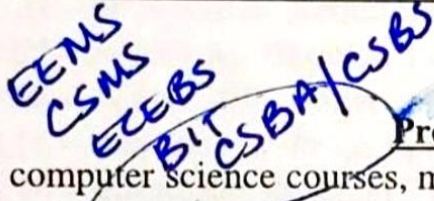


Project Title: Schedule Reconciler	
Start Date: 2020.02.12	End Date: 2020.05.04
Project Manager: Elias Julian Marko Garcia	
Project Sponsor: UMKC Computer Science Department	CSEE
Customer: Campbell, Gina	
Users: Department Administrators	
Stakeholders and Expectations: <ul style="list-style-type: none"> • Team: Coordinate together to resolve project issues and deadlines, communicate and reach-out to Gina and/or Bingham for clarification of requirements and deadlines, reach out to Gina for feedback on design and application behavior. • Gina Campbell: Given an application that facilitates the resolution of department professors' schedules and course scheduling. • CS Department: Obtain the ability to more readily generate schedules across semesters without unnecessary manual intervention on the part of administrators. 	
Purpose: <p>  Problem For scheduling computer science courses, manual intervention is currently still required for the preliminary drafting and generation of potentially valid schedules based on professors' scheduling conflicts and preferences. This means that administrators like Gina Campbell spend hours, with pen and paper, outlining and resolving the minutia of professorial scheduling constraints for a given semester in addition to other hard constraints such as course sequencing and undergraduate credit requirements. </p> <p style="text-align: center;"><u>Opportunity</u></p> <p>Create an application that assists department administrators to resolve course scheduling of a given semester by taking into account the conflict and preferences of professors. Such an application could then generate department course schedules that either reconcile professorial schedules with a given semester's courses or, as fallback, provides second-best schedules to serve as a basis for administrators to resolve more complicated department scheduling conflicts.</p>	

Goals and Objectives:

Goals

Develop an application that facilitates the scheduling of computer science department classes by assisting administrators to resolve department staff and professorial scheduling conflicts and preferences.

Objectives

- Develop a minimally viable product: a web application with a viable scheduling algorithm and ancillary book-keeping utilities.
- Provide a UI/UX experience that is both easy to use for less tech-literate audiences along with maximizing the utility and lifespan of the product until the department acquires better schedule automation. Keep it simple and maximize functionality.
- Report data in a format that is easy to export and use outside of the application. Application results should be easy to integrate into the paperwork, digital or physical, of the department for scheduling documentation.

Schedule Information (Major milestones and deliverables):

- 2020.02.17 – **milestone**: Iteration 1 begins
- 2020.02.23 – **deliverable**: Project charter
- 2020.02.29 – **deliverable**: Requirements document baseline
- 2020.03.02 – **milestone**: Iteration 1 deadline and iteration 2 begins
- 2020.03.06 – **deliverable**: Project plan
- 2020.03.16 – **deliverable**: Project prototype and presentation demo
- 2020.03.16 – **milestone**: Iteration 2 deadline and iteration 3 begins
- 2020.04.03 – **deliverable**: Architecture document
- 2020.04.06 – **milestone**: Iteration 3 deadline and iteration 4 begins
- 2020.04.20 – **milestone**: Iteration 4 deadline and iteration 5 begins
- 2020.04.26 – **deliverable**: Test plan
- 2020.04.27 – **deliverable**: User and system documentation
- 2020.05.01 – **deliverable**: Project results
- 2020.05.04 – **deliverable**: Final product release
- 2020.05.04 – **milestone**: Iteration 5 deadline

Financial Information (Cost estimate and budget information):

- Project Manager [1]: 8 hours/week for 13 weeks: $104 \text{ HR} * \$66/\text{HR} = \6864
- Software Architect [2]: 4 hours/week for 13 weeks: $52 \text{ HR} * \$66/\text{HR} = \3432
- 2 x Software Eng., LII [3]: 4 hours/week for 13 weeks: $104 \text{ HR} * \$42/\text{HR} = \4368
- **Total: \$14664, and \$56.40/HR averaged**

[1]: National average hourly pay, Project Manager.

[2]: National average hourly pay, Software Architect.

[3]: National average hourly pay, Level II Software Engineer.

Project Priorities and degrees of freedom:

Scheduling

- Not flexible, fixed dates.
- End of semester == End of project.
- Teammates personal schedules liable to conflict for collaboration time.

Budget

- Fixed, non-negotiable.

Priorities

- Web application with basic scheduling functionality.
- Ability to save, retrieve, and export scheduling results.

Approach:

Agile with scrum methodology with roughly 2 week sprints for project iteration and re-assessment. Requirements and specification liable to change after any given iteration to maximize likelihood of delivering a functional MVP. Weekly team meetings are a minimum for reporting updates, preferably 3 times a week (one per post-capstone class).

Constraints:

The software should be entirely free and open software (FOSS), with a liberal license so that it may be extended and updated as necessary in the future by any potential maintainers and/or the university itself. This also allows the university to use the software as needed by avoiding any liability with respect to usage constraints and copyright attached by a closed-source or licensing based software application.

Assumptions:

- User is able to access a university computer with a recent version of a mainstream browser, e.g. Chrome or Firefox, and can access the internet and/or internal network.
- Computer has access to printing facilities and/or can export data to common third party client, e.g. email.

Success Criteria:

Baseline success

- Basic scheduling logic.
- Basic data storage, retrieval, and updating logic.
- Basic data exporting logic.
- Hosted from a server, accessible via browser on the network and/or the internet.
- User friendly UI/UX as determined by stakeholder's feedback on final deliverable.

Above baseline (reach goals)

- CS department, Gina Campbell actually want to use application after the semester.
- Application is refined with a UI/UX that can be used for the foreseeable future by department administrators.
- Codebase is rigorously documented and tested for sustainability and future maintenance as needed.

Scope:

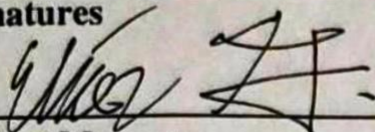
Web Application

- Allows user to upload scheduling data for professors and university courses.
- Allows for the storage, retrieval, and update of a user's saved data.
- Allows for the generation and exporting of scheduling results based on user's inputted data.
- Allows for ease of use and simple workflow via intuitive and/or minimalist UI/UX that does not get in the user's way.

Risks and obstacles to success:

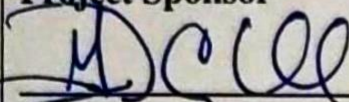
- Team is unfamiliar, unacquainted with each other as students and professionals.
- Team is made up of students, many who do not have any extended experience with developing software.
- Team is made up entirely of students without extended experience in developing *web applications*.
- Team may struggle to communicate and/or cooperate as group projects are liable to suffer from by nature.
- The difficulty of the project problem is not yet fully understood.
- Developing a flexible application that could potentially integrate/extend with other services/use cases remains ambiguous with respect to what integration looks like and what other application, use cases could be.

Signatures

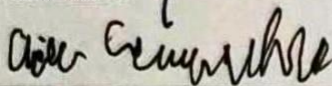


Project Manager

Project Sponsor



Customer



Technical Lead