majobrnds@gmail.com
Phone: 09957580295

Availability: Monday – Friday, 8:00 AM – 5:00 PM

If the issue cannot be resolved remotely, please coordinate with the system administrator or IT department for on-site troubleshooting.

CODE DOCUMENTATION

This section provides a detailed explanation of the program's structure, logic, and implementation approach. It ensures that developers, reviewers, and maintainers can clearly understand how the system functions internally. This documentation describes the organization of the source code, highlights key functions through inline comments, and presents the coding standards followed throughout the development of the **Hotel Reservation Management System (HRMS)**.

Code Structure and Organization

The Hotel Reservation Management System (HRMS) is developed using PHP, HTML, CSS, JavaScript, and MySQL. It follows a modular and organized folder structure to ensure clarity, maintainability, and scalability. The main directories and their purposes are as follows:

Folder / File	Description
/index.php	The homepage of the system where guests can browse and access key features.
/admin/	Contains administrative modules, including dashboard, room management, and reservation management pages.
/includes/	It contains reusable PHP files such as the database connection and configuration files.
/assets/	Stores static resources such as CSS, JavaScript, and image files used across the website.
/pages/	Contains guest-facing pages such as room listings, promos, and contact page.
/database/	May include SQL scripts or initialization files for database setup.

/emails/	Temporary directory for storing email files (used if email notifications are enabled).

Table 6: Folder Structure and Descriptions of the Hotel Reservation Management System (HRMS)

Inline Comments Explaining Key Functions and Logic

To improve code readability, the HRMS source code includes inline comments that describe the purpose and functionality of key code segments.

Here are examples of commonly used functions:

```
// Establish connection to the MySQL database
$conn = new mysqli("localhost", "root", "", "hrms");

// Function to retrieve all available rooms from the database
function getAvailableRooms($conn) {
    $sql = "SELECT * FROM rooms WHERE availability = 'Available'";
    $result = $conn->query($sql);
    return $result;
}

// Insert new reservation into the database
if (isset($_POST['reserve'])) {
    // Capture form input data
    $name = $_POST['name'];
    $email = $_POST['email'];
    $checkin = $_POST['checkin'];
    $checkout = $_POST['checkout'];
    $room_id = $_POST['room_id'];

    // Insert into reservations table
    $sql = "INSERT INTO reservations (customer_name, email, room_id, check_in, check_out, status)
            VALUES ('$name', '$email', '$room_id', '$checkin', '$checkout', 'Pending')";
    $conn->query($sql);
}
```

Image A: Sample PHP Functions with Inline Comments

Inline comments are used throughout the system to:

- Explain database queries and logic.
- Describe variable usage and expected outputs.
- Clarify complex code segments for future developers.
- Mark sections for updates or debugging.

Coding Standards and Conventions

The HRMS follows standard web development and coding best practices to ensure code consistency and maintainability.

1. **Naming Conventions**
    - Variable and function names use **camelCase** such as $roomType and getAvailableRooms().
    - File names use **lowercase_with_underscores** such as room_list.php and config.php.
    - Database tables use **singular nouns** such as room, reservation, and customer.
2. **Indentation and Formatting**
    - Four-space indentation for readability.
    - Consistent spacing around operators and function arguments.
    - Proper use of blank lines to separate logical sections.
3. **Commenting Style**
    - Use // for single-line comments and /* ... */ for multi-line comments.
    - Each function starts with a short description of its purpose.
4. **Security and Validation**
    - Form inputs are validated using both HTML5 attributes and PHP server-side checks.
    - Database queries use prepared statements or input sanitization to prevent SQL injection.
    - Passwords are hashed using password_hash() before being stored.
5. **Reusability and Modularity**
    - Common components such as header, footer, and navigation are stored as separate include files.
    - Functions and database operations are centralized in reusable modules.

6. **Error Handling**

    - Errors are caught and displayed using user-friendly messages.
    - System logs can be checked through PHP's error reporting for debugging.

TESTING DOCUMENTATION

The **Testing Documentation** section provides an overview of the testing strategies, test cases, and results conducted during the development of the **Hotel Reservation Management System (HRMS)**. Its purpose is to ensure that the system functions correctly, meets user requirements, and maintains performance, reliability, and security standards.

Test Plan

Objective

The main objective of testing is to verify that all modules of the HRMS perform according to their intended functions and to identify and fix potential defects before deployment.

Scope

Testing covers all major system components, including user authentication, room reservation, room management, and data handling in the database.

