

**UNITED INTERNATIONAL UNIVERSITY (UIU)**

Dept. of Computer Science & Engineering

Trimester: Summer 2022

Course No: CSE 4495 Title: Software Quality Assurance and Testing

Section: A

Assignment - 1

There is a single problem in this assignment. Discuss with your teammates and submit one file (per group) in eLMS containing the solution

**Cover Page:** On the cover page of your assignment, include the name of the course, the date, your group name, and a list of your group members.

**Problem Description:**

The airport connection check is a high-level function exposed by the API of a travel reservation system. It is intended to check the validity of a single connection between two flights in an itinerary.

For example, a traveler may intend to fly from Gothenburg to Los Angeles, but there is a connection through Frankfurt. Therefore, their itinerary is Gothenburg -> Frankfurt (Flight A) and Frankfurt -> Los Angeles (Flight B).

This service will ensure that the connection through Frankfurt is a valid one. For example, if the arrival airport of Flight A differs from the departure airport of Flight B, the connection is invalid. That is, if we pass in two flights, and Flight A arrives in Frankfurt, but Flight B departs from Munich, it is not a valid connection.

Likewise, if the departure time of Flight B is too close to the arrival time of Flight A, the connection is invalid. If Flight A arrives in Frankfurt at 8:00, and Flight B departs at 8:05, there is not sufficient time to complete the customs process and board the flight.

**validConnection(Flight flightA, Flight flightB)**

**returns ValidityCode**

A **Flight** is a data structure consisting of:

* A unique identifying flight code (string, three characters followed by four numbers).
* The originating airport code (three character string).
* The scheduled departure time from the originating airport (in universal time).
* The destination airport code (three character string).
* The scheduled arrival time at the destination airport (in universal time).

There is also a **flight database**, where each record contains:

* Three-letter airport code (three character string).
* Airport country (string).
  + If the country is in the Schengen Area, this is indicated instead of the home country.
* Minimum domestic connection time (integer, minimum number of minutes that must be allowed for flight connections to be valid).
* Minimum international connection time (more time is required due to need to clear customs and meet regulations)

Note that connection times are calculated based on the originating airport and destination airport of Flight A. If the country of Flight A’s originating and destination airports differ, then the international connection time is required. For example, if you flew from Los Angeles to Frankfurt (USA -> Schengen Area), then the traveler would need to clear customs before boarding flight

B. This requires a longer connection time. If Flight A’s originating airport is in a Schengen Area

country, connections to all other Schengen Area countries are considered to be “domestic flights” and do not require completion of customs.

**ValidityCode** is an integer with value

0 for OK,

1 for invalid airport code,

2 for a connection that is too short,

3 for flights that do not connect (Flight A does not land in the same location that Flight B departs from), or

4 for any other errors (malformed input or any other unexpected errors).

**Design system test cases using the category-partition method for the validConnection**

**function.**

1. **Identify choices (aspects that you control and that can vary the outcome) for the two input flights and the database.**
2. **For each choice, identify a set of representative values.**
3. **Apply ERROR, SINGLE, and IF constraints.**
   1. **ERROR = This representative value will trigger an error no matter that it is paired with.**
   2. **SINGLE = This representative value should give an OK response, but we want to make sure we try it once.**
   3. **IF = This representative value can only be used if a certain value is set for another choice.**
4. **Determine the number of test specifications before and after applying the constraints.**