

Programming Project / C++ – Online Parking System –

Student 1: Lazau Andrei-Tudor

I. Task Description

Student 1 is responsible for managing the parking system:

- Adding, deleting, modifying parking spaces (status, hourly rate)
- Managing parking reservations
- Viewing payment records
- Generating occupancy reports

Student 1 is responsible for interacting with the parking system:

- Searching for available parking spaces
- Making, modifying, and canceling reservations
- Processing payments for parking usage
- Rating previously used parking spaces

II. Data Structures Used by the Team

The following classes will be used:

- **ParkingSpace**: string spaceID, string location, string status, double hourlyRate, int floor
- **TimeSlot**: int startHour, int startMinute, int endHour, int endMinute
- **Date**: int day, int month, int year
- **Reservation**: string reservationID, string spaceID, string userID, Date date, TimeSlot timeSlot, double totalCost
- **Payment**: string paymentID, string reservationID, double amount, Date date, string status
- **Rating**: string spaceID, string userID, int score, string comment

III. File Structure

The following files will be used:

parking_spaces.txt

A file where details about parking spaces will be stored in the form:

<number of spaces>
<space1 ID> <space1 location> <space1 status> <space1 hourlyRate> <space1 floor>
<space2 ID> <space2 location> <space2 status> <space2 hourlyRate> <space2 floor>
...

reservations.txt

A file where details about parking reservations will be stored:

<number of reservations>
<reservation1 ID> <space1 ID> <user1 ID> <date> <startHour:startMinute>
<endHour:endMinute> <totalCost>
<reservation2 ID> <space2 ID> <user2 ID> <date> <startHour:startMinute>
<endHour:endMinute> <totalCost>
...

payments.txt

A file where payment details will be stored:

<number of payments>
<payment1 ID> <reservation1 ID> <amount> <date> <status>
<payment2 ID> <reservation2 ID> <amount> <date> <status>
...

user_cart.txt

A file where Student 2 will store the temporary reservation details:

<space ID> <date> <startHour:startMinute> <endHour:endMinute>

ratings.txt

A file where parking space ratings will be stored:

<number of ratings>
<space1 ID> <user1 ID> <score> <comment>
<space2 ID> <user2 ID> <score> <comment>
...

IV. Interacting with Executables

Application 1 (Admin Panel) will offer the following options:

`./parking_admin.exe view_spaces`

To view all parking spaces in the system

`./parking_admin.exe add_space <spaceID> <location> <status> <hourlyRate> <floor>`

To add a new parking space to the system

`./parking_admin.exe delete_space <spaceID>`

To delete a parking space from the system

`./parking_admin.exe modify_space <status | hourlyRate> <spaceID> <new_value>`

To modify either the status or hourly rate of a parking space

`./parking_admin.exe view_reservations [<date>]`

To view all reservations or reservations for a specific date

`./parking_admin.exe view_payments [<date>]`

To view all payments or payments for a specific date

`./parking_admin.exe generate_report <date>`

To generate an occupancy report for a specific date

Application 2 (User Interface) will offer the following options:

`./parking_user.exe search <date> <startTime> <endTime> [<location>]`

To search for available parking spaces for a time period

`./parking_user.exe view_cart`

To view the current reservation in the cart

`./parking_user.exe add_reservation <spaceID> <date> <startHour:startMinute>
<endHour:endMinute>`

To add a reservation to the cart

`./parking_user.exe modify_reservation <startHour:startMinute | endHour:endMinute>
<new_value>`

To modify the reservation time in the cart

`./parking_user.exe cancel_reservation`

To cancel the reservation in the cart

`./parking_user.exe confirm_reservation <userID>`

To confirm and save the reservation

`./parking_user.exe pay <reservationID> <paymentMethod>`

To process payment for a reservation

`./parking_user.exe rate <spaceID> <score> <comment>`

To rate a previously used parking space