POLITECNICO
MILANO 1863

Automation and Control Engineering

Testing Document

Project of Software Engineering (for Automation)
Professors: Rossi Matteo Giovanni, Lestingi Livia

SchedulEx

26/07/2023

Authors:
Petulicchio Lorenzo (10639923) – Talacci Mattia (10647582)

Sommario

1 - INTRODUCTION

The following documentation helps to explain the different test benchmarks applied to the SchedulEx software.

For each of the most important components of the software, test sets have been developed to

to ensure their robustness to possible erroneous user input and to ensure that the results of the system are nevertheless correct.

The tests were performed at the end of the writing of the code in order to validate it and make any changes if problems arose.

It is important to note the independence of the validity of the testing from the chosen programming language and the fact that it cannot guarantee the absence of errors, but allows the presence of errors to be detected.

2 - BLACK BOX TEST

The goal of black box testing is to test what the program is supposed to do and hence, the test cases are derived from the software specifications. In the following examples the main test sets of the SchedulEx application are checked, each one with different test cases in order to see not only the outputs when standard inputs are applied but also when the preconditions are violated. In this way, the robustness of the code is guaranteed and specifically in SchedulEx application this property is fundamental since it is characterized by many input user commands. In this case, we will consider programs as functional software so described by an input-output relationship as follow:

- Takes some values in input.
- Performs computations.
- Returns some value in output.

ProblemSession status management

Software tested

SchedulEx – **Select Page view**

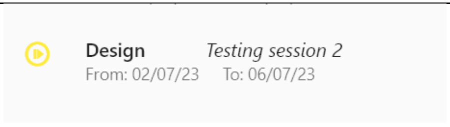
Version

Latest

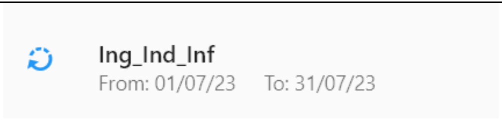
Goal

Check how the main selection page alerts the user to the evolved status of each session, in an easy and comprehensive way.

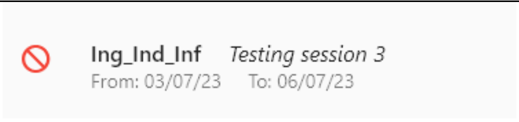
TestCase1. Problem session scheduling not started - Status set to 'NOT STARTED'

Goal	To show intuitively through simple graphics the status of optimization when the problem session is not started yet
Input	ProblemSession Instance with {'status': NOT STARTED} because user hasn't started the scheduling optimization process
Expected output	Status is set to 'NOT STARTED'
Actual output	 Design Testing session 2 From: 02/07/23 To: 06/07/23
Results	The code set the icon accordingly to the status of the ProblemSession

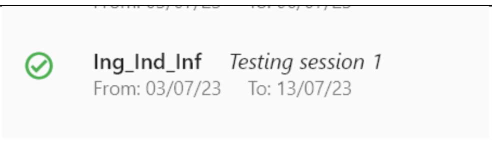
TestCase2. **Problem session scheduling started – Status set to 'STARTED'**

Goal	To show intuitively through simple graphics the status of optimization is started, and the scheduling optimization process is ongoing
Input	ProblemSession Instance for which user start the optimization process
Expected output	Status is set "STARTED" and a graphic must be visualized that suggests that the process is ongoing
Actual output	 Ing_Ind_Inf From: 01/07/23 To: 31/07/23
Results	The code set the icon accordingly to the status of the ProblemSession

TestCase3. **Problem session scheduling is infeasible – Status set to 'NOT SOLVED'**

Goal	To show intuitively through simple graphics the status of optimization is started, and the scheduling optimization hasn't produced any results, cause the problem is infeasible
Input	Problem Session Instance that brings to an infeasible problem
Expected output	Status is set "NOT SOLVED" and a graphic must be visualized that suggests that the problem couldn't be solved
Actual output	 Ing_Ind_Inf *Testing session 3* From: 03/07/23 To: 06/07/23
Results	The code set the icon accordingly to the status of the ProblemSession

TestCase4. **Problem session scheduling is feasible – Status set to 'SOLVED'**

Goal	To show intuitively through simple graphics the status of optimization
Input	ProblemSession Instance with {'status': NOT STARTED} with a feasible scheduling
Expected output	The problem session obtained an optimal scheduling
Actual output	 Ing_Ind_Inf *Testing session 1* From: 03/07/23 To: 13/07/23
Results	The code set the icon accordingly to the status of the ProblemSession

