

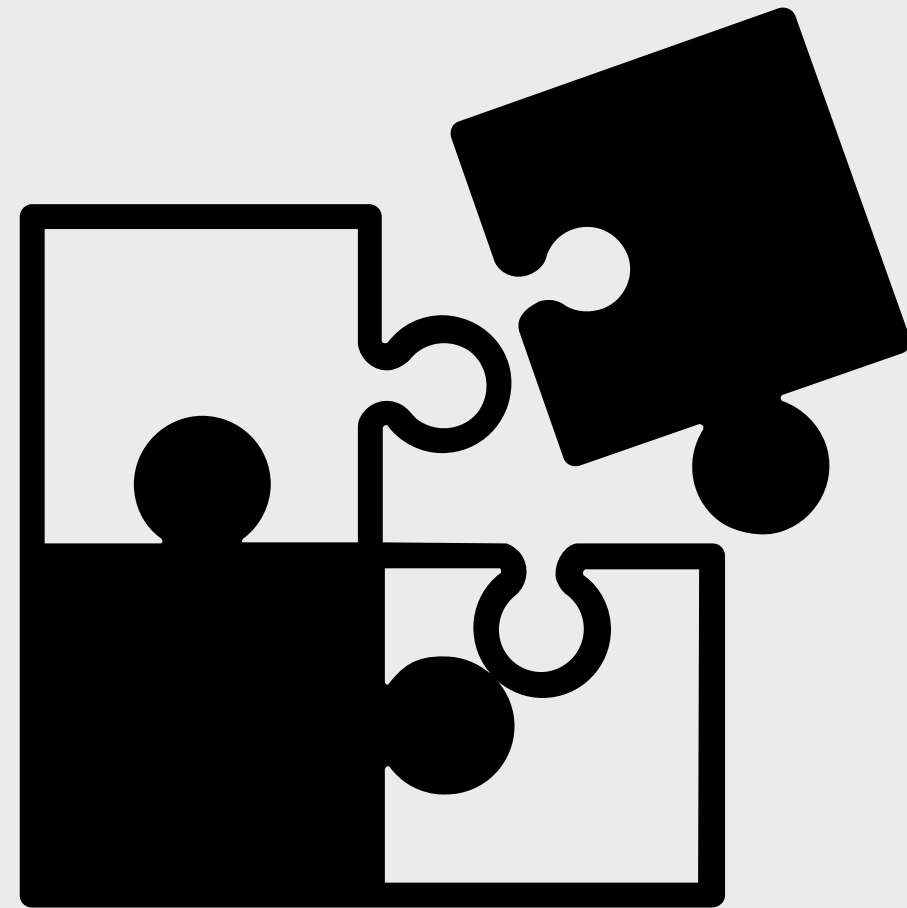
Airport

2MIEIC06_G26:

Vinícius Corrêa - up202001417

Tomás Pires - up202008319





Problem Description

Information management in an airline,
implementing an innovative
information management system
tailored to your needs.

