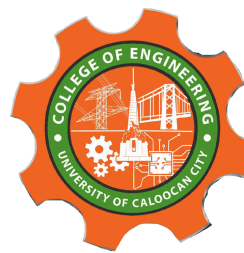




UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT



Software Design

Progress Report No. 8

Data Structures and Algorithms

Submitted by:

Group: BLDG.

Gabijan, Rhovic M.

Balana, Jerkielle O.

Balaoro, Judge Wayne B.

Barbas, Steven Jade P.

Dispo, Lei Andrew T.

Laput, Mark Danielle E.

Instructor:

Engr. Maria Rizette H. Sayo

January 24, 2026

I. Objectives

In this section, the goals in this laboratory are:

- To describe a good conceptual design of the system
- To identify one of the three levels of structure of program – Data Structure

II. Methods

One of the most difficult things for program readers to understand is the way in which data is structured and used. A data map is very useful in interpreting the code's actions, especially when the system handles many files of varying types and purposes.

Why do we need Data Structure?

- It gives different level of organizing data
- It explains how data can be stored and accessed
- It provides operation on group of data, such as adding an item, inserting an item, sorting items, deleting an item, and updating items
- It provides a mean to manage huge amount of data efficiently
- It provides fast searching and sorting a data

General Instructions:

Create an Algorithm and Flowchart, Data Flow Diagram/ERD of your proposed system (Use LucidChart to draw your flowchart and other drawing software)

III. Results

System Algorithm:

Step 1: Start the process.

Step 2: User visits the website.

Step 3: Check date availability:

- If the date is not available, display "Fully Booked" and proceed to Step 10.
- If the date is available, proceed to Step 4.

Step 4: The user fills out the form and uploads the payment receipt.

Step 5: Save the data to the database with a status set to "Pending".

Step 6: System/Manager performs an Admin Review.

Step 7: Validate the payment:

- If the payment is invalid, trigger "Reject Booking" and proceed to Step 10.
- If the payment is valid, proceed to Step 8.

Step 8: Update the booking status in the database to "Reserved".

Step 9: Send a confirmation email to the user.

Step 10: End the process.

System Flowchart

The system flowchart illustrates the step-by-step process of making a reservation in the Palacio Feliz Booking System. The process begins when the guest checks the availability of a selected date. If the date is unavailable, the system ends the process. If available, the guest proceeds to submit booking details and proof of payment. The system saves the booking as “Pending” and forwards it for admin verification.

The administrator validates the payment. If valid, the booking status is updated to “Reserved.” If invalid, the booking is rejected.