ProblemSession creation and modification

Software tested

SchedulEx – Session Page view

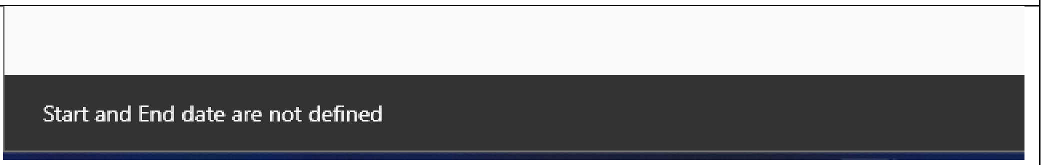
Version

Latest

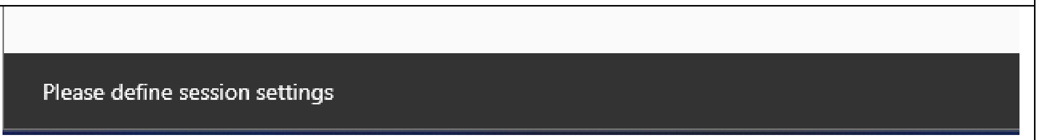
Goal

To show how the Session page code react to starting of the scheduling process when the user doesn't specify some essential parameter.

TestCase1. Session date range is not selected – start

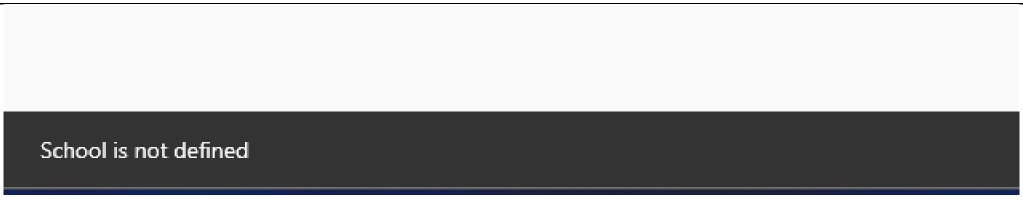
Goal	To show that the code doesn't allow starting of a scheduling for a Problem Session where the dates are missing
Input	A Problem Session Instance where the 'startDate' and 'endDate' values are missing
Expected output	The scheduling process doesn't start and a warning message is visualized: "Start and End date are not defined"
Actual output	 Start and End date are not defined
Results	The code is warning the user that the start and end dates are missing and avoid the start of optimization

TestCase2. Session settings is not selected – start

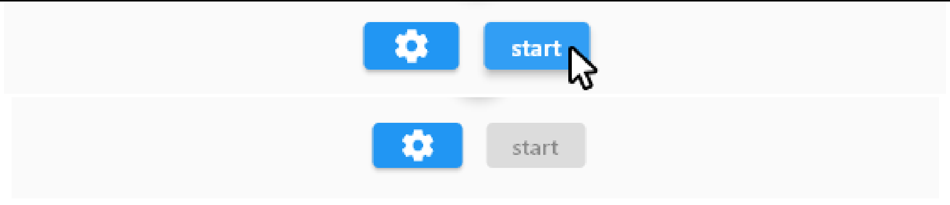
Goal	To show that the code doesn't allow starting of a scheduling for a Problem Session where the session settings are missing
Input	A Problem Session Instance where any of the session settings to be specified are missing
Expected output	The scheduling process doesn't start and a warning message is visualized: "Please define session settings"
Actual output	 Please define session settings
Results	The code is warning the user that the start and end dates are missing and avoid the start of optimization

