Find your perfect timetable… Ask for a live demo now!

NUSMods Planner

Developed by Bay Wei Heng and Raynold Ng in collaboration with NUSMods

Introduction

NUSMODS

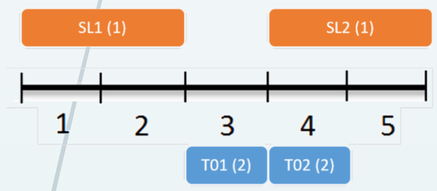
NUSMods Planner is an augmented version of NUSMods that ships with an automatic timetable planner that takes in a list of user supplied modules, the desired workload and additional constraints (i.e. free days, no lessons during certain timeslots) and returns a possible timetable that meets desired workload and constraints if it exists.

Problem Representation*

Timetable auto-generation is not a novel idea, and a simple recursive backtracking algorithm can be implemented to generate a timetable with no clashes. However, this naïve solution takes too long to generate "practical" timetables, thus a new approach is needed.

Satisfiability Modulo Theories (SMT) is the problem of deciding satisfiability of a logical formula, expressed in a combination of first-order theories.

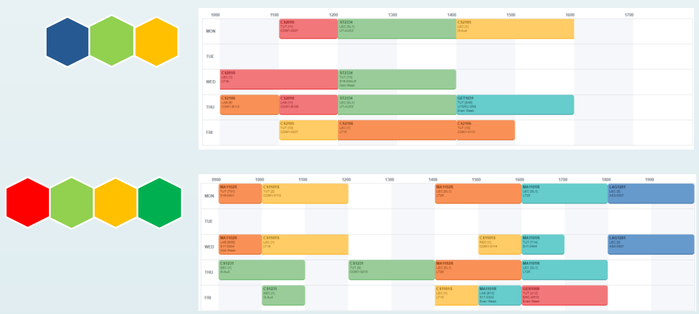
This project uses Boolector, an efficient SMT solver for quantifier-free theories, to "generate" our timetable, after posing user-provided constraints in terms of first-order logic. An example is provided below:



$$\begin{aligned}\phi_m = &(SL = 1 \rightarrow (H_1 = 1 \wedge H_2 = 1)) \wedge \\ &(SL = 2 \rightarrow (H_4 = 1 \wedge H_5 = 1)) \wedge \\ &(T = 1 \rightarrow (H_3 = 2 \wedge H_3 = 2)) \wedge \\ &(T = 2 \rightarrow (H_4 = 2 \wedge H_5 = 2)) \wedge \\ &(SL = 1 \vee SL = 2) \wedge (T = 1 \vee T = 2)\end{aligned}$$

Features

Example Options and Timetables:



User-centric Design

Minimalistic
- Only most essential features included

Interactive Tour
- Helps to onboard new users

User Feedback
- Avenue to provide suggestions available

FAQ Section
- Helps users understand design choices

Future Work

NUSMods has approved our feature to be included in their next release (v3), which will be out around December. From now till then, we will be working on rebasing and merging our code, as well as improving code quality to their required standards.

We will also be working on additional features. One feature many users would like is travel minimization. Although that in itself is computationally infeasible, we have ideas on how to apply certain heuristics to get a good, approximately optimal timetable.

Finally, it is possible to further optimize the response time by porting over the query-generation logic to Javascript using Boolector API directly. This would cut out the RTT between client and server.

User Feedback

On Usefulness:

> really well done! :) I believe this will help reduce so much hassle when planning for mods

On User-Friendliness:

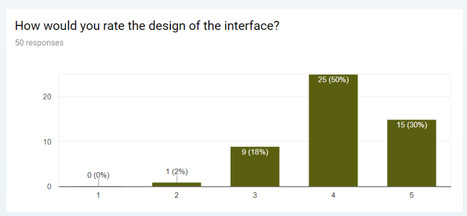
> We have tested the site and found the UI/UX of the site very likeable and enjoyable - Peer evaluators

On whether project is ready for use by target audience:

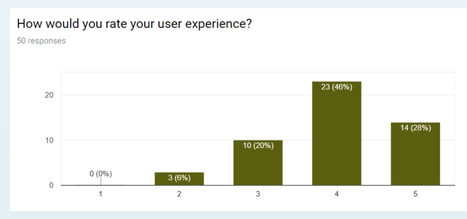
> All the features are fully functional and work exactly as described :D … [It's] more than ready - it's been deployed and used by actual users. Good job you guys! – Project Advisor

