POLITECNICO
MILANO 1863

Automation and Control Engineering

Requirements Analysis and Specification Document

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

SchedulEx

26/07/2023

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

Summary

1 - INTRODUCTION

This document constitutes the Requirement Analysis and Specification Document (RASD). Its purpose is to analyse the requirements that will lay the foundations of application services, to specify the application domain, the entities involved and their relationship, to clearly explain the objectives, the constraints and the features that are going to be implemented.

Document structure

The document is divided into 6 chapters. The first one provides an overview of the current situation to deal with, describing the job that the software will facilitate pointing out the target and explaining generally it and which are the entities.

The second chapter is about feasibility study, here the feasibility of the project is discussed looking at different aspects. Moreover, it is present a section regarding a survey for student that helps to evaluate how to define the commitment required by an exam.

Then are presented the use cases and scenarios to understand better the possible situation of usage.

The fourth chapter regards the requirements, they are listed divided by type.

The fifth one contains the class diagram and describes the whole system.

While the last one presents the logic of the optimization algorithm.

Project purpose

In the university context the exam session is the most important period for students, it has a specific range of time in which the students can take their exam and in most of the cases a specific date and time slot is assigned to each exam, so the student has to decide when to take it. The cases in which the student has a significant number of exams are many and the cases in which a student has to take exams of past semesters are common too.

To decide when an exam is scheduled a complex process is used by the registrar’s office of Politecnico and this brings inefficiency, in particular because the student feedback is not taken into consideration.

The purpose of SchedulEx is to automate the process of scheduling the exam calendar session, in the presented situation it is analysed the Politecnico di Milano as university of reference. The main idea is to have a web app that helps the university during the exam session scheduling. Using it the workflow is reduced and it prevents several revisions of the calendar that takes time that employees can dedicate to other activities and also the professors and students can plan in advance their exam session.

This project could not only help employees of the registrar's office in their job, but it aims to facilitate students with their own session planning too. This is due to the fact that different types of information regarding the exam commitment are processed, they are directly obtained from a batch of students. The project could also have secondary purposes, for example managing in the best way possible the presence of students in Politecnico facilities.

Current situation and problems

Actually, at Politecnico di Milano the registrar’s office takes care to create the exams schedule for each exam session. This job is manually and not so efficient. Indeed, it takes a long time, and it requires different reviews. The exams to schedule are split among personnel of the registrar's office, after this, the exam scheduling is put together and each one makes sure that there is no overlapping.

In this way a first draft is obtained, then it is sent to professors who reporting their requirements by requesting respective changes by a certain date beyond which changes can no longer be accepted.

At this step, the modified draft is published to the students that analyse it and notify changes to do in order to meet their requirements.

Finally, the examination session schedule is definitively published.

With this situation the requirements taken into consideration are:

- For students:
    - Exams of the same semester and same year must have two nights of distance each other.
    - The exams of the same course must have at least 14 days of distance.
- For professors:
    - They can give their unavailability for what concerns days and time slot.
- For facilities:
    - The calendar is only predictive and only during the verifying phase the room are assigned.
    - It tries to do an equidistributional weighting with the number of students for each course in order to not have more than a certain number of students simultaneously.

The logic adopted to schedule the exams is: firstly, intercourses and courses with multiple staggers are scheduled, secondly the mandatory exams and lastly are scheduled the elective courses exams.

Each exam is characterised only by a number (between 1 and 6) that identifies the semester, the identification year and semester is not used.

Semester enumeration

	January-February	June-July
1st year	1	2
2nd year	3	4
3rd year	5	6

This method brings several problems:

1. A lot of resources in terms of time is spent due to the flow adopted to obtain the final schedule.
2. The students most of the time receive a calendar that if difficult to fit with their necessity, for example the time to prepare well for an exam is not compliant with the distance between some exams.
3. Students receive the exam calendar not very early, so they will make their plan late without having the possibility to study during the semester in a very effective way.

Targets

SchedulEx aims to optimise student's experience during exam session. To do so, the following targets are taken as the benchmark for evaluating performance:

- Maximize the exam distance.
    - Creating an optimal session student can plan better when to take an exam.
- Meet the requirements of
    - Professors that could have some days of unavailability of need a specific time between two calls of their exam.

  - Students that can't have in the exam session calendar two exams with a distance of less than two nights. Moreover, two calls of the same exam must have at least fourteen days of distance.
  - University facilities (classroom in particular), because to have a good situation to take an exam there must be number large enough of sits in the classroom.
- Automating a long and labour-intensive process making it:
  - Shorter in order to give the calendar to student and professor in advance.
  - More robust to variation, because it could happen that is necessary to change something in the planning and it should be done easily.