TestCase3. Session school is not selected – start

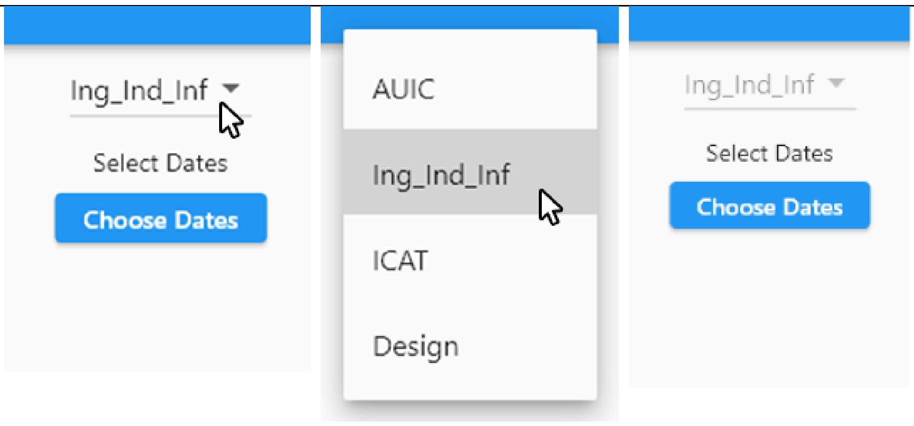
Goal	To show that the code doesn't allow starting of a scheduling for a Problem Session where the school for which is starting the scheduling is missing
Input	A Problem Session Instance where the 'school' value is missing
Expected output	The scheduling process doesn't start and a warning message is visualized: "School is not defined"

Actual output	 School is not defined
Results	The code is warning the user that the start and end dates are missing and avoid the start of optimization

TestCase4. After the session problem is started, start button can't be pressed

Goal	To show that the code doesn't allow starting of a scheduling for a Problem Session that has already been started by a user
Input	The user clicks on start button in a ProblemSession that has already been started
Expected output	The button should be disabled
Actual output	
Results	The code doesn't allow to start again the same Problem Session

TestCase5. Session school is selected and not editable

Goal	Show that school once that is selected, since some of the parameters depends on school (professors, exams), must be edited only once
Input	The user clicks on the element to select the school of a problem session in which this parameter has already been selected
Expected output	The button should be disabled to further changes
Actual output	 Ing_Ind_Inf Select Dates Choose Dates AUIC Ing_Ind_Inf ICAT Design Ing_Ind_Inf Select Dates Choose Dates
Results	The code doesn't allow the user to change the value after the first selection

Adding unavailability Dates

Software tested

SchedulEx – **Unavail Page view**

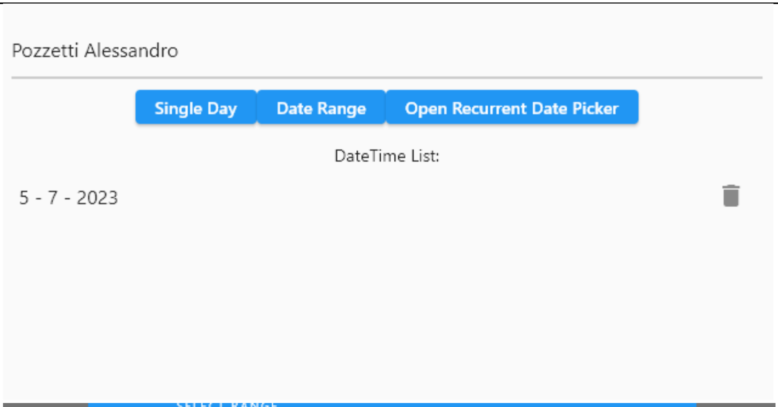
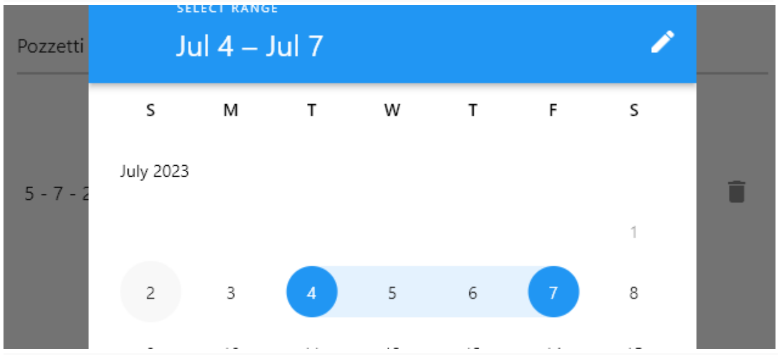
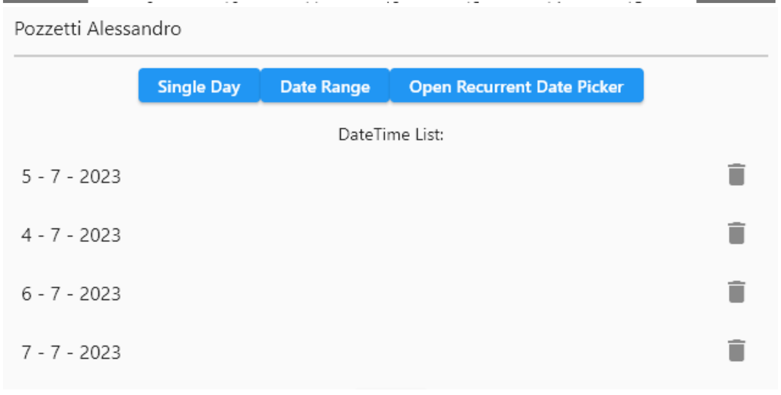
Version

