Statement of work

Version 1.0
01/03/2024

School of Computing Resource Maximization Planning System

PROPOSER: Belinda Bergin

TEAM MEMBERS:
Alex Boxall (Backend designer)
Edward Nivison (Project manager)
Filip Mazur (Spokesperson)
Hexuan Meng (Backend and testing)
Matthew Cawley (Developer)
Rachel Cao (Beckend and admin)
Sineeha Kodwani (Deputy Spokesperson)

Background

In the dynamic environment of the Australian National University (ANU), particularly within the School of Computing, the efficient allocation of computer lab resources stands as a critical operational necessity. Belinda Bergin, the esteemed Education Project Officer at the School of Computing, currently shoulders the responsibility of assigning teaching activities to appropriate computer labs. This task, inherently complex and time-consuming, involves a considerable degree of problem-solving and logistical finesse, given the myriad of constraints and requirements specific to each course and teaching period.

This challenge is not unique to the School of Computing but resonates across various colleges within ANU, highlighting a systemic issue in the optimal utilization of campus resources, especially during the inter-semester intervals. The existing process, while functional, falls short of achieving the pinnacle of resource efficiency, often leaving a fraction of the valuable lab spaces underutilized or misallocated.

Understanding this issue, we're starting a project called the "School of Computing (SoCo) Resource Maximisation Planning System for Labs". This project seeks to revolutionize the current scheduling paradigm. The project is envisioned as a strategic response to the pressing need for a more streamlined, intelligent, and flexible scheduling system. By harnessing advanced scheduling algorithms and user-friendly interfaces, the system aims to maximize the utilization of computer labs, accommodating an extensive array of constraints, ranging from specific course requirements and staff availability to room capacities and technological amenities.

The overarching vision of this project aligns with the broader objectives of ANU to enhance operational efficiency, foster academic excellence, and optimize the use of its physical and technological infrastructure. Through collaborative discussions, active engagement with key stakeholders, and iterative feedback cycles, this project endeavours to encapsulate the diverse needs and expectations of the university community.

In essence, the SoCo Resource Maximisation Planning System for Labs aspires not only to address the immediate logistical challenges faced by the Education Project Officer and faculty members but also to lay a foundation for a more adaptable, efficient, and resource-conscious campus culture. The ultimate goal is to unveil a solution that not only streamlines the scheduling process but also contributes to the strategic objectives of ANU, paving the way for a more sustainable and effective allocation of its treasured academic resources.

Overview

The SoCo Resource Maximisation Planning System for Labs is a project that aims to create a more efficient way to schedule classes requiring a computer lab.

The solution will allow SoCo to make the most efficient use of limited computer lab space whilst adhering to pre-defined constraints, restrictions and requests from teaching staff.

The goal of this project is to deliver an application that can produce the best schedule for a varying set of constraints and courses for each teaching period.

Project objective

Efficiency: To develop a system capable of creating the most efficient lab schedules, maximising the use of available resources.

Flexibility: To accommodate a wide range of constraints and course requirements, ensuring the system's applicability across various teaching periods and evolving academic needs.

Usability: To provide a user-friendly interface that allows staff to input constraints, view schedules, and make adjustments as needed.

Scalability: To ensure the system can handle an increasing number of courses, labs, and constraints without a decrease in performance.

Basis

The following assumptions will be made about the problem, and form the scope of what is expected of the solution (the system):

1. The system is only concerned with the scheduling of labs for COMP courses at ANU, in the following rooms: HN1.23, HN1.24, N109, N111, N112, N113, N114

2. Only labs are to be considered by the system, and not, e.g. lectures or drop ins.

3. The lab schedule produced, should, as much as possible:
    a. schedule labs to rooms efficiency, such that fewer rooms are used
    b. for each course, have a variety of options for lab time (e.g. morning and afternoon classes, classes on different days, etc.)
    c. clash with as few other labs as possible, especially those with fewer labs, and those with many students in common.

4. The system is not expected to find a solution that meets all of the above conditions for all courses, and may prioritise certain courses or conditions.

5. The system will be provided with, for each class:
    a. the expected number of student enrolments
    b. the expected number of tutors
    c. hours that labs for that class must not be scheduled within
    d. the length of the lab
    e. the times and lengths of their lectures

6. The system will be provided with, for each room:
    a. the maximum number of students
    b. whether or not that room has projectors and/or recording systems
    c. other special features of that room that are relevant for lab scheduling

7. All weeks will be treated identically, which therefore means the system will:
    a. not take into account public holidays
    b. not take into account room unavailability in certain weeks, including due to deferred exams
    c. not take into account labs that only run in certain weeks

8. The system will treat dual-coded courses as a single COMP course containing the combined number of enrolments (e.g. 'COMP2300' will cover all of COMP2300, ENGN2219 and COMP6719)

9. Postgraduate courses will not be considered, as most will be merged into an undergraduate course code as per the above assumption.

10. Lab start times must be on the hour or half hour (e.g. 8am, 8:30am, etc.) and have a length that is a multiple of half hours, and between 1 and 3 hours long.

11. The system is only expected to be able to schedule at most 30 COMP courses, and at most 200 labs in total across those courses

12. Clashes between labs do not need to be taken into account so long as the clashing courses have at least 3 lab times

The system also has a number of constraints that will be worked within:

1. The system must be usable by less technical staff (e.g. professors or admin staff), and therefore should make it easy to input the data (e.g. through a CSV, or GUI).