General description

The software can be used at the beginning of each semester to be able to publish the provisional calendar at times compatible with the personal organisation of individual students regarding their exam session.

The software is designed precisely from the needs of students who are struggling with preparing multiple exams at the same time resulting in high stress levels, etc.

In addition, our software aims to automate a process that is currently done by hand to avoid human error and optimise processing time. The users involved are:

- Active users: *secretariat*
- Passive users: *students, professors* and *Politecnico.*

The system will mainly be used by the student secretariat and dedicated committee members.

For experimental purposes only, the improvement and consequent optimization of the forecast calendar for the Leonardo site for the 3I school is considered. This is the largest school; it may be extended to other schools in the future and other universities too.

Only the Automation Engineering calendar is currently being considered.

During the winter and summer exam sessions there are two calls per exam, while in fall just one. Since most engineering schools have this limit on the number of calls of each exam. However, the software can also be used for study programmes with a number of calls greater than two.

The allocation of classrooms is done later, at the time of exam scheduling, a constraint on the maximum number of students simultaneously present at the Politecnico for exams is considered. It is probable that then it is not feasible to manage the classrooms but this is not something that you can know in advance as the student's registration for the calls closes a few days before the date of the exam (this deadline is temporally after the publication of the calendar so in case of problems changes will be made by hand later as needed as in fact is still done by the student secretariat).

User characteristics

The people involved in the direct or indirect use of the software belong to three types: professors, students and Politecnico employees.

Professors are the PoliMi's professors who are teachers of a course as written in the study plan card of the course, in this category are also included assistants, tutors or other collaborators of the main professor that help him during the exam.

Students, they are all students enrolled in PoliMi, from the first year to the fifth, without distinction if they are enrolled in Bachelor Degree, Master Degree or Single Courses. Everyone who has to take at least one exam, doesn't care whether they will take an exam during the exam session or which exams they will take, and it doesn't care if they are in the course or not. Each student that can potentially take an exam in the session in which the software is utilised is in this category. For example, students who are in Erasmus or similar program and can’t take any exam are not in this category as laureandi that have finished their exams.

PoliMi employees are firstly registrar's office personnel and commission personnel.

These three classes are divided in this way.

Direct user:

- PoliMi employees (registrar's office personnel and commission personnel)

Indirect user:

- Professors
- Students

Direct involved entities

- **Registrar's office employees:** they are the ones that will directly interact with the software, in particular with the GUI because they insert input data, they make the software perform the scheduling and they receive the output (calendar).
- **Professors of commission:** they communicate independently with their colleagues to gather information about unavailability of each professor of each course. Actually, it is required that professors need to communicate each own unavailability, otherwise it will not consider in scheduling.

Indirect involved entities

- **Students**: they receive the software output (a calendar with all scheduled exams). They can find the calendar on Polimi website, and their role is to decide which exams take and in which call take it.
- **Professors**: they have to communicate their unavailability in terms of days or time slot or request about cable classroom necessity to registrar’s office via email or form.
- **Classroom reference**: they are informed regarding when classroom is occupied.

2 - FEASIBILITY STUDY

The feasibility study helps to decide objectively whether to proceed with the development of SchedulEx. Aspects such as technological and technical limitations, market, marketing strategy, personnel requirements, timeframe, and financial projections will be presented. At the end it is discussed the results of a survey took to understand what the student think about the optimal exam calendar scheduling.

Technological Feasibility

All the technologies (better explained in the Design Document) are available on the market and largely diffused so the project is feasible in terms of realization and maintenance. The knowledge needed to realize the project are notions from different fields: web app, database, back-end and optimization algorithm. But being very used nowadays it is easy for other people to develop the software.

The critical point lies in the optimization algorithm because depending on the parameters such as number of exams, available days, constraints and weights parameters it is possible that a feasible solution can't be provided, so having soft constraints and changing the weight parameters can be fundamental to obtain a feasible solution. Moreover, regarding the hardware there aren't strict requirements, the only things to take care about is the ram of the processor that is necessary to solve the optimization algorithm in a reasonable time.

Technical Considerations

SchedulEx can be integrated in the calendar scheduling system of the universities, it will be used by secretary office and of the people who managed the exam session calendar. It will be easy to use with a standard procedure, so it is not necessary that these people learn new skills.

SchedulEx need a server to store the backend and run the optimization algorithm that can require a lot of time and a high number of computational resources. So, it is important for universities that want to implement SchedulEx to have this kind of IT infrastructure in place.

Another aspect to take into consideration in that the version of the software presented in this project works well with situations in which the exam session is a period free of lectures or other activities of the student, because these can bring more, or different constraints now considered now. Hence before implement SchedulEx it is important to know how the exam session of the university is.