Latest

Goal

To show how the Unavail Page code react to the user that is entering a day that is already entered

TestCase1. Adding a date Range in which is included a day already specified

Goal	To show that if the user is entering a date Range in which a day that is already entered is included, this is not added
Input	DateRange from 4/7/23 to 7/7/23 and 5/7/23 already included
Expected output	The dates included are not repeated
Actual output	 Pozzetti Alessandro Single Day Date Range Open Recurrent Date Picker DateTime List: 5 - 7 - 2023  SELECT RANGE Jul 4 – Jul 7 S M T W T F S July 2023 1 2 3 4 5 6 7 8  Pozzetti Alessandro Single Day Date Range Open Recurrent Date Picker DateTime List: 5 - 7 - 2023 4 - 7 - 2023 6 - 7 - 2023 7 - 7 - 2023
Results	the code is robust to redundant user entries

Results visualization

Software tested

SchedulEx – Calendar Page view

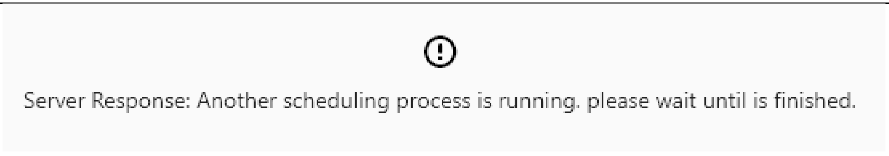
Version

Latest

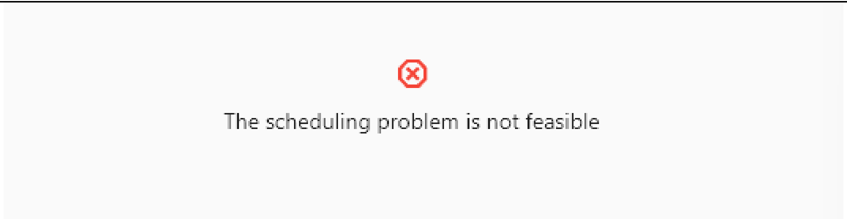
Goal

To show how the Calendar Page show the message of the scheduling process result.

TestCase1. Another problem session started while another is still going

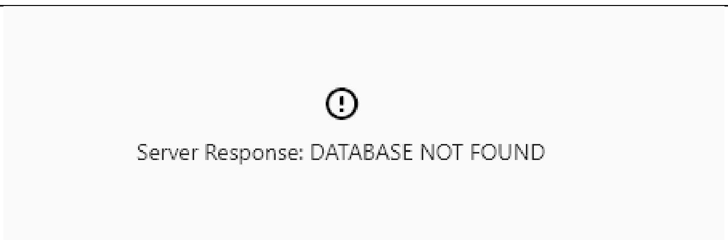
Goal	To show that the user can't have two optimization processes running at the same time
Input	The user starts the optimization process while another process is still ongoing
Expected output	A message is visualized: "Another scheduling process is running"
Actual output	 Server Response: Another scheduling process is running. please wait until is finished.
Results	The code doesn't allow to start another process avoiding the computational overload of the backend server

TestCase2. Scheduling optimization problem is not feasible

Goal	How the code react to a not feasible scheduling process
Input	A problem session Instance that brings to a not feasible solution
Expected output	A message is visualized: "The scheduling problem is not feasible"
Actual output	 The scheduling problem is not feasible
Results	the code intuitively notifies the user of the negative result concerning the session that has started

TestCase3. Exam database not found

Goal	How the code reacts when the corresponding school database was not found
Input	A problem session Instance but referring to a school that has not data accessible from the backend
Expected output	A message is visualized: "DATABASE NOT FOUND"

Actual output	 Server Response: DATABASE NOT FOUND
Results	the code intuitively notifies the user of the problem concerning the session that has started

3 - WHITE BOX TEST

The white box test performed is related to the Optimization_Manager.py script since it is the core of the process management.

