Philipps-Universität Marburg
Fachbereich Mathematik und Informatik
AG Programmiersprachen und -werkzeuge

Colors:

Green: Done

Blue: Done differently

Yellow: Partly done / Not properly tested

Red: Not done

A Web-Based Interactive Schedule Planner for Lectures and Exams

Motivation

Each semester the appointments for lectures, exercise classes and exams need to be planned. This requires checking that no appointments of courses overlap, which are mandatory or likely to be followed together. For the big basic courses, the appointment are fixed for years and only change under very rare circumstances, but compulsory elective courses can change their schedule. And since they sometimes are offered irregularly, the schedule of their last run may not be compatible with the current semester. Creating an optimal schedule automatically is very difficult, because several boundary conditions exist for scheduling, which are not easy to formalize. Therefore, a graphical tool is needed that visualizes the current schedule, allows to change it by dragging events to a new place and highlights possible conflicts.

Assignment

For this assignment an interactive web page should be implemented with React and the Material UI, which is fully browser-based, i.e., without a server component. The web app should use the Node Package Manager for the build configuration and be hosted as a Gitlab Page. It must be possible to upload an initial schedule based on the schedule of previous semesters or a previously saved schedule. And it must be possible to select which lectures to include in the schedule.

Two planning modes must be available: Firstly, lectures and exercise classes are scheduled for the whole lecture period and each week has the same schedule, with the extension that some appointments may be bi-weekly and some courses may be block events; block events may take place outside the normal lecture period. One course can have multiple lecture and exercise classes per week. Secondly, exams are spread over a period of three or two weeks and fall into two examination periods. One class can only have one exam per exam period; courses with oral exams may or may not be included in the planning.

Information relevant for the planning are the lecturer, kind of event, the room and the expected number of participants. It must be possible to specify and edit this information, and some of this information may be optional. Furthermore, it should be possible to consider conflicts between lectures. For this purpose, as part of the assignment, the suggested study plans which are part of the exam regulations of all study programs should be analyzed and dependencies should be derived. This is, courses suggested to be studied in the same semester of any of the programs are in strong conflict and courses in adjacent study year have a weak conflict. Based on these information, the planner should check for conflicts in the planning and mark them; also other kinds of conflicts should be considered like overlapping lectures by the same instructor or in the same room. Conflicts should also consider patterns like bi-weekly events and block events; in particular block lectures can potentially also overlap with the exam periods, which should be detected.

It should be possible to move appointments in a convenient way such as using drag-and-drop. The current planning should be stored in the *HTML 5 local storage* such that the status is not lost if the web page is closed accidentally and planning can be continued when the page is opened again. Furthermore, it should be possible to download the current planning and to upload it again, for example, to revert to a previous planning status. It should also be possible to save the current planning in the form of a human readable list that can be used as instructions for booking the appointments in the course catalog and reserving corresponding

Info


Teaching Organization Tools


React, Node, MUI


Typescript (or Javascript)


FoPra or Project Work


3–5 Persons


Theory
Practice

Contact

Prof. Dr. Christoph Bockisch

Interested?


Info Meeting: April 18, 2024 at 16:00h

Participate in the non-binding info meeting! For further information on the meeting and short term changes, please join the ILIAS course.
`https://uni-marburg.de/j0kmMT`

rooms. In this list, it should be visible, which appointments have been changed compared to the initial planning.

Requirements

Core Requirements

The core requirements are described above.

Optional Requirements

- When dragging an event, the planner may highlight possible target time slots, where no conflict would be detected.
- The planner should have a read-only mode, in which a schedule is only displayed but cannot be edited; in this mode, the schedule information may be read from a file that is (manually) uploaded in the Git repository of the Gitlab Page project.
- It should be possible to apply filters to the view of the schedule planner, e.g., by instructor or by study program.
- The schedule (possibly with applied filters) should be exported to a PDF.
- The planner should offer possible choices of rooms for events based on the expected number of participants and room-planning for overlapping events.
- It should also be possible to include a list of possible rooms instead of just one.
- When the appointments of basic courses are moved, a warning should be displayed, because these appointments should usually not change.

Requirements from Protocols

13.05.2024 - 14:15

- Try to keep Wednesday afternoon (14:00) free from lectures (weak conflict)
- (Monday to Wednesday lectures)
- (Thursday to Friday exercises)
- Constraints from Bachelor courses most important.
- Study programs (May have to be entered manually. It should be flexible.):
    - Computer Science (start in winter or summer)
    - Business Informatics
    - Mathematics
    - Minor Mathematics
    - Business Mathematics
    - DataScience
    - Teaching profession
    - Finance
    - Cultural data studies
- Advanced courses are more flexible.