Market and Competitors

The market for which SchedulEx is developed is composed of all universities in Italy or not, it is important to analyse the different aspects of them but with possible changes SchedulEx can fit with the university's requirements. Moreover, it is possible that SchedulEx in the future can be able to schedule other problems, for example the lectures with the relative classroom, but in any case, for the university environment.

The main competitors are EMS Software, ExamSoft, Ad Astra and CollegeNet.

- EMS provides a software solution for managing academic events, including exams. Their software offers features for space planning, resource management and student registration.
- ExamSoft specializes in exam management and offers comprehensive software for creating, administering, and grading exams. Their system allows you to create questions, schedule exams, track results, and generate reports.
- Ad Astra offers a suite of academic planning software that includes specific features for exam scheduling. Their software helps institutions optimize classroom allocation, manage schedule conflicts, and plan resources needed for exams.
- CollegeNet offers software called "25Live" that helps academic institutions plan and manage a wide range of events, including exams. Their system offers features for space planning, resource management and calendar publishing.

The main advantages that SchedulEx offers is the fact that it is design on the specific needs of the students, thanks to a survey the preferences of the students are collected, and they are used to find the optimal weights for the optimization.

Marketing

To promote software that offers optimal university exam planning, there are several marketing strategies that can be adopted to make buyers understand the possible benefits of the product. It is important to emphasize the key benefits of university exam planning software, such as reducing errors, optimizing resources, saving time, and improving the student experience. The main strategy that can be adopted is to contact directly the university, in particular the registrar's office, or who is in charge to manage the exam session calendar. For sure have the possibility to show them how the results of SchedulEx increase the quality comparing them with the current scheduling is essential. Another option could be participating in event as exhibition is which also universities are involved.

Once the customer is acquired it is important to make partnership and get in touch with its network.

Organization and Teamwork

To design, develop and sell SchedulEx a team that can cover all the aspects is needed. The professional role required are:

- Front-end developer: He must have a good knowledge of webapp frameworks and UX/UI skills to be able not only to design a good software but also to think how it can be easy to use.
- Back-end developer: He must have a good knowledge of programming languages in order to develop all the logic that manage the entire program.
- Mathematical engineer: He must be able to work with linear optimization problem, understanding how to draw it and how to increase the performance, for example in terms of computational time.
- Marketing and sales Manager: He must be able to conclude trade agreements and promote the product.

On the whole, the organisation for SchedulEx is designed to ensure that the application is developed and marketed effectively, while also providing assistance if there are changes, for example in the optimisation process.

Road Map

The development of SchedulEx is expected to last 16 months from the start of the project to market entry. Below is a calendar with the main milestones of the development cycle of the project in its entirety:

- September 2023: project kick off.
- January 2024: having a first version of the software and test starting.
- March 2024: closing of the test phase, result comparison and marketing initiatives starting.
- June 2024: contacts with universities and demo session to show the advantages.
- September 2024: partnership agreement
- December 2024: first exam session scheduling with SchedulEx

The main assumption is that at the kick off the team is already composed, at least for the programming part.

Financial Analysis

On the revenue side, the idea is to have a trial plan at an advantageous price that can be used for a single exam session by the university and a standard plan with an annual licence. Multi-year packages can also be devised, which bring financial benefits to the customer.

On the other hand, the costs to be borne are those of the personnel required for software development, marketing, sales, management and office space. No special equipment is needed other than servers for internal product testing.

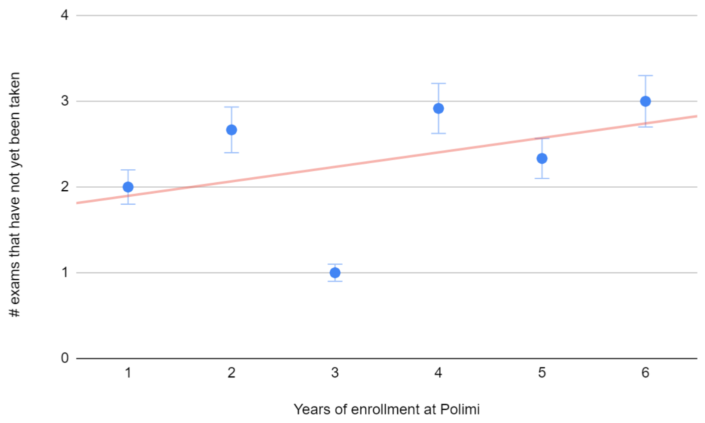
To have a concrete idea of the costs/income and not just a qualitative analysis the project can be studied deeply but for the project presented it is not taken into consideration a deep economic analysis.