Solution Description

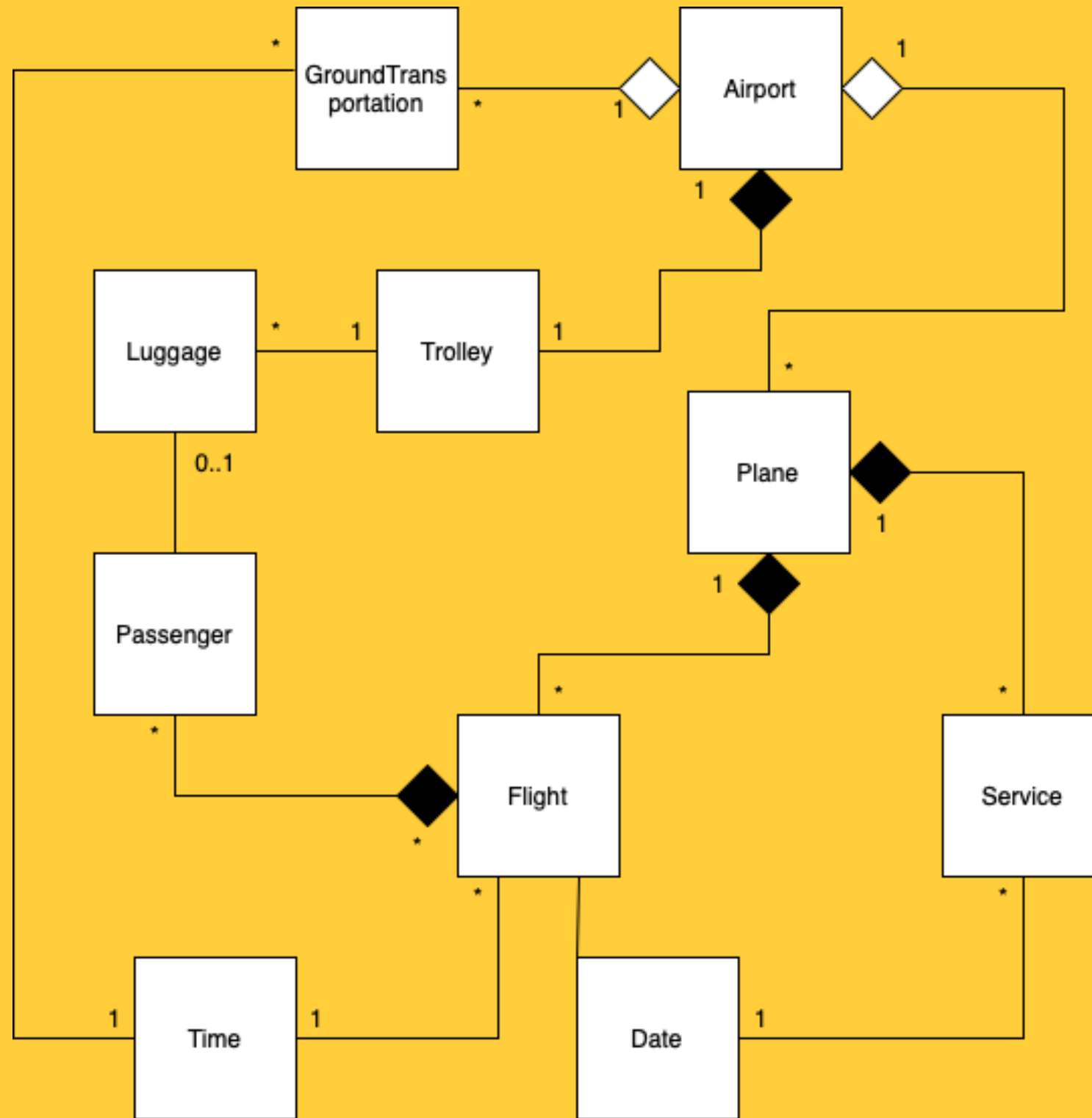
- use of **appropriate classes** to represent the entities involved.
- correct **data structure** implementation.
- use of **search** and **sort** algorithms
- **writing** and **reading** files, including the display



Relevant Algorithms

- InsertionSort
- SequentialSearch
- Sort (STL)





Class Diagram

Files Structure

airports.txt

```
Francisco Sá Carneiro Airport
Porto
OPO

Humberto Delgado Airport
Lisboa
LIS

Guangzhou Baiyun International Airport
Baiyun-Huadu
CAN

Hartsfield-Jackson Atlanta International Airport
Atlanta
ATL

Indira Gandhi International Airport
Delhi
DEL

Istanbul Airport
Arnavutköy
IST
```

transportations.txt

```
metro
0.1
10:12:00

autocarro
0.73
10:21:00

comboio
1.34
10:03:00

autocarro
0.73
11:09:00

metro
0.1
10:23:00

autocarro
1.4
10:15:00

autocarro
```

```
Database::~Database() {
    for(auto & airport : airports)
        delete airport.second;
}

void Database::loadAirports() {
    std::fstream airportFile;
    airportFile.open(AIRPORT_FILE, ios::in);
    while (!airportFile.eof()) {
        std::string name, city, code;
        std::getline(& airportFile, & name);
        std::getline(& airportFile, & city);
        std::getline(& airportFile, & code);

        airports.insert( x: std::pair<std::string, Airport*>{code, u2: n
        std::getline(& airportFile, & name); //dummy read
    }
}

unordered_map<std::string, Airport*> Database::getAirports() {
    return airports;
}
```

READ Stream
from airports.txt

List of Implemented Features



Create, Read, Update and Delete

- Create/Read/Update/Delete Flights:
 - Plane
- Read information of files:
 - Database
 - Airport
- Create/Delete planes:
 - Airport
 - Menu
- Create/Delete/Read Services:
 - Plane