01.07.2024 10:15

Workflow to create new schedule

There are two data sources, a JSON and a XLS file.

- First a JSON file is used to read old data. The modules are displayed in a list. From the list there can be selected the desired appointments or modules, which are loaded into the calendar.
- Then it is modified and new modules are created.
- Having a ready schedule it should be downloadable into a human readable form.
- The schedule is manually transferred into Marvin.
- In Marvin the schedule can be further modified in respect to other study plans and room management.
- Marvin contains the authoritative schedule. It can be downloaded.
- It should be possible to upload the download from Marvin into the interactive schedule planner. Changes should be highlighted.

Creating new modules or appointments it may be nice to have an auto complete field.

In the appointments list there may be selected a default value.

For planning it may be enough to see an exemplary week including biweekly and block appointments.

21.08.2024 11:00

Distributing Issues

Some issues are related to each other. We formed issue groups and assigned the groups to team members.

- Upload (Mostafa)
- Download (Mostafa)
- CRUD (Michael)
- Filter (Michael)
- Conflict analysis (Michael)
- Pipeline (Alexander)
- Colours (Alexander)
- Saving locally (Alexander)
- Number of Weeks (optional) (unassigned)

29.08.2024 15:00

Clarifications

- Record changes: When uploading an initial schedule, working on it and finalizing it, there should be a printable version of the before and after, highlighting what has been changed.
- Bi-weekly appointments should be marked appropriately.
- We are to think in 2 hours timeslots. But c.t. and s.t. may be noted.
- In the Grouping Dialog there should be the possibility to select and deselect all.
- Copying semesters and periods should be done correctly, adjusting the times correctly.
- Periods: Last lecture week is first exam week. First exam period lasts 3 weeks. The second lecture period lasts two weeks and comprises the last two complete weeks in March or September.
- The downloading format we can choose ourselves, but it should be uploadable again.
- Human readable format: It should be possible to show the schedule to others so that they can confirm or deny the schedule. Ideas:
  - PDF with schedule
  - ~~JSON file, which lecturers can upload in our app, or which is human readable.~~
  - ~~Link to our app or another website where their schedule is displayed. Maybe this can be acheived by setting filters.~~
  - ~~Excel also may be an option as most lecturers are familiar with Excel.~~
- How improve performance? Use profilers. Update in less than a second is ok.

- Studyplans can be found on the website: [Link](https://www.uni-marburg.de/de/fb12/studium/studiengaenge)
- Conflicts:
  - Studyplan modules per Semester should not overlap.
  - Studyplan modules should not overlap with the modules of the next year, as some may have to repeat modules.
  - Summer and winter study start may be treated as two different studyplans (especially Computer Science).

Milestones

Until Milestone 4a) - 26.9.2024

- Convenient import & export & management of data
- Calculation of conflicts
- Highlight conflicts

Until Milestone 4b) - 31.10.2024

- HTML 5 local storage – auto save calendar
- Gitlab Page as host
- Resolve todos
- Improve code & performance
- Tests

16.10.2024 10:00

Human readable form

- PDF is ok.
- Generate PDF page per lecturer.
- Add fields LehrveranstaltungsID, rhythm and type
- Automatic generation of e-mail, that still can be edited before sending would be nice, but not necessary.

Data import

- It should be possible to import some initial data, via JSON-File.
- Tutorials and lectures may have different LehrveranstaltungsIDs, but belong to the same module. There are two conventions:
  1. Using sequential numbers for tutorials
  2. Adding "Übung" in front of the tutorial title.
- Our program should detect tutorials, exams and lectures, that belong together automatically as far as possible.
- But during the importing process it shall be possible to bring tutorials, exams and lectures together.
- What belongs together may be stored in an extra file and could be committed in the repository. This data does not change very often.
- In the xlsx-File there were hidden columns and the data in the second sheet in the last columns is meant to be JSON-data. It can be copied and pasted into a separate file. Then it is JSON.

The Workflow

1. Export the modules from the next semester from Marvin into an xlsx-file.
2. Open the app and upload the modules.
3. Select the modules to be deactivated or deleted.
4. Close the selection dialog.
5. Upload the modules from previous semesters (JSON-files from the archive).
6. Select the modules to be deactivated or deleted.
7. Close the selection dialog.
8. Copy the selected modules into the next semester.
9. Edit the schedule and resolve conflicts.
10. Download the schedule.
11. Distribute the human readable schedule to lecturers (via email).
12. Incorporate feedback.
13. Download Before & After schedule.
14. Include changes into Marvin.
15. Download final schedule for the archive.