Test sets applied in the white-box testing will be identified based on the following criteria: Statement Coverage, Edge Coverage, Condition Coverage, Path Coverage

Then, in each test for every coverage criterion, a coverage-based test evaluation will be performed to

assess the test validity:

$Percentage\ of\ coverage = (Number\ of\ items\ covered / Total\ number\ of\ items) \times 100$

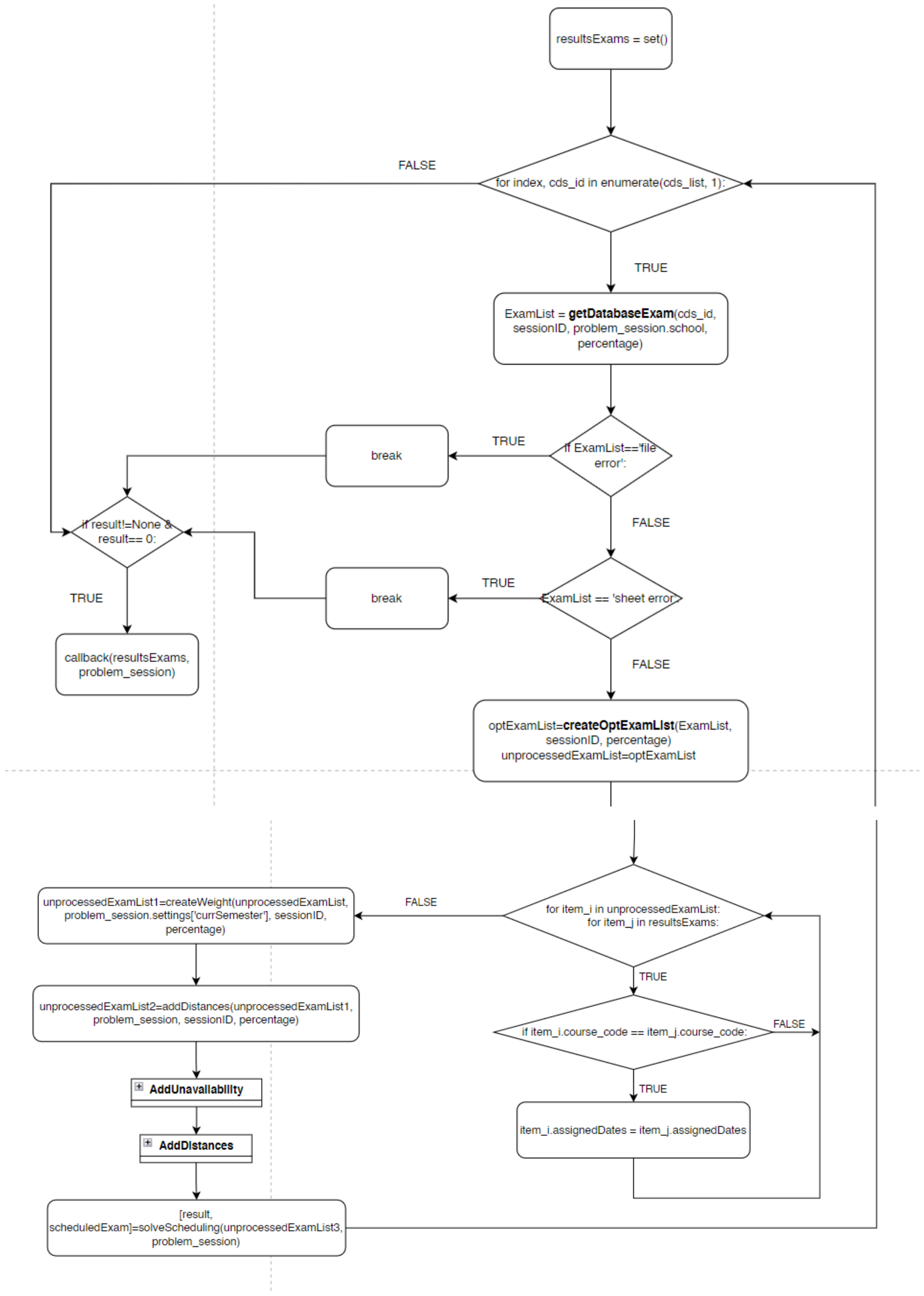
Optimization_Manager.py

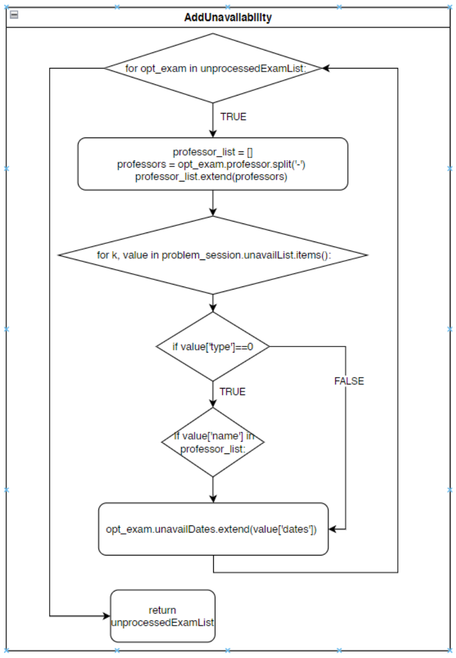
This script retrieves problem and exam data from the Firebase database and an Excel file, respectively. It