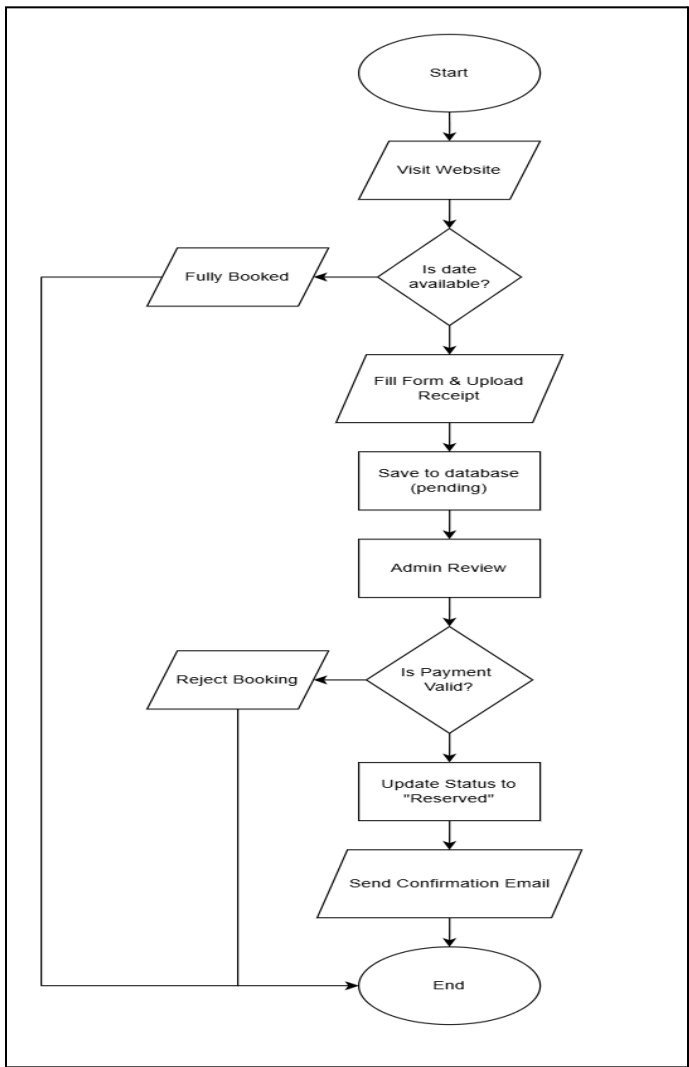


Figure 1. System Flowchart of the Palacio Feliz Booking System

Data Flow Diagram:

The Level 1 Data Flow Diagram of the Palacio Feliz Booking System is derived from the previously presented Level 0 (Context Diagram). While the Level 0 diagram shows the system as a single process, the Level 1 diagram expands it into detailed internal processes.

It illustrates how data flows between the Guest, the Administrator, and the system processes such as Check Availability, Submit Booking, Verify Payment, and Manage System. The diagram also shows how booking, payment, and admin data are stored in their respective data stores.

This provides a clearer view of how the system processes and manages booking transactions.

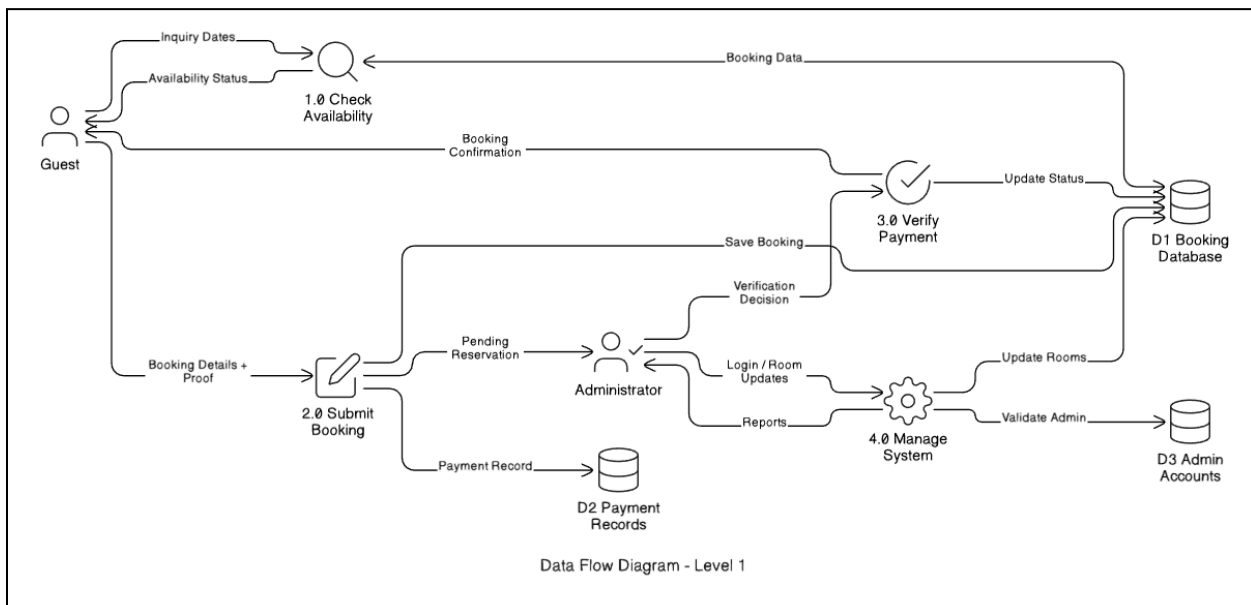


Figure 2. Data Flow Diagram (Level 1) of the Palacio Feliz Booking System

Entity Relationship Diagram:

To Complement the architectural design, the **Entity-Relationship-Diagram (ERD)** illustrates the database structures and how information is stored within the **Palacio Feliz Booking System**. The diagram identifies four key entities and their relationships to ensure data integrity during the booking process:

Data Relationships and Entities (ERD LEVEL 1):

- **Guest:** Represents the customer. It stores identity and contact details (Name, Email, Phone) required to facilitate communication and booking ownership.