Student Survey Responses

To study the feasibility of the optimization algorithm, in particular the aspects regarding the weights parameters a survey was provided to students, this is divided into five sections: student profile, current situation, optimal exams session, required commitment of exams, exam planning.

The majority of respondents were students from the Automation and Control Engineering department (56%), with others mainly from the 3I Engineering department. In the **first section** of the survey, the goal was to gather information about the current situation and students' perceptions of the problem. The survey asked participants to indicate the number of exams they have not yet taken, plotted against their years of enrolment at PoliMi. As expected, this number increased linearly with the years of enrolment, reflecting the increasing number of exams to be completed. However, we realised that this question was not well formulated, as there was no distinction made between bachelor's and master's degree students.

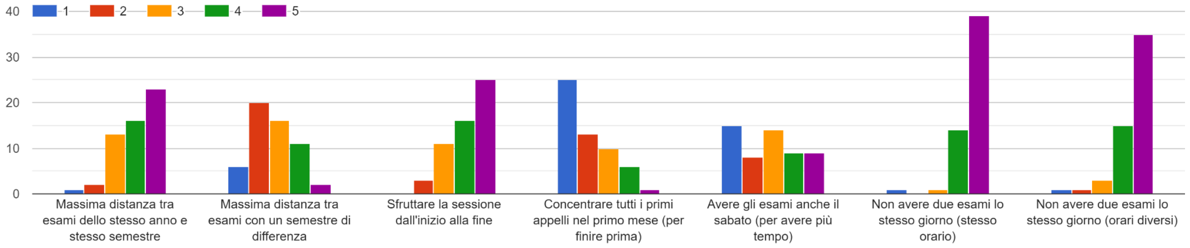
Figure 1: Correletation between exam not taken and year of enrollment



71.2% of the students rated the arrangement of exams during the session as an average of 1 or 2 (on a scale of 1 to 4), associating the following main issues:

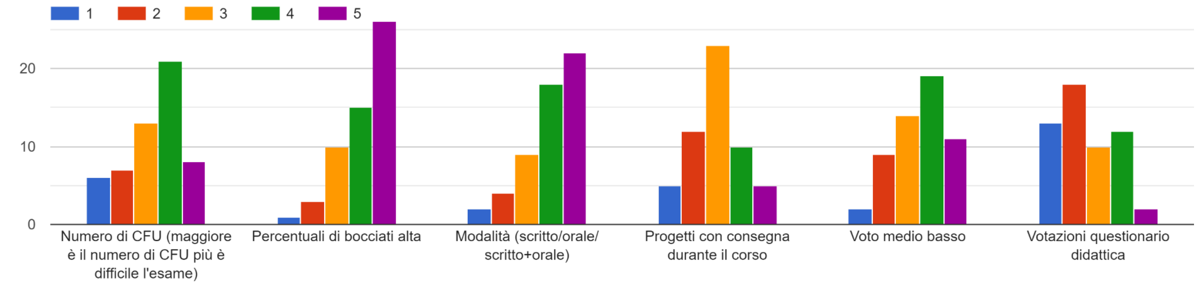
Exams too close together	70.9%
Poorly utilised session period	47.3%
First call date too close to end of lectures	41.8%
Overlapping exams	41.8%
Methods of examinations that overlap	36.4%

In the **second section**, it is asked to students to evaluate potential targets to focus our optimization algorithm on. As predicted, the most requested aspects were exam overlaps and distances between exams. Although the option "Maximizing the use of the entire exam session" was selected as very important, students did not have a clear stance on utilizing Saturdays to make the most of the exam period.



In the **third section** it tries to identify possible parameters to evaluate the difficulty of an exam and his relative required effort, in an objective way.

Through the responses in this section, it is possible to perform more targeted parameter tuning based on the available data. Indeed, “CFU” credits are not considered an objective measure indicating the actual amount of study time required for a specific exam, but at least can be a first metric for understanding the size of a course. The "high failure rate" is considered the most relevant factor for our purpose, unlike the "teaching evaluation questionnaire" and "low average grade". It would be interesting to further develop the "Examination mode" category, as it was highly selected by students, but since it lacks differentiation between options (written, oral, written/oral), it will not be taken into consideration in this initial development phase.



The survey asks how many days it took to prepare for exams categorised as "little or very effort demanding," but the results are very uneven and of little use to an implantation in this early development.

3 - USE CASES AND SCENARIOS

Scenarios

In this section are presented the possible scenarios, being the software used exclusively by the registrar's office the main use cases regard it but are listed also cases in which students or professors interact with the output of the software or with the directly involved users.