Testing Strategies

The testing process for HRMS follows a combination of **black-box testing** (focusing on functionality) and **white-box testing** (verifying internal code logic). The following strategies were used

1. **Unit Testing** – Testing individual functions and modules such as login, room addition, and reservation creation.
2. **Integration Testing** – Ensuring that interconnected modules (reservations and room availability) work together seamlessly.
3. **System Testing** – Verifying that the entire system performs correctly as a whole.
4. **User Acceptance Testing (UAT)** – Testing the system with intended users (Admin and Guest) to confirm that all functions meet expectations.

Testing Environment

- **Hardware:** Intel i5 or higher, 8GB RAM minimum
- **Software:** Windows 10, XAMPP Server (Apache, MySQL), Visual Studio Code, Google Chrome/Edge
- **Database:** MySQL

Test Cases

This subsection presents the list of functional test cases performed to validate the behavior of major system components. Each test case defines what was tested, the expected outcome, and the actual result.

Test Case ID	Module	Test Description	Expected Result	Actual Result	Status
TC-01	Login	Verify that admin can log in with valid credentials.	Admin dashboard loads successfully.	expected	Passed
TC-02	Login	Attempt to login with invalid credentials.	Displays “Invalid username or password” message.	expected	Passed
TC-03	Room Management	Add a new room with valid details.	Room is successfully added and displayed in the list.	expected	Passed

TC-04	Room Management	Edit an existing room's details.	Updated information is displayed correctly.	expected	Passed
TC-05	Reservation	Create a reservation with complete data.	Reservation record is saved in the database.	expected	Passed
TC-06	Reservation	Attempt to reserve without filling required fields.	Displays an input validation message.	expected	Passed
TC-07	Customer Records	View stored guest information.	Customer data is displayed accurately.	expected	Passed
TC-08	Availability	Booked rooms remain available (feature not yet implemented).	System does not change room status automatically.	Known limitation	Pending update
TC-09	Logout	Ensure the system logs out and redirects to login page.	Session ends and returns to login page.	expected	Passed
TC-10	Security	Attempt to access admin page without login.	Redirects to login page with restricted access.	expected	Passed

Table 7: System Component Testing – Test Cases and Results

Non-Functional Testing

This subsection evaluates the overall quality attributes of the system such as performance, usability, compatibility, security, and database reliability. These tests ensure the system not only functions correctly but also provide a smooth user experience.

Test Type	Description	Expected Result	Outcome
Performance Testing	Check page loading time for main modules.	Pages load within 2–3 seconds.	Passed
Compatibility Testing	Test system behavior on various browsers (Chrome, Edge, Firefox).	Displays correctly in all browsers.	Passed
Usability Testing	Evaluate navigation and overall user interface.	Interface is easy to use and consistent.	Passed
Security Testing	Test access restrictions for non-admin users.	Unauthorized users are blocked.	Passed
Database Testing	Verify that inserted data is saved correctly in MySQL.	Data is accurate and consistent.	Passed

Table 8: Non-Functional Test Cases and Results

Test Results and Defect Reports

Testing confirmed that the HRMS performs all essential operations effectively. However, one limitation was noted: room status does not automatically change to "Occupied" after a successful reservation. This feature is planned for a future update.

Defect ID	Module	Description	Impact Level	Status
D-01	Reservation	Room status does not automatically switch to Occupied.	Low	For future enhancement
D-02	Layout	Text overlap occurs on smaller screen sizes.	Minor	Fixed
D-03	Validation	System allows past dates when selecting reservation date.	Medium	Fixed

Table 9: Test Results Summary and Defect Report

MAINTENANCE GUIDE

The **Maintenance Guide** provides the procedures and practices necessary to ensure the continued performance, reliability, and improvement of the **Hotel Reservation Management System (HRMS)**. System maintenance is an essential part of the software life cycle, as it involves monitoring, updating, and improving the system based on user feedback, bug reports, and evolving requirements.

Maintenance Procedures

To maintain the system efficiently, regular monitoring and preventive actions are recommended. The following maintenance activities should be performed by the system administrator or designated IT personnel.

1. **Database Maintenance** – Regularly back up the MySQL database to prevent data loss. Backups should be stored securely in both local and external storage. Periodic optimization of tables in phpMyAdmin can help maintain database performance.

2. **System File Backup** – Create copies of all PHP, CSS, and JavaScript files before applying any modifications. This ensures that the system can be restored to a working state in case of unexpected errors.

