



**University of Baltistan  
Public College Skardu**



## ***Olympic Sports City Booking System*** **(Data Structure and Algorithm)**

**By:**

*Mr. Karamatullah Payalo — Reg# 2024-uobsac- 1370*

*Mr. Zeeshan Saleem — Reg# 2024-uobsac-1396*

*Mr. Muhammad Musa — Reg# 2024-uobsac-1363*

*Mr. Yasir Ahmed — Reg# 2024-uobsac-1402*

**Submitted to:**

Mr. Sajid Ali

Department of Computer Science

## Table of Contents

|                                   |   |
|-----------------------------------|---|
| Abstract.....                     | 3 |
| Introduction .....                | 3 |
| Targeted Audience .....           | 3 |
| Significance of the Project ..... | 4 |
| Scope.....                        | 4 |
| Source Code Explanation .....     | 4 |
| Conclusion.....                   | 8 |
| References .....                  | 9 |

## Abstract

The 'Olympic Sports City Booking System' is a Python-based application developed to streamline and digitize the process of managing sports facility bookings. This system demonstrates the effective integration of Data Structures and Algorithms (DSA) with GUI-based interaction. Users can easily book, search, cancel, and view sports reservations. Data persistence is achieved through file handling, while efficiency is maintained via sorting and searching algorithms. The GUI, developed using Tkinter, enhances user experience by providing a friendly and interactive interface for all booking operations.

## Introduction

The rapid growth of digital systems has made manual record-keeping obsolete in most industries. Sports facilities, in particular, require efficient systems to handle reservations, avoid double-bookings, and manage fee calculations. The 'Olympic Sports City Booking System' was built to address these challenges. It allows users to register bookings by entering player details, sports names, and time slots, while automatically preventing conflicts.

This project integrates key DSA principles including linear searching for locating bookings, bubble sorting for arranging records alphabetically, and conditional checks to ensure time-slot accuracy. The Tkinter GUI provides a more practical interface for real-world use, allowing both beginners and administrators to manage bookings effortlessly.

## Targeted Audience

The target users of this system include:

- **Students** — to understand and implement Data Structures and Algorithms (DSA) in practical software development projects.
- **Sports Administrators** — to automate the process of booking, managing, and scheduling sports games efficiently.
- **Developers** — to learn how backend algorithms and file handling can be integrated with graphical user interfaces (GUI) using Python's Tkinter library.
- **Educational Institutions** — to adopt this system as a model project that demonstrates the real-world application of DSA concepts in management systems.

## Significance of the Project

This project is significant both educationally and practically. From a technical perspective, it applies data structures to efficiently manage and process information. The sorting algorithm ensures that all records remain organized, while the searching algorithm enables quick lookups. The conflict detection function is a real-world application of interval comparison logic, which is widely used in scheduling software. The GUI integration through Tkinter adds an additional layer of accessibility, eliminating the need for command-line operations. It transforms a typical DSA project into a user-oriented application, demonstrating how backend algorithms can enhance real-world usability.

## Scope

Currently, the system supports text-based storage through a file handling mechanism, providing a simple yet functional method to maintain booking records. In the future, it could be expanded by integrating databases such as MySQL or SQLite to handle larger datasets efficiently. It may also evolve into a cloud-based system with online booking options and advanced reporting capabilities.

The GUI can be enhanced with additional design frameworks like CustomTkinter for modern aesthetics. Further improvements could include authentication systems, data visualization for usage patterns, and notification features for booking reminders.

## Source Code Explanation

Import necessary libraries:

```
import os
import random
import re
from datetime import datetime
import tkinter as tk
from tkinter import messagebox, simpledialog

# -----
```

The system follows a modular structure, combining Object-Oriented Programming (OOP) and DSA principles:

1.Booking Class — Defines the structure and attributes of each booking record.

```
# -----  
class Booking:  
    def __init__(self, booking_id, name, contact, game, start_time, end_time, fee, date=None):  
        self.booking_id = booking_id  
        self.name = name  
        self.contact = contact  
        self.game = game  
        self.start_time = start_time  
        self.end_time = end_time  
        self.fee = fee  
        self.date = date if date else datetime.now().strftime("%Y-%m-%d")
```

2. File Handling — Implements read and write operations using Python's file system for persistent storage.

```
# -----  
# File Handling Functions  
# -----  
def save_to_file(bookings):  
    with open("bookings.txt", "w") as f:  
        for b in bookings:  
            f.write(f"{b.booking_id},{b.name},{b.contact},{b.game},{b.start_time},{b.end_time},{b.fee},{b.date}\n")
```

3. Conflict Checking — Validates time slots using conditional logic to prevent overlapping reservations.

```
8 # -----  
9 # Algorithms  
0 # -----  
1 def is_conflict(bookings, game, start, end):  
2     for b in bookings:  
3         try:  
4             if b.game.lower() == game.lower():  
5                 if not (int(end) <= int(b.start_time) or int(start) >= int(b.end_time)):  
6                     return True  
7         except Exception:  
8             continue  
9     return False
```

4. Searching & Sorting Algorithms — Linear search and bubble sort enhance data retrieval and organization.

```

def search_by_id(bookings):
    key = input("Enter booking ID to search: ")
    for b in bookings:
        if b.booking_id == key:
            print("\nBooking Found:")
            print(b)
            return
    print("\nBooking not found!")

def search_by_name(bookings):
    name = input("Enter name to search: ").lower()
    found = False
    for b in bookings:
        if b.name.lower() == name:
            print(b)
            found = True
    if not found:
        print("No booking found for this name.")