1) It is time to start scheduling the exam session for the 3i school, Professor Svelto creates a session to schedule the associated calendar, together with the members of the reference committee, Alice, Mario and Carlo. Everyone will need to collect the unavailability of professors and be able to initialize problems related to that school.
2) Professor **Svelto**, after communicating with the school dean, initializes the scheduling problem for School 3I with the session start and end dates present on the academic calendar.
3)
    a. Professor **Franchi** had a previously unscheduled commitment that does not allow him to be able to be there on the 20th of June. Since it is a possible exam day and the deadline for being able to report unavailability has not yet passed, the professor notifies the student secretary of his new unavailability to be changed. **Mario** receives Professor **Franchi**'s request to add unavailability and enters the new unavailability in the associated session.

    b. Professor **Amaldi** will hold a conference at a foreign university that does not allow him to be present from July 25-30. **Carlo** receives a request to add Professor **Amaldi**'s unavailability and places the new unavailability in the associated session.

    c. Professor **Clericetti** also teaches in a science high school and is unavailable during the session period every week on Tuesdays. **Alice** receives the request to add Professor **Clericetti**'s unavailability and enters the new recurring unavailability into the associated session.
4) On Saturday, July 8, **Politecnico** organized an open day for high school students approaching their postgraduate choice. For this reason, the Leonardo campus will not be available. **Mario** adds this unavailability to the problem session associated.
5) Professor **Giannetto** had told the student secretary in early March that he would have a commitment in the second week of July, but this commitment has been cancelled so he tells the student secretary that he is available during the period he had previously stated as unavailable. **Alice** receives the change from Professor **Giannetto** that he is available at a time when he had previously said he was unavailable and changes the professor's unavailability to the associate session. Then she add a comment, to keep track of this editing.
6) Having collected all the unavailability, and compared with colleagues, **Carlo** starts the automatic scheduling process.
7) The chair publishes new rules regarding the distance between two appeals and the number of appeals of the same exam in a single session, the student secretary makes these changes and starts a new session, to get a new schedule.
8) Having obtained a schedule, Professor **Svelto** wants to post the resulting calendar with exam-related dates on the school website for all students to view.

Use cases

Use Case 1. Create ProblemSession

Participating actors	- Secretary Users
Entry Condition	Usually, an exam session calendar must be created some months before the actual start
Flow of events	- **Secretary User** activates the "Create ProblemSession" function. - **SchedulEx** respond by presenting the GUI to insert all the needed information. - **Secretary Users,** once they logged in, can open the same page created before to add any information to it
Exit Condition	

Use Case 2. Set StartEndDate

Participating actors	- Secretary User - School Dean (*indirect user)*
Entry Condition	A ProblemSession must exist
Flow of events	- **Secretary User** activates the "Open ProblemSession" and **SchedulEx** return the GUI for the relative ProblemSession. - Here he clicks on the button and activates "set StartEndDate", is returned a calendar in which he can easily select the start and the end of the exam session. - After Prof Svelto confirmation, the selected dates are stored by **SchedulEx**
Exit Condition	An acknowledgement of the data saving that occurred is visualized

Use Case 3. Add Unavailability - Professor

Participating actors	- Secretary User - Professor (*indirect user)*
Entry Condition	A ProblemSession must exist
Flow of events	- ***Professor*** communicates to the secretary his unavailability. - **Secretary User** activates the "Open ProblemSession" that involves **Professor** and **SchedulEx** return the GUI for the relative ProblemSession. - Here he clicks on button and activates the "Add Unavailability". - **Schedulex** return a form in which **Secretary User** can defined the type (professor), the name (Franchi), and dates in which professor is unavailable. - Thanks to a GUI, **Secretary User** specifies the dates of unavailability (Single day/Period/Recurring) and add to the Unavailability form. - After **Secretary User** confirmation, the overall dates are computed and stored by **SchedulEx**
Exit Condition	An acknowledgement of the data saving that occurred is visualized

Use Case 4. Add Unavailability – PoliMi Unavailability

Participating actors	- Secretary User - Politecnico (*indirect user)*
Entry Condition	A ProblemSession must exist
Flow of events	- ***Politecnico*** communicates to the secretary that some class will not be able to use during certain dates. - **Secretary User** activates the "Open ProblemSession" that involves the rooms that will be used by the School involved and **SchedulEx** return the GUI for the relative ProblemSession.