Top Feature

Menu:

```
while(true) {
    std::cout << "Available commands:\n";
    std::cout << "[1] Display all entries\n";
    std::cout << "[2] Search for plane with specific license\n";
    std::cout << "[3] Display planes of a specific type\n";
    int choice = inputHandler<int>( msg: "Input your option : ");

    std::string filter;
    Plane *desiredPlane;

    switch(choice){
        case 1:
            displayTable( planes: currentAirport->getPlanes());
            return;
        case 2:
            filter = inputHandler<std::string>
                ( msg: "Please enter the license number of the plane : ");
            desiredPlane = currentAirport->findPlaneWithLicense(filter);
            if(desiredPlane)
                displayTable( planes: vector<Plane*>( il: {desiredPlane}));
            else
                std::cout << "Plane not found.\n";
            return;
    }
}
```

```
void Menu::displayTable(const vector<Plane *> planes) const {
    std::cout << "┌───────────────────────────────────────────────────────────────────────────────────┐" << std::endl;
    std::cout << "│ license number │ type │ seat capacity │ scheduled flights │" << std::endl;
    std::cout << "└───────────────────────────────────────────────────────────────────────────────────┘" << std::endl;
    std::cout << setiosflags(std::ios::left);
    for(Plane* p : planes){
        std::string license = formatEntry(p->getLicense(), PLANETABLE_LICENSE_WIDTH);
        std::string type = formatEntry(p->getType(), PLANETABLE_TYPE_WIDTH);
        std::string capacity = formatEntry( entry: to_string(p->getCapacity()), PLANETABLE_CAPACITY_WIDTH);
        std::string nFlights = formatEntry( entry: to_string(p->getNumberOfFlights()), PLANETABLE_NFLIGHTS_WIDTH);

        std::cout << "│" << setw(PLANETABLE_LICENSE_WIDTH) << license;
        std::cout << "│" << setw(PLANETABLE_TYPE_WIDTH) << type;
        std::cout << "│" << setw(PLANETABLE_CAPACITY_WIDTH) << capacity;
        std::cout << "│" << setw(PLANETABLE_NFLIGHTS_WIDTH) << nFlights << "│\n";
        std::cout << "└───────────────────────────────────────────────────────────────────────────────────┘" << std::endl;
    }
    std::cout << "┌───────────────────────────────────────────────────────────────────────────────────┐" << std::endl;
    std::cout << setw( n: 0);
}
```


Main Difficulties

- Adding luggage to the transport car, maybe too abstract to visualize.
- Sorting by user filter.
- Developing the project by 2 people instead of 3.
- Menu had to extract a lot of unknown knowledge.

```
void Trolley::luggageThroughTreadmill(const Luggage &l1) {...}  
  
void Trolley::insertLuggageIntoTrolley() {...}  
  
bool Trolley::trolleyIsFull() const {...}
```

```
Welcome to the VRG Airline Management System.
Please enter the codename of the desired airport, or 'q' to quit to the command line.

Currently available airports:

Aeroporto de Lisboa (Lisboa) : lisbon
Aeroporto Sá Carneiro (Porto) : scarneiro
```

1

```
[3] SERVICE SCHEDULING SYSTEM
[4] CLIENT MANAGEMENT SYSTEM
[5] BAGGAGE AUTOMATIC CHECK-IN SERVICE MANAGEMENT
[6] LOCAL TRANSPORT SEARCH SYSTEM MANAGEMENT

[q] Go back

1
[1] Add new aircraft into airport's database
[2] Display current airport's aircraft database.
[3] Delete aircraft from current database

[q] Go back

1
Please enter the plane's license number : N2431
Please enter the plane's type code : A310
Please enter the plane's seat capacity : 220
New aircraft successfully added to database.

[1] Add new aircraft into airport's database
[2] Display current airport's aircraft database.
[3] Delete aircraft from current database

[q] Go back
```

2

```
[1] Add new aircraft into airport's database
[2] Display current airport's aircraft database.
[3] Delete aircraft from current database

[q] Go back

2
Available commands:
[1] Display all entries
[2] Search for plane with specific license
[3] Display planes of a specific type
Input your option : 1
```

license number	type	seat capacity	scheduled flights
N2431	A310	220	0
N2341	A380	575	0

```
[1] Add new aircraft into airport's database
[2] Display current airport's aircraft database.
[3] Delete aircraft from current database

[q] Go back
```

3

Execution Example

- 1 Main menu / Selection Airport
- 2 Register of Planes
- 3 Specifications

Effort of Each Member of The Group



Vinícius Corrêa
50%



Tomás Pires
50%