MAL2018 INFORMATION MANAGEMENT AND RETRIEVAL REPORT 2

BSSE2409191

**Assessment Report 2: TSC HOTEL BOOKING MICROSERVICE**

**1.Introduction**

This report is for Assessment MAL2018 Information and management retrieval coursework 2. The documentation and implementation for the TSC Hotel Booking Microservice. The Microservice allow customer to handle room booking efficiently through user friendly interface while maintaining backend functionality via Restful. Including Database, integration with PHP and secure design.

|  |  |
| --- | --- |
| GitHub Repository | <https://github.com/ZULKAMAL12/Assessment-2-Information-management-and-retrieval> |
| Testing | <http://risedragon.fwh.is>  username: [user@example.com](mailto:user@example.com)  password: userpassword123 |

This report details the background of the project, analysis and design considerations, implementation process, evaluation results, and legal, social, ethical and professional issue.

**2. Background**

This section presents the purpose of the TSC Hotel Booking Micro service and the problem its address. This also to improve scalability, maintainability and security.

TSC Hotel & Tourism need a modern booking system that allow customer to:

* Create account and Login.
* Book a room and mange Booking (edit, booking).
* View room Availability
* Make A payment

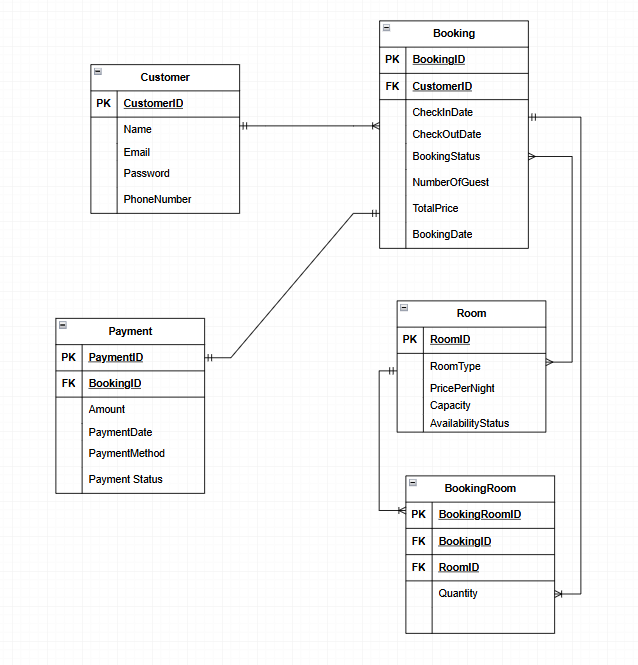
The Database Includes:

* Rooms: Categorized into Family room, Suite and Standard Deluxe.
* Customer: Registered user who can create and manage booking
* Payments: Securely processed and track for all booking

**3. Analysis and Design**

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

Ther ERD below illustrate the relationships between key entities, ensure proper normalization.



**Relationships**:

• **Customer**-to-**booking**: A customer can make multiple Booking, but each Booking

associated with a customer. (One-to-Many).

• **Booking**-to-**Room**: Booking is made for one Room, but one Room can have multiple

Booking on different date and times. (Many-to-One).

• **Payment**-to-**Booking**: Each Booking can have only one Payment record and each Payment

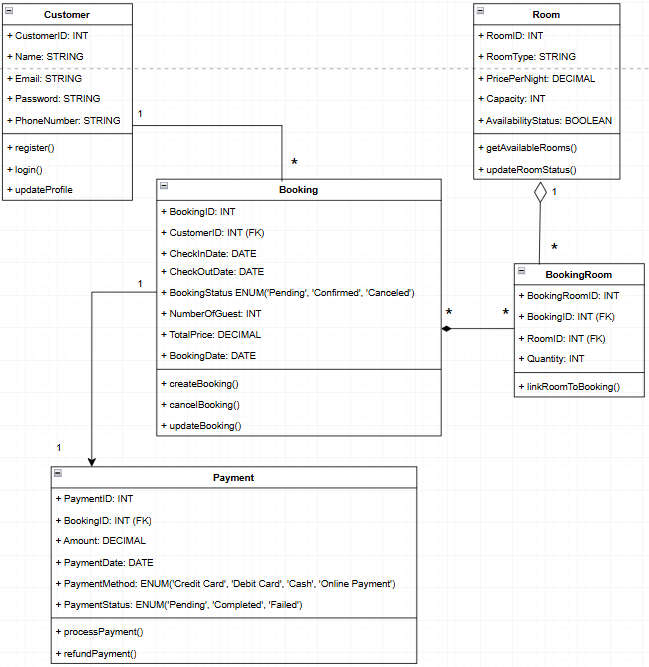
is associated with only one Booking. (One-to-One).

• **Many**-to-**Many** relationship between **Booking** and **Room** resolved via **BookingRoom**.

**3.2 UML DIAGRAM**

The UML diagram highlight the classes and methods implemented.