	• Here he clicks on button and activates the "Add Unavailability". • **Schedulex** return a form in which **Secretary User** can defined the type (University), the identifier (e.g. "Lauree Polimi"), and dates in which rooms are not available. • Thanks to a GUI, **Secretary User** specifies the day of this type of unavailability, and add to the Unavailability form. • After **Secretary User** confirmation, the overall dates are computed and stored by **SchedulEx**
Exit Condition	An acknowledgement of the data saving that occurred is visualized

Use Case 5. Modify/Delete Unavailability and Add Comment

Participating actors	• Secretary User • Professor (*indirect user)*
Entry Condition	A ProblemSession must exist
Flow of events	• *Professor* communicates to the secretary his change about his unavailability. • **Secretary User** activates the "Open ProblemSession" that involves **Professor** and **SchedulEx** return the GUI for the relative ProblemSession. • Here he searches for the unavailability associated to **Professor** and select it. • **SchedulEx** return a resume of that unavailability in which **Secretary User** can performs some action to modify the input field or delete it. • **Secretary User** activates "Delete unavailability" function and **SchedulEx** perform the action. • After this modification, **Secretary User** insert his comment to keep track of this editing.
Exit Condition	An acknowledgement of the data deleting that occurred is visualized

Use Case 6. Start Optimization Process

Participating actors	• Secretary User
Entry Condition	A ProblemSession must exist and all information has been entered
Flow of events	• **Secretary User** needs to compute the overall calendar considering all information entered before. To do so, he/she activates the "Start Optimization". • **SchedulEx** start to compute the relative problem session and provide information on its status. • **Secretary User** cannot modify no more the Problem Session
Exit Condition	An acknowledgement of the starting process that occurred is visualized.

Use Case 7. Change Settings

Participating actors	• Secretary User
Entry Condition	A problem session must exist
Flow of events	• Student Secretary receives the new settings. • **Secretary User** activates the "Open ProblemSession" that involves the rooms that will be used by the School involved and **SchedulEx** return the GUI for the relative ProblemSession. • Here he clicks on button and activates the "Change Settings". • The secretary user makes the changes • After **Secretary User** confirmation, the overall dates are computed and stored by **SchedulEx.**
Exit Condition	An acknowledgement of the data saving that occurred is visualized

Use Case 8. Calendar publishing

Participating actors	- Secretary User
Entry Condition	Scheduling procession has completed
Flow of events	- **Secretary User** activates the "view calendar" that opens and display the calendar. - **Secretary User** can download a excel format of the calendar that can be publish on the PoliMi website.
Exit Condition	An acknowledgement of the fact that the calendar is online

4- LIST OF REQUIREMENTS

The goal of SchedulEx, as mentioned above, is to automate the scheduling process of the calendar, leaving users free to collect the necessary information and unavailability in the manner they prefer. The whole part of communication between direct and indirect users is not counted within the software implementation.

Functional Requirements

Here all the functional requirements are listed.

R1. SchedulEx must let the user to create a session to compute the Optimal Calendar associated to a specific School.
R2. SchedulEx must let the user to modify a session to compute the Optimal Calendar associated to a specific School.
R3. SchedulEx must let the user to delete a session to compute the Optimal Calendar associated to a specific School.
R4. SchedulEx must let the user to login and visualize all the sessions that has been created or create new one.
R5. SchedulEx must let the user to insert School name for which the session is defined.
R6. SchedulEx must let the user to *insert* exam session dates, more precisely the Start and End Date.
R7. SchedulEx must let the user to *change* exam session dates, more precisely the Start and End Date.
R8. SchedulEx must let the user to insert comments for the relative session.
R9. SchedulEx must let the user to add an unavailability.
R10. SchedulEx must let the user to delete an unavailability.
R11. SchedulEx must let the user to select unavailability type (Professor or Politecnico).
R12. SchedulEx must let the user to select professor name for the unavailability from a list or insert a string identifier for the rooms.
R13. SchedulEx must not allow to select a professor that doesn't belong to School previously selected.
R14. SchedulEx must let the user to add a single day for a specific unavailability.
R15. SchedulEx must let the user to add a multiple day period for a specific unavailability.
R16. SchedulEx must let the user to add a recurrent day of the week for a specific unavailability.
R17. SchedulEx must let the user to delete a single day for a specific unavailability.
R18. SchedulEx must let the user to delete a multiple day period for a specific unavailability.
R19. SchedulEx must let the user to delete a recurrent day of the week for a specific unavailability.
R20. SchedulEx must let the user to change the distance between exams.
R21. SchedulEx must let the user to change the distance between calls of the same exam.
R22. SchedulEx must let the user to change the distance between calls for a specific exam to be selected.
R23. SchedulEx must let the user to start the optimization even if not all professors have unavailability.
R24. SchedulEx must let the user to start the optimization only if start and end date of the exam session and the settings are present.
R25. SchedulEx must find a solution of the scheduling problem otherwise it must return a message of impossibility to schedule the problem.

