

LAB211 Assignment

Type: Long Assignment
Code: J1.L.P0034
LOC: 300
Slot(s): N/A

Title

Homestay Booking Management

Background

GroupTrip, a travel company specializing in organizing group tours suitable for dynamic and creative young people, aims to digitize the tour booking process for its customers.

Information on homestays and tours is stored in two files: **Homestays.txt and Tours.txt**.

The traveler's booking information will be saved in the **Bookings.txt file**.

| Homestays.txt | Description |
|--|---|
| HS0001-Alee DaLat Homestay-3-12A/6 3rd February Street, Ward 1, Da Lat City, Lam Dong Province-15 | Information on a line: < homeID, homeName, roomNumber, address, maximumcapacity> |
| HS0002-Tournesol Villa Homestay-4-12 Van Kiet Street, Ward 3, Da Lat City, Lam Dong Province-18 | + Maximumcapacity: maximum number of tourist. |
| HS0003-HomeStay Butter House-2-Agency Sub-district, Farm Town, Moc Chau, Son LaProvince-10 | |
| HS0004-Moc Chau Hill House-3-Tay Tien Hill, Sub-district 12, Moc Chau, Son La Province-12 | |
| HS0005-Papa's Garden Homestay-3-1 Phuong Thanh Street, Me Thuong Village, Ha Giang 1 Ward, Tuyen Quang Province-10 | |
| HS0006-Dong Van CliffSide House 4-10 Don Cao, Dong Van Town, Ha Giang Ward, Tuyen Quang Province-14 | |

| File Tours.txt | Description |
|---|--|
| T00001,TPHCM-Da Lat,3 days 2 nights,300,HS0001,10/01/2026,12/01/2026,5,TRUE | Information on a line: < tourID, tourName, time, price, homeID, departure_date, end_date, number_Tourist, booking > |
| T00002,TPHCM-Da Lat,3 days 2 nights,280,HS0002,10/01/2026,12/01/2026,6,TRUE | + Price: Calculated per tourist/tour. |
| T00003,TPHCM-Moc Chau,4 days 3 nights,350,HS0003,14/01/2026,17/01/2026,3,TRUE | + booking : The default value is TRUE. |
| T00004,TPHCM-Moc Chau,4 days 3 nights,350,HS0003,19/01/2026,22/01/2026,3,TRUE | |
| T00004,TPHCM-Moc Chau,4 days 3 nights,400,HS0004,19/01/2026,22/01/2026,3,TRUE | |

Constraints

1- Constraints on Tours:

- tourID must be unique.
- tourName cannot be empty.
- time cannot be empty.
- price must be a positive number integer.
- homeID cannot be empty, must follow the format “HS0000” and it must exist on the HomeStay list.
- departure_date cannot be empty, must follow the format “dd/MM/yyyy”.

- g. end_date cannot be empty, must follow the format “dd/MM/yyyy” and it must be less than or equal to the departure date.
- h. number_Tourist must be a positive number integer and smaller than the corresponding HomeStay Maximumcapacity.

Note: You cannot create a tour where departure_date or end_date falls within the range [departure_date and end_date] of another tour with the same homeID.

2- Constraints on bookings:

- a. bookingID must be unique, must follow the format “B00000”.
- b. fullName cannot be empty.
- c. tourID must exist in the tour list contained in the Tours.txt file.
- d. booking_date cannot be empty, and it must be less than the departure date of the corresponding tour.
- e. phone cannot be empty and it consists of 10 digits.

Program Specifications

Students are required to build a **Java console application** using the OOP approach **to manage tours and bookings in the Homestay**.

The program must implement the following functions:

- 1- Add a new Tour
- 2- Update a Tour by ID
- 3- List the Tours with departure dates earlier than the current date
- 4- List the total Booking amount for tours with departure dates later than the current date
- 5- Add a new Booking
- 6- Remove a Booking by bookingID
- 7- Update a Booking by bookingID
- 8- List all Booking by the fullName or a partial fullName.
- 9- Statistics on the total number of tourists who have booked homestays
- 10- Quit program

Features and Assessment:

| No. | LOCs | Function |
|-----|------|---|
| 1/ | 20 | Automatic Data Loading: At startup, the program loads data from the required files. If any mandatory file is missing, an error message is displayed and the program terminates. |
| 2/ | 30 | Add a new Tour: Accept tourID, tourName, time, price, homeID, departure_date, end_date, number_Tourist, Add to the list if constraints are satisfied. Display success/failure message. |
| 3/ | 30 | Update a tour by tourID: If the tour does not exist, show: “ <i>This tour does not exist!</i> ”. Otherwise, allow updating tourName, time, price, homeID, departure_date, end_date, number_Tourist, booking. Empty input means skipping the update for that field. |
| 4/ | 30 | List the tours with departure dates earlier than the current date: No data input. |

| | | |
|-----|------------|--|
| 5/ | 30 | List the total booking amount for tours with departure dates later than the current date: No data input. Sort the data in descending order by Total Amount. |
| 6/ | 20 | Add a new Booking: Accept bookingID, fullName, tourID, booking_date, and phone. Validate constraints. Update the booking field in the tour list to True. |
| 7/ | 20 | Update a Booking by ID: Accept bookingID. If the booking does not exist, show: "This Booking does not exist!". Otherwise, allow updating fullName, tourID, booking_date, and phone. Empty input means skipping the update for that field. |
| 8/ | 20 | Remove a Booking by ID: Accept bookingID. If the Booking does not exist, show: "This booking does not exist!". Otherwise, remove it. After Removed, the booking and number Tourist field in the tour list will be able to updated. |
| 9/ | 20 | List all Booking by the fullName or a partial fullName: Accept fullName or partial fullName (e.g., Binh); display all matching bookings in the table format. |
| 10/ | 50 | Statistics on the total number of tourists who have booked homestays: statistics on the total number of tourists who have booked each homestay (HomeName, Number_Tourist) |
| 11/ | 30 | Quit program: Upon exit, if any related data files have been modified, the system shall prompt the user to save the changes and act according to the user's selection. |
| | 300 | |

Note for Students:

- **Follow OOP principles (encapsulation, inheritance, polymorphism, abstraction).**
- **Follow Computational Thinking (Decomposition, Pattern Recognition, Abstraction, and Algorithm) for reporting and presentation.**
- **Input validation must strictly follow constraints.**
- **Show appropriate messages for invalid or missing data.**
- **Evaluation: Understanding, Correctness, data validation, and coding style.**
- **Submission**
 - o **Source code**
 - o **Diagram**
 - o **Flow-chart**