2. The system must produce lab schedules in a reasonable time

3. The system must schedule labs in line with ANU policies, which include:
    a. ensuring all labs are scheduled within the hours of 8am-8pm
    b. ensuring that tutor to student ratios in each lab are at most the following, plus or minus 5:
        i. 1:15 for 1000 level courses
        ii. 1:23 for 2000 level courses
        iii. 1:26 for 3000 and 4000 level courses
        (the tutor to student ratio may be adjustable within the system)
    c. limits on how many labs in a row a tutor may teach

4. The system must ensure that labs are not scheduled at the same time as the lecture for the same course

5. The system should be able to show what the flow-on effects are should one of the labs be forcefully moved to a different time or room

The system has no external dependencies besides data about the labs, rooms and enrolments.

Success Criteria

From the listed assumptions and constraints above, and by considering the various stakeholders, we have the following success criteria that the project should meet:

1. The system has an easy way to upload/enter lab and enrolment data that could be used by administrative staff, e.g. a graphical interface, or input through a data file (e.g. CSV)

2. The system can generate effective and policy-conforming schedules for the allocation of labs for the School of Computing (where 'effectiveness' is given in assumption 3 in the *Basis and Scope of Work* section)

3. The system can show the flow-on effects of a manual change in a lab's time or location

4. The system is scalable enough such that it can schedule all of the labs that the School of Computing would normally run (e.g. up to 200 labs), in a reasonable period of time.

5. The system is flexible in that it can be adjusted to work for e.g. changes to tutor ratios, or take into account times where labs cannot be run.

Plan

Accepting data:

1.Develop a method to read raw data files and input course information such as course code, tutorial time, students' number

2. Develop a method and check if the extracted data is valid.

3. Allow users to input csv files as raw data, or through a graphical user interface

Processing data

1. Store the tutorial data in database for allocating algorithm.

2. Design and implement an algorithm for the system to allocate labs to times while taking into account student and tutor ratios.

Returning data

1. Develop an interface to visually display allocated labs and related courses.

2. Design the UI of the display timetable.

Stakeholders

The stakeholders involved in this project are, in order of importance:

STAKEHOLDER	REASON
Client	They are commissioning the system so that it can serve their needs in allocating labs to classes
School of Computing administrative staff	They may be using the system after it is complete to enter data about course enrolments, lab sizes, etc.
Course convenors	Need to provide information to the system/School about tutor numbers, and may also have other requirements that affect scheduling

And the project indirectly affects some other stakeholders:

STAKEHOLDER	REASON
Students	Effective scheduling of classes has a number of benefits to students:Leads to fewer clashes, meaning students can attend more classes Ensures the tutor to student ratio is maintained, so students get enough support during classes
Tutors	Effective scheduling of classes ensures that tutor to student ratios are not exceeded, making it easier for tutors to teach
Other users of the buildings in scope	Effective scheduling means less room times are taken up by labs, allowing others to use the rooms for other purposes

Process

We will employ an Agile development methodology, iterating through phases of planning, design, development, testing, and deployment. This approach allows for flexible adaptation to evolving requirements while ensuring project milestones are met. We will also use project management and collaboration tools to track progress, manage tasks, and facilitate communication.

Regular status meetings with stakeholders and project review meetings will be conducted to discuss progress, obstacles, and next steps. Updates will include project status, upcoming milestones, risk assessments, and change requests.

We will conduct risk assessment meetings to identify and analyse potential risks, develop mitigation strategies for high-probability or high-impact risks, and monitor the risk indicators continuously.

We use GitHub for our issues and change management system. We use it to log and assign issues for resolution. We will also establish a clear escalation path for timely resolution of critical issues, ensuring minimal impact on project timelines. We also use it for submitting, reviewing, and approving change requests. All changes will be documented and communicated to relevant stakeholders to ensure transparency and alignment.

Some internal communication, and running online meetings, will be done through Microsoft Teams.

Schedule

Week 1: Team Formation

Week 2: Client Meetings, Statement of Work drawn up, Landing page, GitHub, assignment of roles.

Week 3: Audit 1, Backend leader will decide on a codebase to use, Frontend leader will begin researching frontend solutions (more than already skimmed).

Week 4: Structure will have a basic layout, a plan of action for what needs to be completed first will be drafted, and likely a new Statement of Work drafted as well.

Week 5: Developers will work on getting deliverables and milestones completed efficiently.

Resources and Costs

The only expected costs for the project would be for the hardware for the server (i.e. a computer). During development/prototyping a student's computer could be used, and the final system is expected to run on computers provided by the School of Computing / university at no cost. There should be no need for domain name registration as the system is expected to run on the internal university network.

The other resources required is data about labs, rooms and enrolments, which have been provided by the client.

Contingency Plan

In the case that a deliverable is showing itself unable to reach completion by the given time window, there will be a few things that will happen.

Firstly, there will be regular checkups with the team in the weekly Team meetings held to make sure that everyone is keeping afloat with the milestones that they have been working on. If anyone needs assistance, they are always welcome to reach out to the other team members.

However, if a deliverable is not able to be completed due to a reason outside of the teams control. It will be superseded to high priority and will be resolved as quickly as possible to allow the project to move forward. This means if there is an issue, for example, a hosting service for the database fails, the team will work to find a suitable replacement product in the meantime or a temporary solution to allow the workflow of the project to continue. With the permission from the client(s) and stakeholder(s) respectively.

And under the case where a deliverable is not feasibly able to be completed in the time frame or the time is best suited for a different deliverable. Then there will be a client meeting organised with the respective client(s) about the removal of that deliverable and the allocation of that time and resources to a different deliverable. However, this should be the last stage after trying to resolve the issue.

IP and License

The IP of the solution will remain the property of the students in the group, so long as the School of Computing is able to continue to use and modify the software freely.

Acceptance