External Interface Requirements

The following section will give a more detailed description, in terms of hardware, software and communication interfaces.

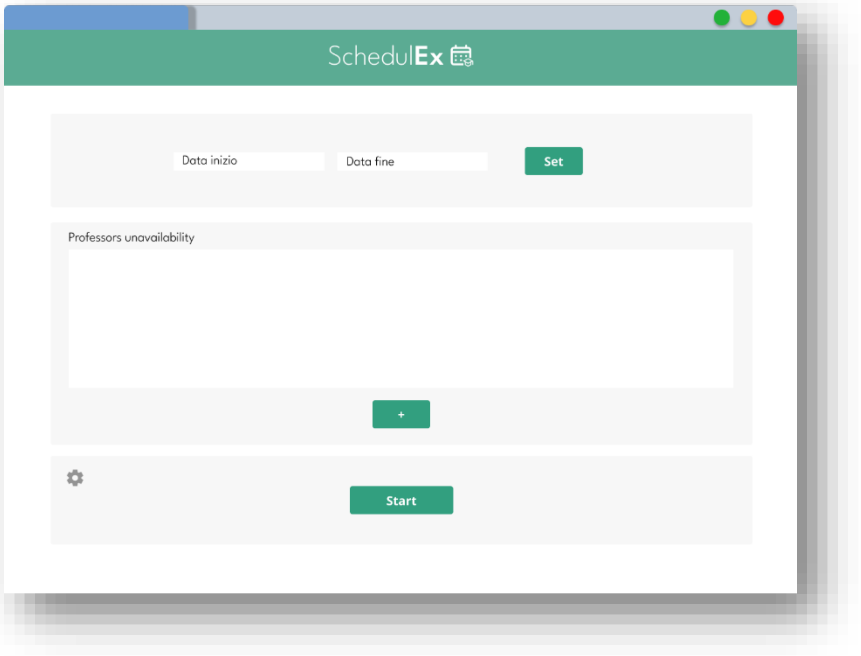
Hardware Interfaces

The hardware needed to use the software of this project is a pc with monitor (to visualise graphical user interface), keyboard and mouse to insert input data and a printer (not strictly necessary) to print the calendar. Having a workstation where lunch the optimization algorithm is necessary in order to have an acceptable execution time.

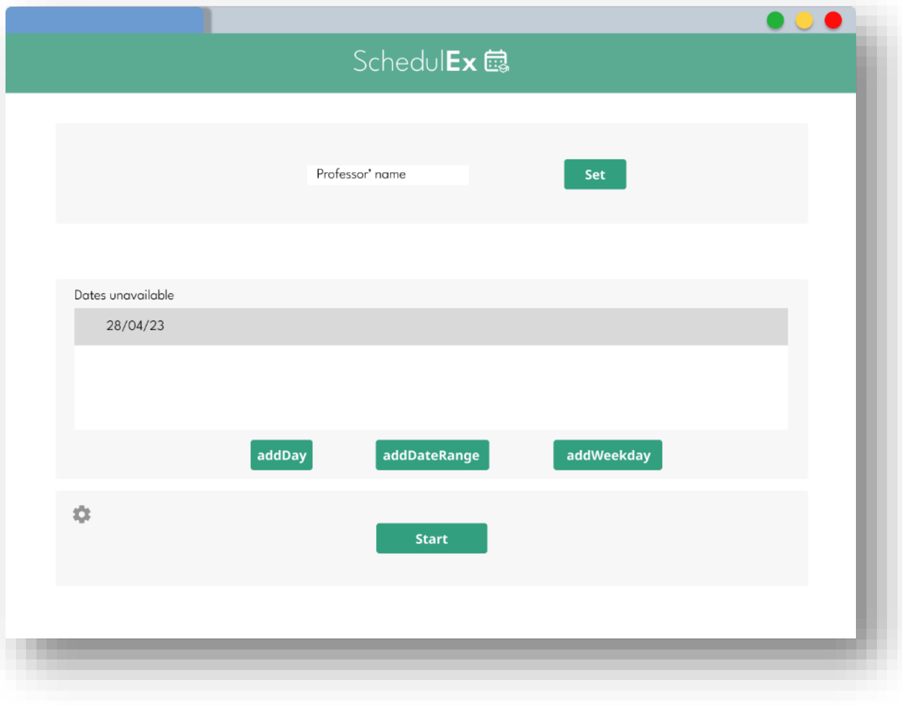
Software Interfaces

The front-end (GUI) must communicate with the back-end, this last one must communicate with the database and should launch the optimization algorithm. The back end communicates with database both for its proper request but also for front end ones. The rest of this subsection contains some mockups that show an example of suitable user interface for SchedulEx.