Usability Statistics

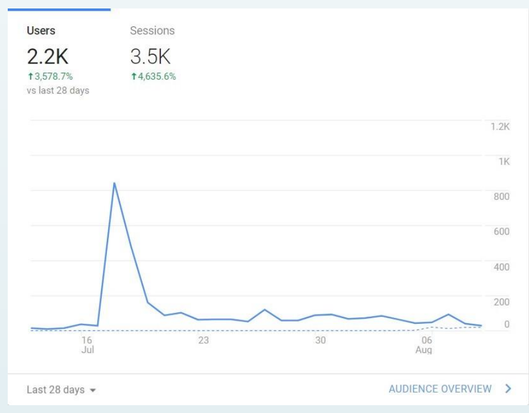
Average Score: 4.08



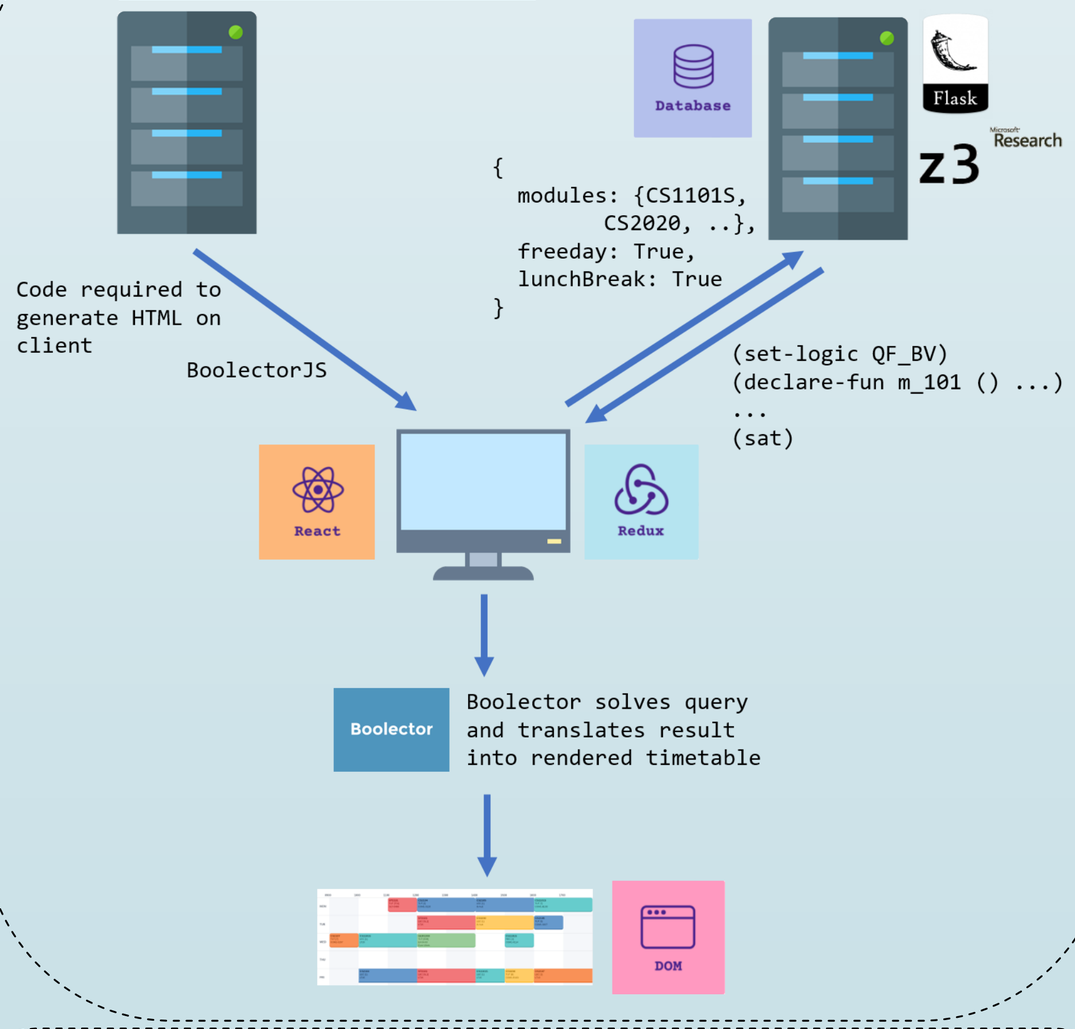
Average Score: 3.96



Google Analytics



Software Stack*



Acknowledgements

We would like to thank our mentors Li Kai and Zhi An for their guidance on the NUSMods codebase. We would also like to thank our advisor Kenneth Lu for his invaluable advice on UI/UX and software engineering.

*For full details see technical report: https://tinyurl.com/nusmodsplanner
Printed copies are available upon request

Server: https://github.com/raynoldng/nusmodsplanner-server

Website: https://github.com/raynoldng/nusmods-planner