* **Customer**: Handle user information, authentication and account management.
* **Room**: Manage room availability.
* **Booking**: Support CRUD operation for Booking.
* **Payment**: Process and track payment
* **BookingRoom**: Link **Rooms** and **Bookings**.



3.3 Database Normalization

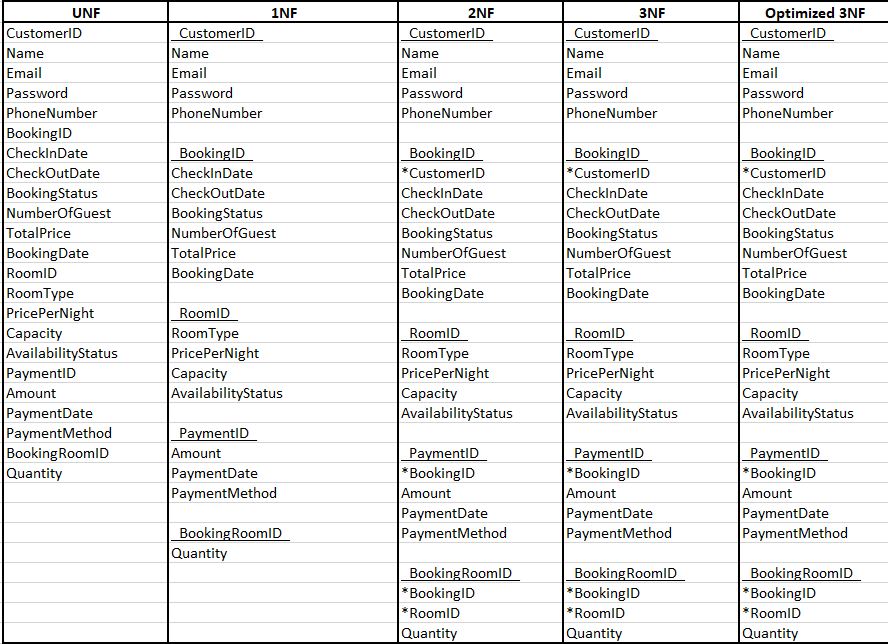
The Database Design

1. UNF: Original dataset with redundant information.

2. 1NF: Remove multi valued attribute.

3. 2NF: Remove partial dependencies.

4. 3NF: Remove transitive dependencies.



**4. Implementation**

Using Local database xampp

4.1 Backend

The backend uses PHP for handling RESTful API request.

* POST /register: Register new user.
* POST /login: Authentication users.
* POST /bookings: Create new booking.
* PUT /bookings/{id}: Update an existing booking.
* DELETE /bookings/{id}: Cancel a booking

4.2 Database

The Database Include Five Tables;

1. Customer
2. Room
3. Booking
4. Payment
5. BookingRoom

Stored procedure handles the key operation like booking creation and payment process. The trigger ensure room availability updated.

4.3 Views and Trigger

* Views: Display active booking with customer room and customer Details.
* Triggers: Automatically update room availability when booking created or cancelled.

**5. Evaluation**

Testing was conducted using browser-based form submissions.

* 1. Testing Result
* User Authentication: Verified secure login/logout process
* CRUD Operations: Ensure functionality for creating, reading, update and deleting bookings.
* Room Availability: Validate automatic updates via trigger
  1. Observation
* All functional requirement were met.
* Area for improvement
* Implement token-based authentication
* Enhance error handling for edge cases.

**6. Legal, Social, Ethical, and Professional**

6.1 Legal

* Compliance with Data protection act 2018 ensure secure user data handling.

6.2 Social

* Improve user experience through responsive and friendly design and navigation.

6.3 Ethical

* Secure coding practice prevent misuse of customer data.

6.4 Professional

* Adherence to OWASP top 10 guideline.

Privacy

* Password using password\_hash () before storage.
* Sensitive user and payment data is encrypted during transmission using HTTPS.

7. Conclusion

The report Outline the successful implementation of the TSC Hotel Booking Microservice. The project demonstration a database design, secure backend functionality and user-friendly feature. Future enhancement including adding advance security and integration notification.

Reference:

1. W3Schools (2025) ‘PHP MySQL Integration’, *W3Schools*. Available at: <https://www.w3schools.com/php/php_mysql_intro.asp> (Accessed: 1 January 2024).
2. YouTube *PHP Tutorial for Beginners - Step by Step Guide*. Available at: <https://www.youtube.com/watch?v=MV8AT9a2oSM&list=PLWxTHN2c_6cbh1C7yIskoXszoTl-okogt> (Accessed: 20 December 2025).
3. InfinityFree (2025) *Free Hosting Platform*. Available at: <https://infinityfree.net/> (Accessed: 1 January 2025).
4. OWASP Foundation (2024) *OWASP Top Ten Security Risks*. Available at: <https://owasp.org/www-project-top-ten/> (Accessed: 23 December 2025).