

# Assessment Timetable Scheduler

## User Guide

---

**Dublin City University**

COMSCI3

Student: Jingyuan Wu | Supervisor: Darragh O'Brien

# 1. What This Application Does

---

This tool helps a module coordinator create a timetable for project presentations.

I built it for CSC1049 to make scheduling easier.

It handles constraints that are difficult to check by hand, like making sure no lecturer is double-booked, that every panel includes the project's supervisor, and that the schedule is as compact as possible.

## 2. Installation

---

### 2.1 Requirements

- Python 3.10 or later
- pip (usually comes with Python)
- Internet connection for the first install (to download OR-Tools)

### 2.2 Steps

1. Download or clone the project folder
2. Open a terminal in the project folder
3. Run: `pip install -e .`
4. Wait for OR-Tools to download (it is about 50 MB)
5. Run: `python main.py`

If you see an error about tkinter on Linux, run: `sudo apt-get install python3-tk`

Quick Start:

1. `pip install -e .`
2. `python main.py`
3. Load sample JSON
4. Click Run

### 2.3 Launching the Application

Double-clicking `main.py` may not work on all systems. The safest way is to open a terminal in the project folder and run `python main.py`.

## 3. Getting Started

---

### 3.1 Welcome Screen

When you first open the application you will see the welcome screen. Click 'New empty schedule' to start from scratch, or 'Open existing JSON' to load a saved schedule.

The Add and Delete buttons in the Entities tab will not work until you have created or opened a schedule. This is intentional — there is nothing to add to yet.

### 3.2 Recommended Workflow

6. Go to the Entities tab and open the Time Slots sub-tab. Add your timeslots first.
7. Add your lecturers in the Lecturers sub-tab.
8. Go to the Availability tab and click cells to mark which lecturers are free at which slots. Green = available.
9. Add students and projects in the Entities tab.
10. Set up constraints in the Constraints tab (panel size, lunch slots, etc).
11. Go to Run & Output and click Run.

The order matters. The availability grid will be empty if you have not added lecturers and timeslots yet.

## 4. The Entities Tab

---

### 4.1 Time Slots

A timeslot represents one assessment slot, for example Monday 9:00–9:30. You need at least as many slot-room combinations as you have projects.

#### Adding slots one at a time

Click Add slot. You will be asked for an ID, date, start time, end time, and an optional label. Date must be in YYYY-MM-DD format. Time must be in HH:MM format. The end time must be after the start time.

#### Generating a batch of slots

Click Generate batch. Enter a date, the time the day starts and ends, and the slot duration in minutes. The application will create slots automatically across the whole day at the given interval.

Example: date 2026-03-10, day start 09:00, day end 17:00, duration 30 minutes will create 16 slots.

### 4.2 Lecturers

Add a lecturer with an ID (e.g. L01) and full name. You can also set Max/day, which limits how many assessment panels they can appear in on any single day.

After adding lecturers, go to the Availability tab to mark their available slots. A lecturer with no available slots cannot be assigned to any panel.

### 4.3 Students

Add a student with an ID and name. You will be asked to assign them to a project immediately. If there are no projects yet, you will need to add a project first.

Students can have unavailable slots — these are blocked in the solver so their assessment is never scheduled at those times. You can edit this later in the JSON file if needed (there is no GUI for this yet).

### 4.4 Projects

Add a project with an ID and title. After adding it, click 'Set supervisor' to assign a supervisor from your lecturer list. The supervisor constraint means the supervisor must be on the assessment panel.

If `must_include_supervisor` is turned on in the Constraints tab (it is on by default), every project must have a supervisor set before the solver will run.

## 5. The Availability Tab

---

This tab shows a grid of lecturers (rows) by timeslots (columns). Green cells mean the lecturer is available at that slot. Grey means unavailable.

Click any cell to toggle it. The availability is not saved to the config until you save the file or run the solver — the Constraints tab's `write_back` and `availability flush` happen automatically when you click Run or Save.

You can scroll the grid using the scrollbars or the mouse wheel. Hold Shift and scroll to scroll horizontally.

## 6. The Constraints Tab

---

### 6.1 Basic

- Rooms — how many assessment rooms are available at the same time. Default is 1.

