



MUSEUM MANAGEMENT SYSTEM

University Software Engineering Project

| |
|--|
| Philip Griffin M00961140 - Secretary |
| Foyzur Rahman M00948381 - Code Dev |
| Tamjid Karim M00948372 - Code Dev |
| Tanweer Ali M00946884 - Code Tester |
| Abdallah Ahmed M00940645 - Team Leader |

Contents

| | |
|--|---|
| Introduction:..... | 2 |
| Project Description:..... | 2 |
| Report Layout:..... | 2 |
| Design: | 2 |
| Data Structures: | 2 |
| Bookings:..... | 2 |
| Customer: | 3 |
| CustomerExpeditions: | 3 |
| Expeditions:..... | 3 |
| Managers:..... | 3 |
| Algorithms: | 3 |
| ERD Diagram: | 4 |
| Testing: | 0 |
| Conclusion: | 0 |
| Work Completed: | 0 |
| Limitations and Critical Reflections:..... | 0 |
| Future Changes: | 0 |
| References: | 0 |

Introduction:

Project Description:

This project is for a Museum Management system allowing for several features to be added for managing and maintaining a system for a museum. This also includes the creation and storage of a database holding the data for customers' information along with their booked expeditions. The goal for this project is to implement all necessary features a system would need that would allow for ease of use managing and maintaining a museum. The users of this system will be the management and staff of a museum so the functions will reflect this user base. Planned features for the system will include viewing customers, bookings and expeditions; along with adding and handling the earnings and control over employees in terms of hiring along with firing. For inspiration for our system, we looked into existing museum systems and structures such as the British Museum. ¹

Report Layout:

This report will be structured to allow sections for design, testing, conclusion and for necessary references. Design will present the chosen data structures and algorithms used for the system and will justify why these were used, using time complexity as one key indicator as to their usage. The algorithms will be written in pseudocode. Testing will show the approach we used for testing the reliability of the system along with a table of test cases to give idea of how what we tested. Conclusion will be summary of work we were able to complete with what features that were added successfully along with a reflection on the limits that the system has and what caused them; as well as how our approach would change in a future task similar to this.

Design:

Data Structures:

For our program we created appropriate data structures that would ensure we can fully control and store data in an efficient and streamlined manner.

Bookings:

Table: BookingID, CustomerID, ExpeditionID, BookingDate, TimeSlot, Price, Status.

This data structure was chosen as a way to store all necessary data for a given booking including the customer booked, the expedition and information about said booking. This also allows for several tables to be linked through foreign keys such as CustomerID and ExpeditionID. Along with this fields such as BookingDate and TimeSlot allow for ease of scheduling bookings and expeditions as they allow for time-based queries and conflict resolutions. Fields such as Price allow for the system to further track important features such as payment, a key to a management system.

¹ The British Museum (n.d.).

Customer:

Table: CustomerID, Name, Email

This table was chosen as a way to store customer data away from bookings and allows for a customer to book several expeditions without the need to copy their data such as email.

CustomerExpeditions:

Table: CustomerID, ExpeditionID

This table is used as a junction table used to link a customer to a certain expedition. This table acts as a many-to-many relationship as it allows a customer to go on many expeditions along with an expedition to have many customers. The table keeps the database structure clean and avoids duplication of data.

Expeditions:

Table: ExpeditionID, Name, Type, Description, ExpeditionName, TimeSlot

This table holds the main data for all expeditions. Fields such as Type and TimeSlot allow for ease of sorting and filtering along with fields such as Description provide meaningful data for users and managers when managing the system.

Managers:

Table: ManagerID, Username, Password, Role

This was chosen as a means to store the necessary data for authorization to the management system as only managers and admins should have access to said system. The Role field allows for several roles to be within the system and so different levels of functionality can be added to each role such as admin not being able to change expedition, but a manager can.