```

5. Tkinter GUI Module — Replaces command-line inputs with a graphical interface featuring buttons, labels, and entry boxes for user interaction.

```

title = tk.Label(root, text="Olympic Sports City Booking System", font=("Arial", 20, "bold"), bg="#90caf9",
title.pack(fill="x", pady=10)

# Buttons
btns = [
    ("Add Booking", self.add_booking),
    ("View Bookings", self.view_bookings),
    ("Search Booking", self.search_booking),
    ("Cancel Booking", self.cancel_booking),
    ("Sort Bookings", lambda: sort_bookings(self.bookings)),
    ("Fee Structure", self.fee_structure),
    ("Exit", root.quit)
]

for text, cmd in btns:
    tk.Button(root, text=text, command=cmd, font=("Arial", 12), bg="#64b5f6", fg="white",
width=25, pady=5, relief="raised").pack(pady=5)

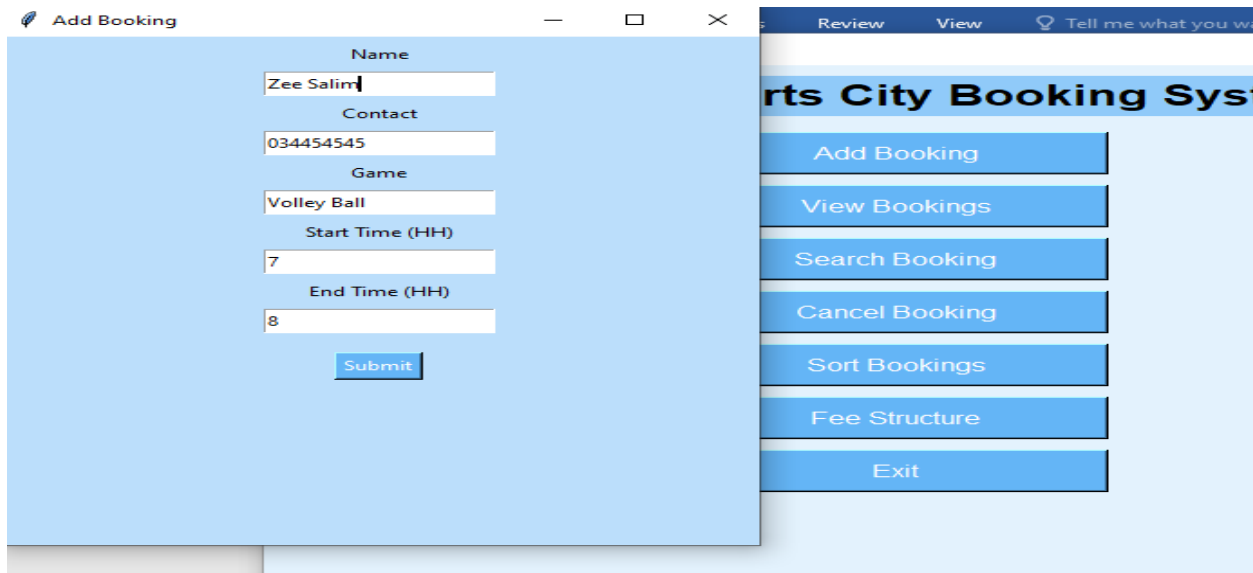
```