- **Booking:** The central transaction entity. It records the specific "Inquiry Date" and the current lifecycle stage of the reservation (Pending, Reserved, or Rejected).
- **Payment:** A supporting entity that holds the "Proof of Payment" (receipt upload) and the validation status determined during the Admin Review.
- **Admin:** Represents the system's internal users who have the authority to process payments and update booking statuses.

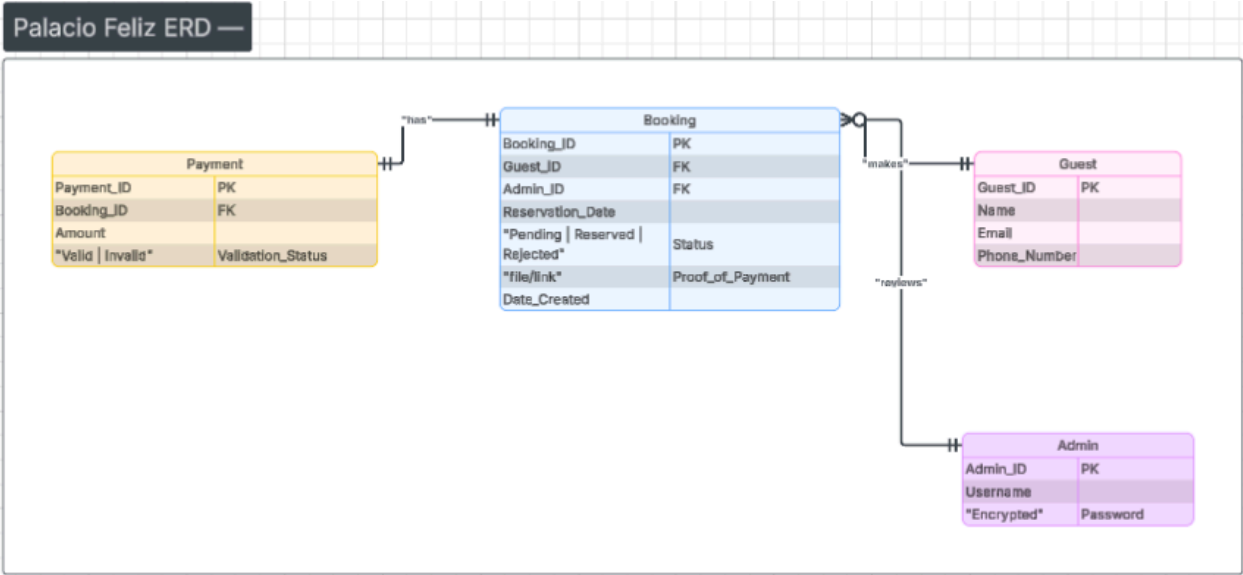


Figure 3. Entity Relationship Diagram (Level 1)

System Cardinality

- **One-to-Many (1:N):** A single **Guest** can make multiple **Bookings** over time, and an **Admin** can manage many different **Bookings**.
- **One-to-One (1:1):** Each **Booking** is strictly associated with one **Payment** record to ensure that every reservation is accounted for with a unique proof of transaction.

Hardware and Software Configuration

- **Database Management:** The system uses a Relational Database Management System (RDBMS) to maintain the links between Foreign Keys (FK) like Guest_ID and Admin_ID within the Booking table.
- **Data Security:** Sensitive information, specifically the Admin Password, is stored as "Encrypted" data to prevent unauthorized access to the booking management portal.

IV. Conclusion

The Palacio Feliz Booking System successfully transforms a manual reservation process into a structured digital workflow. Through the use of a system flowchart, the logical steps of the booking process were clearly illustrated. The Level 1 Data Flow Diagram expanded the previous Level 0 context diagram to show detailed data movement between entities, processes, and data stores. Finally, the ERD defined the database structure, ensuring data integrity and proper relationships among Guest, Booking, Payment, and Admin entities.

References

- [1] Co Arthur O.. "University of Caloocan City Computer Engineering Department Honor Code," UCC-CpE Departmental Policies, 2020.

- [2] A. Dennis, B. H. Wixom, and D. Tegarden, Systems Analysis and Design: An Object-Oriented Approach with UML, 6th ed. Hoboken, NJ, USA: Wiley, 2021.

- [3] I. Sommerville, Software Engineering, 10th ed. Boston, MA, USA: Pearson, 2016.

- [4] E. Yourdon, Modern Structured Analysis. Englewood Cliffs, NJ, USA: Prentice Hall, 1989.