then creates a list of "optExam" objects for optimization, calculates weights and distances, and incorporates unavailability data. Finally, it performs an optimization process using the "solveScheduling" function and updates the status and results accordingly. The overall process aims to schedule exams efficiently and handle unavailability constraints.

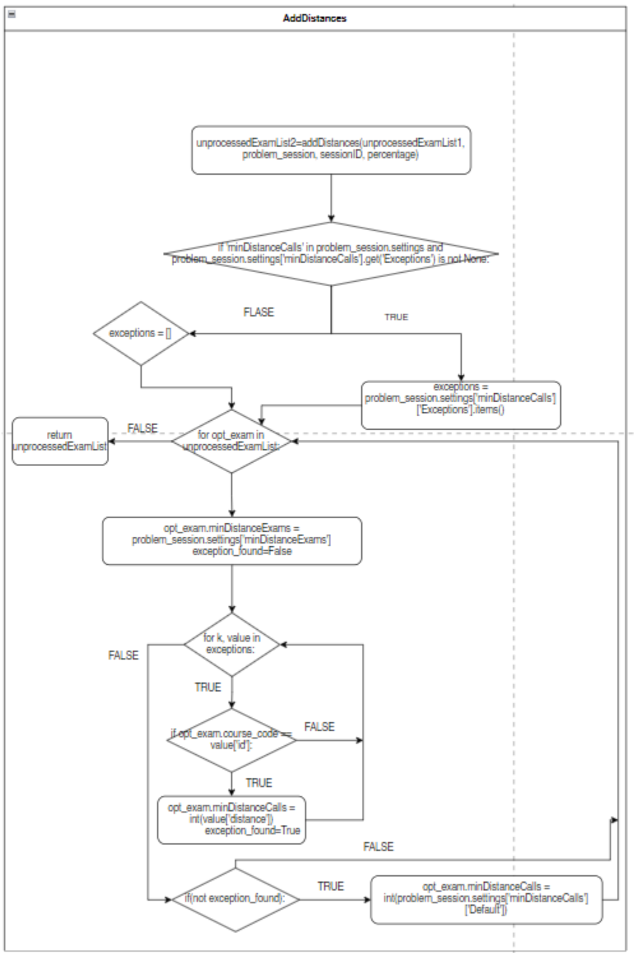
The tested software is Optmization_Manager.py, the version of the tested software is the latest and the goal is to identify is there are some errors in the code that don't let to realize the designed process.