6. Integration Layer — Connects GUI events (like button clicks) with backend logic functions for seamless operation.

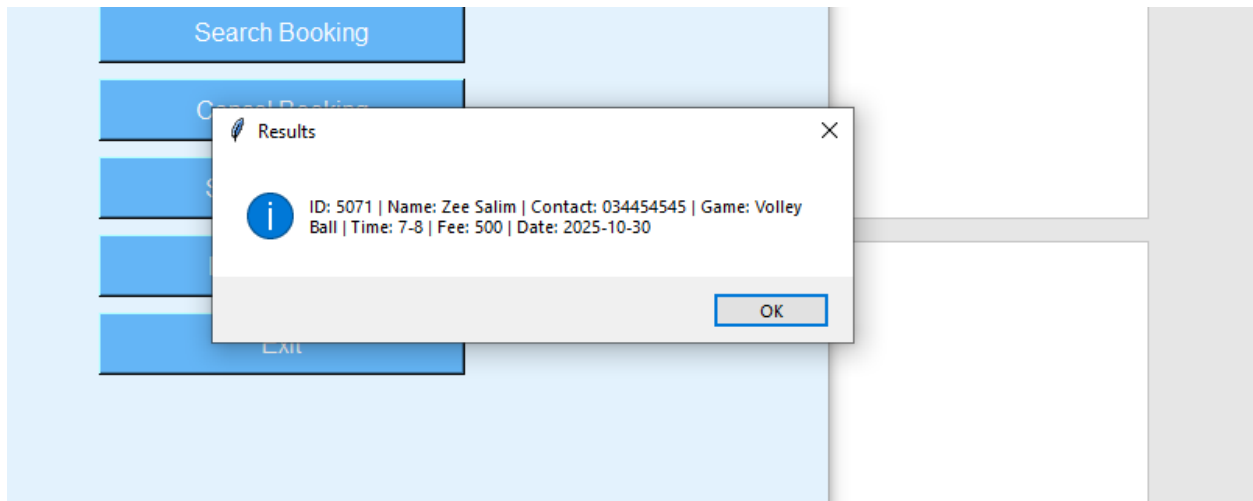
7. Visualization – Menu will be display as when we run the program



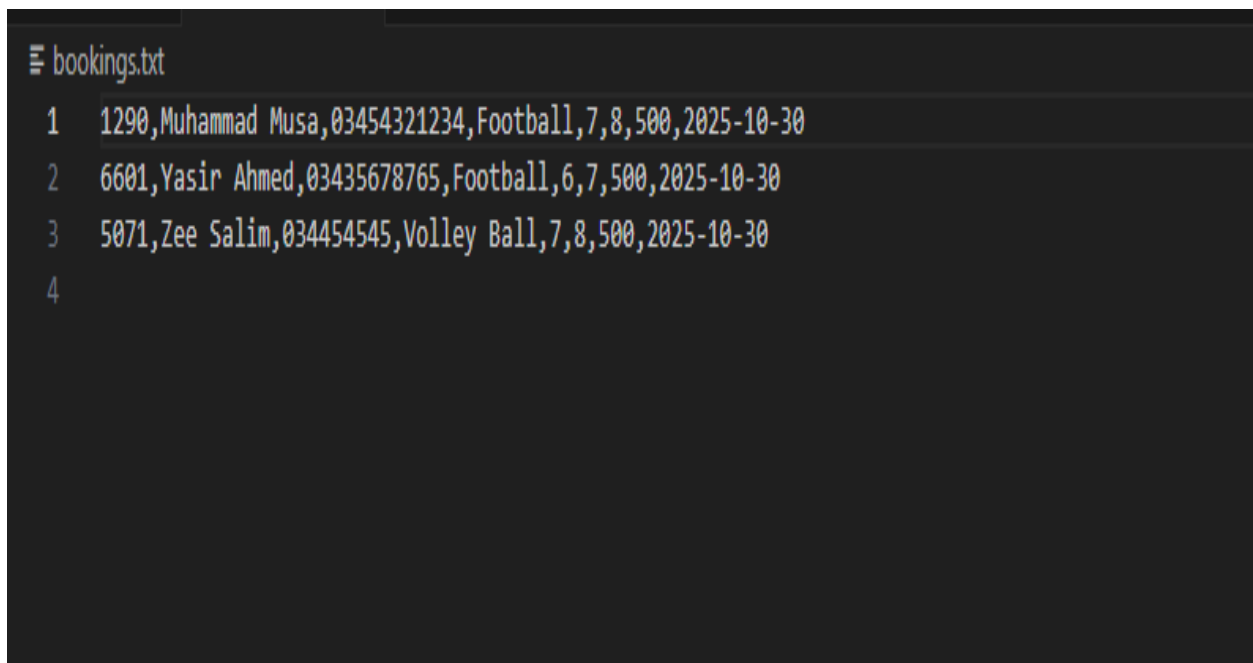
#### 8. Booking System – Required information when we booking



#### 9. Searching System – You can search both by names by id or by games



10. Sorting System – When we sort it sort by names in the txt file



This combination of OOP, algorithms, and GUI illustrates how academic DSA concepts can be effectively utilized to create user-oriented applications.

## Conclusion

The 'Olympic Sports City Booking System' project successfully combines theoretical and practical computer science concepts. It demonstrates how efficient algorithms and data structures can

optimize everyday tasks like facility management. Through this project, students gain hands-on experience in programming, logical thinking, and interface design. The addition of a Tkinter-based GUI transforms it from a typical academic exercise into a real-world, usable application. Future enhancements such as database integration, online connectivity, and advanced analytics will further extend its utility and performance.

## **References**

Abd Rahman, Siti Fatimah. "Online Sport Booking System." (2008).

Seijal Reino, Brais. "Web application for booking management and the organization of events in a sports town." (2023).

Mohd Azmir M, Muhd Zain N. Sport Facilities and Equipment Booking System for UiTM Perlis.

-----