Session Page interface

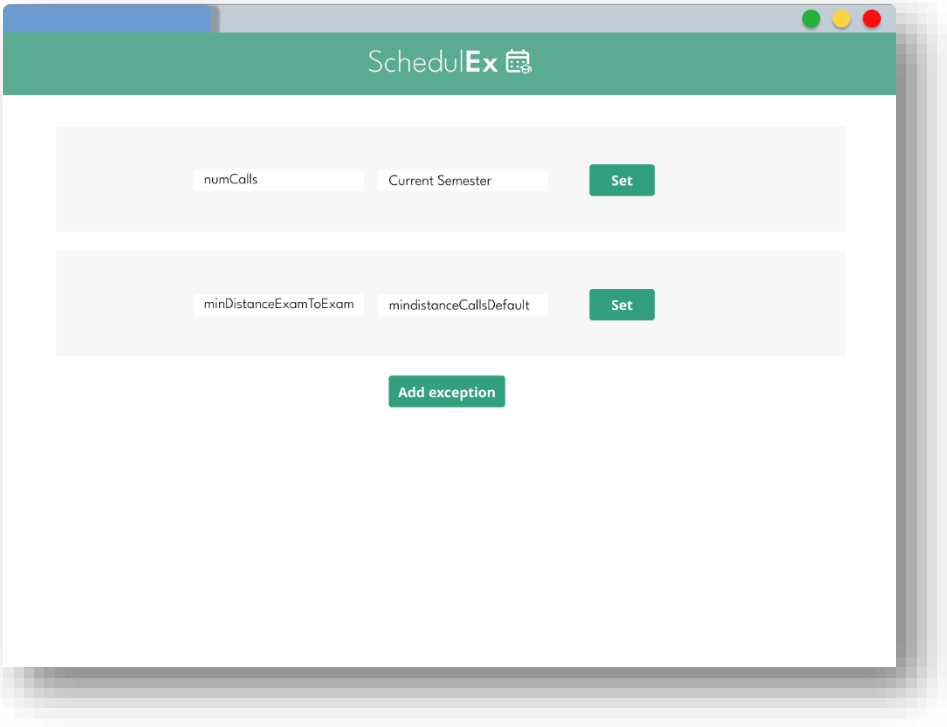


Unavailability Page interface

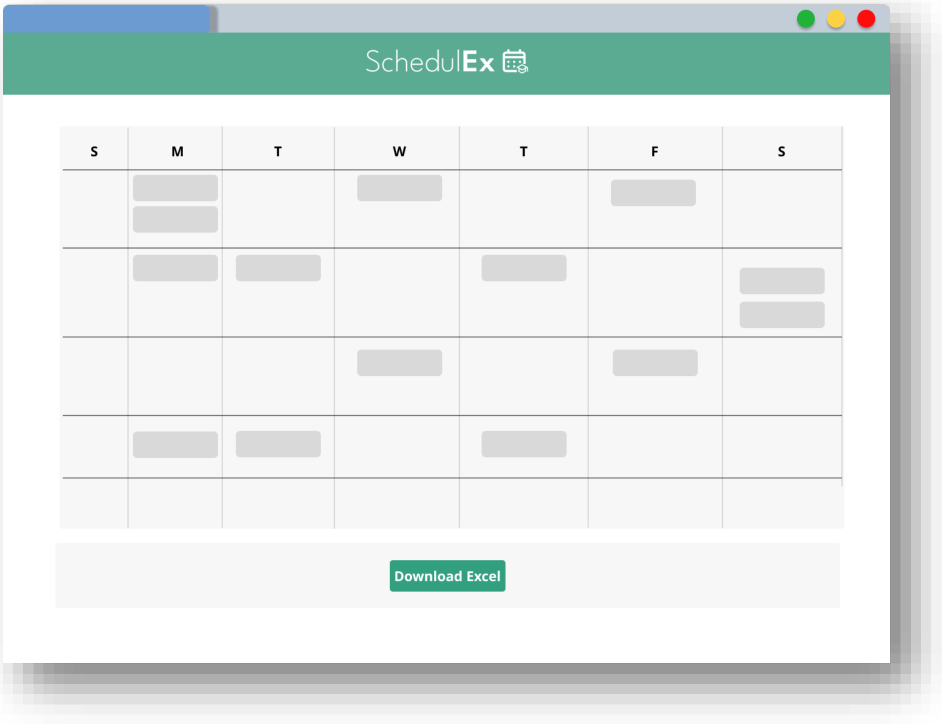


28/04/23

Settings Page interface



Calendar Interface


SchedulEx
S
M
T
W
T
F
S
Download Excel