### 6.2 Panel

- Panel size — how many lecturers must be on each assessment panel
- Supervisor must be on the panel — if checked, the supervisor must appear in every project's panel

### 6.3 Lunch Slots

Select timeslots from this list to mark them as lunch slots. Hold Ctrl (or Cmd on Mac) to select multiple. The solver will try to avoid scheduling assessments in these slots. How hard it tries depends on the 'Avoid lunch slots' weight below.

### 6.4 Objective Weights

These control the soft objectives in the solver (Slice 3 only). Set a weight to 0 to disable that objective completely.

Weight	What it does	Default
Compact schedule	Minimises the gap between the first and last assessment	1
Workload balance	Tries to give all lecturers a similar number of panels	10
Avoid lunch slots	Penalises assessments in designated lunch slots	3

### 6.5 Solver Settings

- Time limit — the solver stops after this many seconds and returns the best result found so far. Increase this for large or difficult inputs.
- Workers — number of parallel CPU threads. 0 means use all available cores (fastest). 1 means single-threaded (same result every time, useful for reproducibility).

## 7. Running the Solver

### 7.1 Solver Levels

Choose a solver level from the dropdown at the top of the Run & Output tab.

Level	What it does	When to use it
Slice 1	Assigns projects to rooms only. Does not involve lecturers.	Quick check that the timeslots and rooms are sufficient
Slice 2	Adds panel assignment and lecturer availability as hard constraints.	When you want to include lecturers but do not need workload balancing

Level	What it does	When to use it
Slice 3	Full model. Adds soft objectives: compact schedule, workload balance, lunch avoidance.	Normal use — this is the recommended setting

## 7.2 Reading the Results

After the solver runs, the results table shows each project's assigned slot, room number, and panel. Results are sorted chronologically by the order of timeslots you defined, not by ID string.

The status bar at the top shows **OPTIMAL** (best possible answer found), **FEASIBLE** (a valid answer was found but may not be optimal, usually because the time limit was reached), or **INFEASIBLE** (no valid schedule exists with the current constraints).

The Diagnostics box shows any warnings or errors from the pre-solve checks.

## 7.3 If the Solver Returns INFEASIBLE

This means no valid schedule exists with the current settings. Common causes and fixes:

Possible cause	How to check	Fix
Not enough slots or rooms	Count your projects vs (rooms x slots)	Add more timeslots, or increase the number of rooms in Constraints
Panel size too large	Check panel size vs number of lecturers	Reduce panel size or add more lecturers
Supervisor has no available slots	Check the availability grid	Mark at least one slot as available for each supervisor
Too many students unavailable at same time	Check student unavailability in JSON	Remove some unavailability blocks or add more timeslots

# 8. Exporting Results

## 8.1 Export as JSON

Click 'Export JSON' in the Run & Output tab. The result is saved in the same format as the config file, making it easy to load back or process with a script.

## 8.2 Export as CSV

Click 'Export CSV'. The file includes Slot ID, Date, Start, End, Room, Project ID, Project Title, and Panel columns. The file uses UTF-8 with a BOM so it opens correctly in Excel without encoding issues.

## 9. Saving and Loading

---

Use File > Save (or Ctrl+S) to save the current config as a JSON file. File > Open loads a saved config. The config file contains all your entities, availability, and constraints — not the solver result. Save the result separately using the export buttons if you need it.

## 10. Using the Command Line

---

There is also a command-line version if you prefer not to use the GUI, or want to run the scheduler in a script.

Basic usage:

```
python -m schedule_app.cli --config data/sample_feasible.json
```

With output file and solver level:

```
python -m schedule_app.cli --config my_schedule.json --out  
result.json --solver slice2
```

The sample config file at data/sample\_feasible.json can be used to test the installation.

## 11. Troubleshooting

---

Problem	Likely cause	Fix
Application does not open	tkinter not installed (Linux)	Run: <code>sudo apt-get install python3-tk</code>
'ModuleNotFoundError: ortools'	OR-Tools not installed	Run: <code>pip install -e .</code> in the project folder
Add button does nothing	No schedule open	Click File > New or File > Open first
Solver is very slow	Too many variables for the input size	Reduce timeslots or increase the time limit
CSV opens with wrong characters	Excel encoding issue	File uses UTF-8 with BOM — it should work. If not, use 'Import' in Excel and select UTF-8 manually