Planning software user manual

Objective

With this software you give all the constraints of the employees and all the needs in the template excel, downloadable with the following link. You have to upload the data excel file and the algorithm will create a planning and give you an Excel file at the same location as the software with the proposed planning. The data are to adapt at each planning generation to fit with your needs like the number of hours needed for each person to work and the

https://github.com/kalashclean/planningSoftware/blob/main/data.xlsx

Sheets

In the excel there are many sheets. In this part each sheet is explained. This document is working as a database of the constraints for the software. So a **NUMBER** ID is used for each row and when two rows in different tables are linked, they use the ID of the other linked row. These IDs are used by the software to link information together.

Places

In this sheet you will define all the production sites you have where you need employees. Each site is just defined with a **NAME** and an **ID**. They will be used to associate employees to their favorite place. This constraint is just favorizing people to work on this place but is not obligatory.

Employee

In this sheet you will have to define each employee, with an **EmployeeID**, a **FirstName,** a **LastName**, a **WorkingTime**, a **JobID** and a **FavoritePlaceID**.

- WorkingTime : The number of hours you want the employee to work in the week, often based on his contract or how much he worked previously.

- JobID : Based on the IDs in the Job sheet, each Job used has to be defined in the Job sheet and only the number of the ID associated to the Job is accepted. If an employee can do many jobs, a list of IDs can be entered separated with a **« , »**, any other character will not work. This can be used to symbolize responsibilities too, like be able to close the shop.

- FavoritePlaceID : here is expected the number of the ID of the favorite place of an employee. The software will optimize the results to have the employee as often as possible in this place. It is not guarantee that he is always here if another site needs him.

Job

In this sheet you have to define which responsibilities and job exist in your companies, they can vary (nurse, cooker, people which have the key…). Each Job is defined with a **JobID** (always a number) and a **JobName**, which can be used to describe the job responsibilities.

Shift

In this sheet the goal is to define all the opening slots you have where you will need at least an employee to work (which kind of employee is defined is ShiftRequerement sheet). These sheet is important, you will have to define the good slots to always have the good number of people in each site.

These opening slots are defined with a **ShiftID**, a **PlaceID**, a **DayOfWeek**, a **StartTime**, an **EndTime**, a **ShiftDuration.**

- PlaceID : Is the place where you want to have a worker at this time
- DayOfWeek : is which day you want to have this worker, here only come the day, if many weeks are done, the number of the day of the month can be entered to differentiate them.
- StartTime : hour in which the shift is starting
- EndTime : hour in which the shift is ending
- ShiftDuration : Time between start and end time in hour

ShiftRequirement

In this sheet you will have to say for each Shift which job you want and how many people from this job. If you need many jobs for a same shift, you have to do many shift requirement rows.

This sheet is defined with a **ShiftRequirementID**, a **ShiftID**, a **JobID**, a **RequiredCount**

ShiftID : ID of the Shift on which you want to associate a specific job

JobID : ID of the Job of the person you want it come for this Shift

RequiredCount : Number of people with this responsibility which have to come at this shift.

Unavailability

If an employee cannot work at a specific time because he has a private meeting or a regular activities or holidays, you can put his unavailability in the Unavailability sheet.

It is defined with an **UnavailabilityID**, an **EmployeeID**, a **DayOfWeek**, a **StartHour**, an **EndHour**.

- DayOfWeek : it has to be in the same format as the DayOfWeek in Shift sheet. If you have regular unavailability you have to write each of them, for holydays, each day has to be written.

- StartHour: Time when the unavailability starts.
- EndHour: Time when the unavailability ends.

Duty

If you want to influence the software or you want to impose the algorithm someone to work at a specific time, you can add it in the Duty sheet.

It is defined with an **EmployeeID**, a **PlaceID**, a **ShiftID**, a **JobID**, all the IDs take only number of the ID associated with the other tables as input.

EmployeeID : Is the ID of the employee you want to force to work.

PlaceID : is the ID of the place where you want this employee work.

ShiftID: is the ID of the shift where you want this employee work.

JobID: is the ID of the job you want this employee do at this Shift.

Constraints logic and errors

This part is more technical and to help you to better fix the errors you can have. Some time if your requests are unfeasible, the software can tell you if is unfeasible. In this case, you have to proofread your file to find the constraints which are contradictory, for example you do not have enough employees for all the shift you put or too much are in holydays. The software is not able to tell you where the error come from. The only help I can give you is to explain how are built the constraints to do the planning.

Shift requirement

For each Shift requirement, a constraint is created to force the required number of people from a specific job to work. If no Shift Requirement is created, a constraint is created to ask to put 0 worker at that shift.

Duty constraints

A constraint is created to force one employee to work in a place on a job on a specific Shift.

Unavailability constraints

All the shift which overlap the unavailable hours are identified and a constraint is created for each overlapping shift to ask to this employee to do not work at this shift for all the jobs he can do.

Qualified jobs constraints

For each job, a constraint is associated for each employee if the jobID is in the list of his feasible JobID

Working hours constraints

For each employee, the sum of hours is counted and has to be under the number of hours required, to make more planning feasible, it is allowed to make employee work more if needed but discouraged.

Prevent Overlapping shift constraints

To allow to have many shifts a day (morning and afternoon for example) the software has to ensure the time does not overlap. So for each shift, he finds all the overlapping shift and set a constraint for all the employees cannot work in the shift and the overlapping shift.

Favorite Place

For the favorite place, no constraint is created but only a penalty when calculating the final planning if some employees are not in their favorite place.

Hints and current bugs

- Each time you generate a planning, the solution will be different if possible, so if a solution does not fit you, you can generate a new planning.
- All the planning generated are in the same folder as the software, so if you generate a new planning it will erase the previous one. So change the name of the planning if you want to keep it.
- If you want to modify the data sheets while you uploaded it in the software, that will not work until you close the software.
- You cannot generate a new planning if the previous one is opened, as they have the same name, it will try to erase the previous one and will not achieve.