Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

Fundamentos de Programação, 2018/2019

Teacher: Rui Oliveira

Implementation Report

Group 18

fc44605 Cláudia Garcia Belém

1. Project resolution details

To develop this project, I have used the Mac OS (OS 10.14 Mojave) operative system.

During the implementation of the code, the methods/functions names were written according to the camelCase convention whereas the variable names were written according to the underscore (or underline) convention in order to avoid confusion and to facilitate reading the code.

The code was implemented in its whole by the student.

2. Additional functions and their functioning

The following functions were implemented in addition to the functions given by the teacher:

2.1- *dateTime* Module:

getDateFromString - *Converts a date string into integer variables representing the year, month and day.*

dateToString - *Converts a date_tuple to a string.*

getTimeFromString - *Converts a time string into integer variables representing the hours and minutes.*

timeToString - *Converts a time_tuple to a string.*

addPeriodToTime - *Adds period_in_hours and period_in_minutes to time*

addDaysToDate - *Adds a number of days to a date.*

intDateTimeToString - *Receives an integer value and converts it into string.*

selectMostRecentDateTime - *Compares two date/time pairs and retrieves the most recent pair.*

selectMostRecentTime - *Compare two times and retrieves the most recent.*

updateDateTime - *Adds period_in_hour and period_in_minutes to time_str and updates the date_str if necessary.*

2.2- *filesReading* Module:

convertExpertToDictionary - *Convert a string that corresponds to an expert into a dictionary.*

extractExpertsSpecialities - *Converts a speciality_string_tuple tuple into a tuple of strings.*

readClientsFile - *Converts a given file listing clients into a collection*.

convertClientToDictionary - *Convert a string that corresponds to a client into a dictionary.*

2.3- *filesWriting* Module:

createFileName - *Creates a file name based on the information contained in a tuple.*

updateHeader - *Updates the header information present in a header_tuple.*

writeHeader - *Creates a header containing the day, time, company and scope of a file.*

createDirectory - *Receives the file name of a file present in the folder where the output files will be saved and returns the directory of the file.*

writeExpertsFile - *Creates a file and write the experts in it.*

convertExpertsDictToString - *Converts an expert dictionary into string.*

writeSchedule - *Creates a file and write the schedule for each client in it.*

convertScheduleDictToString - *Converts an expert dictionary into string.*

2.4 – *scheduling* Module:

Schedule - *Matches the clients in a clients_list with the experts in an experts_list.*

expertSorting - *Sorts the experts_list according to the date, time and name.*

expertIsEligible - *Receives a client and an expert and verifies is the expert is eligible to perform the requested work by the client.*

assignClient - *Assigns an expert to a client.*

addExpertTravelTime - *Add the 1h travel time to the expert availability time.*

declineClient - *Converts the request to declined.*

updateExpert - *Updates the expert date and time of last request and the total_money gathered.*

calculateExpertTotalMoney - *Calculates the expert total money after completing a service / fulfilling a client request.*

finalSorting - *Sorts the schedule_list and experts_list according to date, time and name of client and expert, respectively.*

2.5 – *cyberConc* Module:

checkConsistencyHeaderFileName - *Checks if the file_path corresponds to the file header.*

checkConsistencyBetweenFiles - *Checks if there are no inconsistencies between the date, time and company of two headers.*

3. Absent functionalities

No functionalities were left to implement.

4. Known errors

No errors were detected throughout this project.