

Travel Packing Rotation System

COP4530 – Programming Assignment 1

1. Group Members

- Isabela Shwindt Azevedo
- Mariah Rangel Barreto

2. Project Description

The Travel Packing Rotation System is a C++ console application designed to help users:

- Manage packing items
- Track trip destinations
- Rotate outfits efficiently

The idea was inspired by organizing a real spring break trip to Finland. The program demonstrates the use of linked list data structures in our context.

The system implements:

- Singly Linked List → Packing List
- Doubly Linked List → Trip History
- Circular Linked List → Outfit Rotation

All linked lists are implemented using templates, allowing them to accept arbitrary data types.

In addition to the linked list structures, we implemented the following domain classes:

- PackingItem
- TripRecord
- Outfit

The program follows a clean layered structure:

- Domain Layer (data classes)
- Structures Layer (templated linked lists)
- Service Layer (business logic)
- Main Program (menu-driven interface)

3. How to Compile and Run

Step 1: Navigate to Project Folder
`cd project1`

Step 2: Compile
`g++ -o travel_system main.cpp services/*.cpp domain/*.cpp`

Step 3: Run
`./travel_system`

4. How to Interact With the Program

The program uses a menu-driven console interface.

Main Menu Options:

1. Manage Packing List
2. Manage Trip History
3. Manage Outfit Rotation
4. Exit

Packing List (Singly Linked List)

- Add item
- Remove item
- Display all items

Trip History (Doubly Linked List)

- Add trip
- Display trips forward
- Display trips backward

Outfit Rotation (Circular Linked List)

- Add outfit
- Move to the next outfit
- Show current outfit

5. Work Distribution

- Isabela Shwindt Azevedo – 50%
 - Implemented all linked list data structures
 - Integrated full system
- Mariah Rangel Barreto – 50%
 - Implemented Service Layer
 - Applied Clean Architecture concepts learned during her summer internship

Both teammates participated actively and collaborated using GitHub. Repository: <https://github.com/mariahrb/COP4530/tree/main/project1>
Since this is a small-scale academic project, all commits were pushed directly to the main branch.

6. Code Quality

- All files are clearly documented with comments.
- Function names reflect their purpose.
- Variable names correspond to the data they hold.
- Code follows clean coding practices and separation of concerns.

7. Academic Integrity Statement

"We understand that there will be no tolerance towards academic dishonesty, and that cheating will lead to an academic referral. We are aware of the identified behaviors that are considered violations of the academic standards for Undergraduate and Graduate students per USF policy."