3. **Error Log Review** – Check the server's error log in XAMPP to identify and resolve potential issues. This helps prevent downtime and data corruption.

4. **Security Updates** – Keep the XAMPP environment, PHP version, and browsers up to date to ensure compatibility and protection against vulnerabilities.

5. **Performance Monitoring** – Regularly assess page load times and database queries to maintain system responsiveness, especially as data volume increases.

Version Control and Release Management

Since the system is developed and deployed locally, **manual version control** is recommended for tracking system changes and updates. Developers should adopt the following version control practices.

- **Version Naming Convention -** Use a simple numeric format such as v1.0, v1.1, v2.0, etc., to label each updated version of the HRMS.
- **Backup Before Update -** Always back up the database and source files before installing or testing a new version.
- **Release Testing -** Test each version thoroughly before deployment to ensure new updates do not introduce new errors or conflicts.

When a new release is ready, replace the old version of the system files in the htdocs directory and import the updated database if necessary.

Bug Fixes and Enhancements

Bugs or errors may occasionally occur after deployment due to user interactions, data inconsistencies, or configuration changes. The following guidelines should be followed when handling these issues.

1. **Bug Identification -** Collect detailed information from users, such as screenshots or error messages, to identify the source of the problem.
2. **Error Reproduction -** Attempt to recreate the issue in a local testing environment to analyze its cause.
3. **Fix Implementation -** Apply the appropriate code changes or database adjustments to resolve the issue.
4. **Testing After Fix -** Conduct retesting to verify that the problem has been successfully fixed without affecting other system functions.
5. **Enhancements -** When adding new features (automatic room status updates or reporting tools), document the feature, test it separately, and integrate it only after successful evaluation.

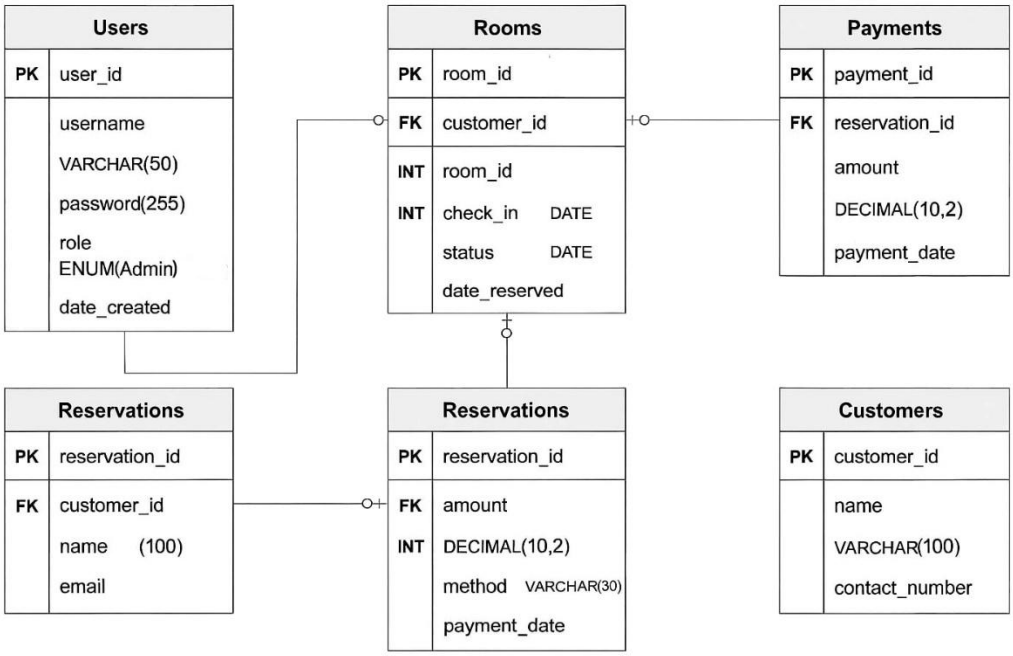
Regular system maintenance, combined with responsible version control and documentation, ensures that the HRMS remains reliable, efficient, and adaptable to future requirements.

APPENDIX

The **Appendix** section contains additional materials that support the understanding, design, and development of the **Hotel Reservation Management System (HRMS)**. These supplementary resources provide visual and technical references that complement the information presented in the main documentation.

Appendix A - Supporting Diagrams

- **A1 Entity Relationship Diagram (ERD)**



Appendix B - Reference Materials

- **Eurotel North Edsa Website (for conceptual reference only)**
  https://eurotel-hotel.com/Home