Control Flow Graph

Here the control flow graph of the process contained in the script is reported.


resultsExams = set()
for index, cds_id in enumerate(cds_list, 1):
FALSE
TRUE
ExamList = **getDatabaseExam**(cds_id, sessionID, problem_session.school, percentage)
if ExamList=='file error':
TRUE
break
FALSE
if result!=None & result== 0:
TRUE
callback(resultsExams, problem_session)
ExamList == 'sheet error':
TRUE
break
FALSE
optExamList=**createOptExamList**(ExamList, sessionID, percentage)
unprocessedExamList=optExamList
for item_i in unprocessedExamList:
for item_j in resultsExams:
FALSE
unprocessedExamList1=createWeight(unprocessedExamList, problem_session.settings['currSemester'], sessionID, percentage)
TRUE
if item_i.course_code == item_j.course_code:
FALSE
TRUE
item_i.assignedDates = item_j.assignedDates
unprocessedExamList2=addDistances(unprocessedExamList1, problem_session, sessionID, percentage)
AddUnavailability
AddDistances
[result, scheduledExam]=solveScheduling(unprocessedExamList3, problem_session)


AddUnavailability
for opt_exam in unprocessedExamList:
TRUE
professor_list = []
professors = opt_exam.professor.split('-')
professor_list.extend(professors)
for k, value in problem_session.unavailList.items():
if value['type']==0
TRUE
FALSE
if value['name'] in professor_list:
opt_exam.unavailDates.extend(value['dates'])
return unprocessedExamList


AddDistances
unprocessedExamList2=addDistances(unprocessedExamList1, problem_session, sessionID, percentage)
if 'minDistanceCalls' in problem_session.settings and problem_session.settings['minDistanceCalls'].get('Exceptions') is not None:
FLASE
TRUE
exceptions = []
exceptions = problem_session.settings['minDistanceCalls']['Exceptions'].items()
return unprocessedExamList
FALSE
for opt_exam in unprocessedExamList:
opt_exam.minDistanceExams = problem_session.settings['minDistanceExams'] exception_found=False
FALSE
for k, value in exceptions:
TRUE
if opt_exam.course_code == value['id']:
FALSE
TRUE
opt_exam.minDistanceCalls = int(value['distance']) exception_found=True
FALSE
TRUE
if(not exception_found):
opt_exam.minDistanceCalls = int(problem_session.settings['minDistanceCalls']['Default'])

Statement Coverage

In the test set are reported only the values that influence the process flow.

Test set:

{<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam not present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study not present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id="")="", minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-04, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=1, name=2.0.1, dates)= 2023-07-03, Database exam present, sheet for each course of study present >

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type="", name="", dates)= "", Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=1, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=1, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>}

Whit this set of tests all the statements are successfully covered.

Total Number of statements: 35

Covered statements: 35

Percentage of coverage (on the edges) =100%

Edge Coverage

In the test set are reported only the values that influence the process flow.

Test set:

{<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam not present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study not present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id="")="", minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-04, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=1, name=2.0.1, dates)= 2023-07-03, Database exam present, sheet for each course of study present >

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type="", name="", dates)= "", Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=1, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=1, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>}

All the edges are successfully covered with the provided test set.

Total Number of edges: 48

Covered edges: 48

Percentage of coverage (on the edges) =100%

Condition Coverage

In the test set are reported only the values that influence the process flow.

Test set:

{<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam not present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study not present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id="")="", minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-04, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=1, name=2.0.1, dates)= 2023-07-03, Database exam present, sheet for each course of study present >

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type="", name="", dates)= "", Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=1, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=1, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>}

All the conditions are successfully covered with the provided test set.

Total Number of conditions: 30

Covered edges: 30

Percentage of coverage (on the conditions) =100%

Path Coverage

In the test set are reported only the values that influence the process flow.

Test set:

{<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam not present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study not present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id="")="", minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-04, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=1, name=2.0.1, dates)= 2023-07-03, Database exam present, sheet for each course of study present >

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type="", name="", dates)= "", Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=2, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=1, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>

<startDate=2023-07-03, endDate= 2023-07-17, school= Ing_Ind_Inf, currSemester=1, minDistanceCalls(Default)=4 minDistanceCalls(exception, id=85743)=6, minDistanceExam=2, numCalls=2, unavailList(type=0, name= Nicola Gatti, dates)= 2023-07-03, Database exam present, sheet for each course of study present>}

All the paths are successfully covered with the provided test set.

Total Number of conditions: 22

Covered edges: 22

Percentage of coverage (on the conditions) =100%

Results

Looking at the testing result in a white box approach it is possible to highlight that the Optmization_Manager and so the management of the process is robust. There could be internal function not extremely robust the overall management can deal with them.

4 - CONCLUSION

All tests performed were successfully passed, therefore all requirements listed in the RASD document were met by SchedulEx.