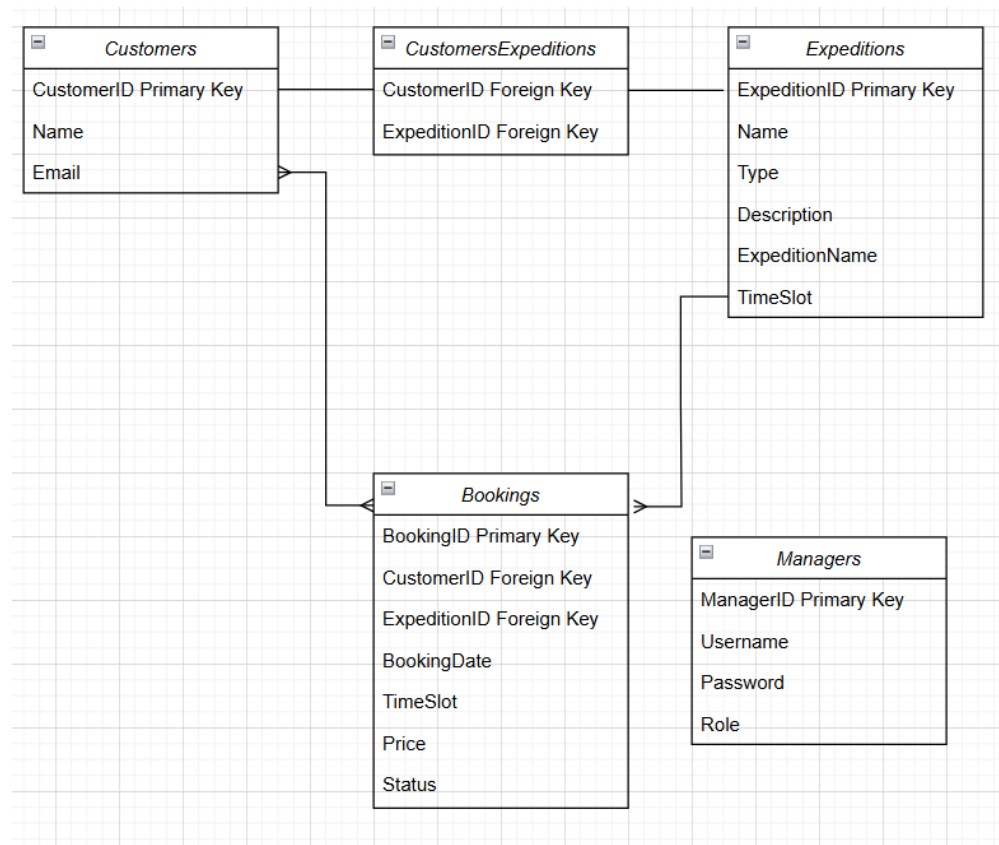
Algorithms:

Why we chose to implement key functionality with pseudocode.

The C# code shared across various forms in the museum tour management application primarily serves to manage and display data in an organized, user-friendly manner. Each form, such as the Customer Form, Bookings Form, Expedition form, and AvailableExhibitionsForm, initializes and configures controls like DataGridView to display relevant information (e.g., customers, bookings, expeditions, and exhibitions). These controls are dynamically populated with data retrieved from the database, using methods such as GetAllCustomersDataTable() or GetAvailableExhibitions (). This ensures that the user can interact with and view up-to-date information in an intuitive layout.

In addition to displaying data, the code also includes functionality for managing user interactions, such as the "Back" button to navigate away from the current form. The use of properties like Dock, AutoSizeColumnsMode, and Selection Mode enhances the responsiveness and usability of the interface, ensuring it adapts to various screen sizes and allows users to easily select and view the details of each row in the DataGridView. These designs ensure smooth interaction with the system. **The File above shows our pseudocode.**

ERD Diagram:



Testing:

[illegible]

Conclusion:

Work Completed:

For this project of a Museum Management System our group has created a functional system that should hold most functionality such a system would need to work well. The system stores and handles all relevant data that a museum management system would need through a thought out and well implemented database. The database was designed and implemented to be as efficient as needed with necessary functions to manipulate the database for the systems features. Along with this the system uses an intuitive and pleasing UI to allow the users for the system to easily use all functions and features of the system without struggle. The system allows for the user (manager) to view important information such as daily earnings, current employees and all bookings with the function to filter via time slot.

Limitations and Critical Reflections:

On reflection of this project, not enough time was placed on designing the data structures to ensure more functionality can be implemented. This may have been caused by the limited experience our group had in designing and implementing a whole data structure within a big project. Another limit would be the complexity of algorithms that were used within the system, this could have been caused by going with complex functionality to the system.

Future Changes:

Ensure more time will be used to design the system more thoroughly with regard to its later implementation. Along with proper ordering of necessary functionality to ensure more complex features are to be added.

References:

The British Museum (n.d.). *Welcome to the British Museum*. [online] British Museum. Available at: <https://www.britishmuseum.org/>.