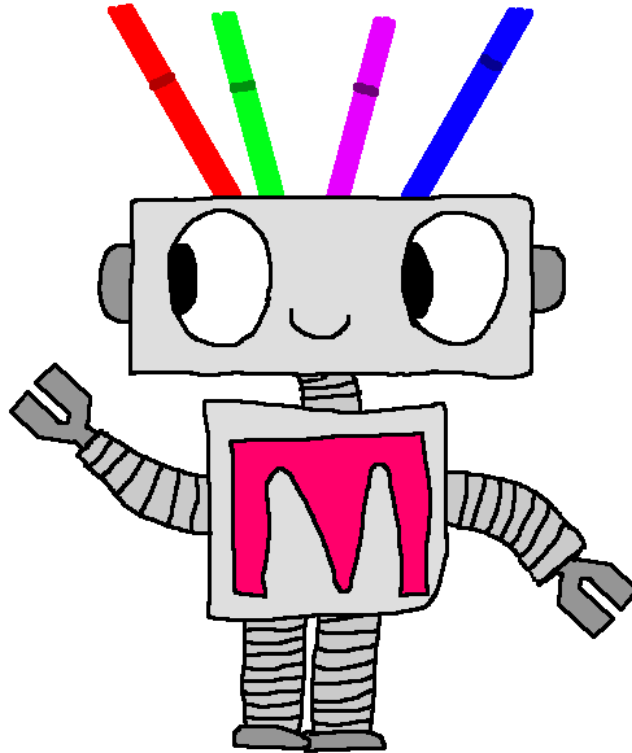


M.A.R.C


Marker Allocation Reconciliation Configurator



An automated system with markers on the mind

Team member	ID	Roles (Provisional)
Spencer Smith	988448770	Team leader
Qiwon Kan	208565088	Back-end REST API Dev
Dave Wright	1570128	Service Delivery
Sukrat Ubha	421843390	Back-end database
Pak Wing Andrew Hung	535199951	Rest API Client Dev

Executive Summary

 Currently, Course Coordinators send expected workloads of the course and marker preferences to the Marker Coordinator. Then, the Marker Coordinator gathers all the information from every course coordinator and puts it manually into a spreadsheet. This is tedious, time consuming and risks mistakes as there is no systematic facility.

Therefore, we are creating a web-based tool: MARC: Marker Allocation Reconciliation Configurator. It is a database of student markers who are allocated to marking assignments & tests of various classes.

MARC is intended for the Marker Coordinator, Course Coordinators, and Students(Markers). Students register their details & interests, while the Marker Coordinator and Course Coordinators can coherently organize students into their assignments as markers.

Our team members have done web development courses such as CS235 and CS335 along with numerous other programming languages which are similar in logic. Hence, we are confident in creating and implementing all these exciting features into this project!

Background and Rationale



Marker coordinator plays an important role in finding eligible markers and allocating them to the particular course. However, they have to do everything manually in the existing model which is time consuming and tedious.

Spreadsheets are the current tool, and are used to manually track and assign markers to courses. While being time consuming, this approach also requires constant input from the marker coordinator. This input may be in the form of adding courses, adding markers to the spreadsheet and filtering potential markers based on their skills, experience and preference; as set by the course coordinators, if any.

To expand, the marker coordinator currently has to send emails to both markers in order to recruit potential candidates as well as the course coordinators to gain relevant information about the course assessment being marked. These emails are then, in some cases, followed by reminders and clarifications, making the whole process strenuously tedious - and to put it simply: manual. After applications from the markers are received, the marker coordinator manually selects candidates who fit the criteria, assigns from the total marker hours available, and the hours needed to mark the assessment (done by taking into consideration the number of students and tutor hours). In many cases, the number of hours to assign is measured by a percentage, to mitigate potential uncertainties i.e. number of students in the course.

The client, Bukhard Wuensche, has elucidated that there are currently no existing solutions to the issues he has put forth.

Since this project will be the first of its kind, at-least in the Science department (as cited by the client), the center of attention has been to reform the approach implied to resolve the issues at hand.

The following solution is intended: The course-coordinators will be able to submit relevant information for the course through the web interface. This information will automatically be updated into the database. The markers can submit applications with their required relevant information which will also be automated to the database and will be used by the marker coordinator to determine successful applicants. This part of the project can also be automated once prerequisites are met under the given time limit but is not deemed essential by the client. The marker coordinator will have complete authority of the database to edit information in full capacity.

This system will be called MARC, or Marker Allocation Reconciliation Configurator.

It is also important to note that since a solution like the one outlined above is unprecedented for the UoA marker coordinators for the Science department, this project is worth exploring in it's very essence because we will be developing a solution that has the potential to solve similar problems experienced by every marker coordinator for the UoA, even those outside the Science department.

In building this project, the value attributed to the final product will be produced by the usefulness for the client. In this case, since the client will no longer have to manage an extensive spreadsheet, rather manage a more UI based approach in the form of a website and database, the value is all the more supplemented - with further contributions from automation wherever possible (i.e. Marker allocation, marker application submissions and database management).

Application Design Methodology



Allocating markers to courses involves navigating many constraints. The objective of the MARC tool is to not constrain allocations, but rather merely facilitate the decisions made by the Marker Coordinator. In fact, MARC is intended to remove constraints that are imposed on the Marker Coordinator needed to simplify his work.

It is intended to be highly visual and self-explanatory, with high interactivity.


It should be familiar in that it works similar to what used to work, but different from what didn't work: it leverages the Marker Coordinator's previous system's strengths such as colours and organized statistics.

The information should be centralized, and identical to both Course Coordinators and the Marker Coordinator.


It must be secure, fast and never fail; The design of the tool should be humble first in its required functionality before becoming ambitious with other parts and features.

Specific Aims / Objectives / Deliverables

~~Web Application for Marker Coordinator and Course Coordinators~~

MARC attempts to create a live 'working model', to allow fluid changes to marker allocations at any time, and by either party, governed by abilities & constraints imposed by the Marker Coordinator. 

MARC is to impose no constraints on the Marker Coordinator, but rather facilitates ease of Marker assignment and interaction with Course Coordinators. Allocation errors are at most presented as warnings.

~~Key atomic unit of the entire system is the allocation status of Students.~~ 

On a per-course basis:

- Marker-Coordinator has the ability to pre-allocate Students to a class with the provisional status of 'Marker-Coordinator Offered'.
- Course Coordinators have the ability to pre-allocate students to a course, with provisional status of 'Course-Coordinator Requested'.
- Marker-Coordinator has singular authority to change a student's association with a course to 'Assigned'

Additional features and constraints in order to facilitate the above are listed as follows.

Web interface for Marker Coordinator:

The screenshot shows a web interface for a Marker Coordinator. It is divided into three main segments: Overview, Details, and Global functions.

Overview (Left Pane): This pane displays a table of classes and markers. The table has columns for Class, Marker, and Status. The classes listed are COMPSCI 223, COMPSCI 220, COMPSCI 101, and COMPSCI 316. The markers listed are Spock, James Kirk, Hikaru Sulu, Worf, Gowron, and Koloth. The status of each marker is indicated by a color-coded box: green for 'Human', blue for 'Requested', and red for 'Human'.

Details (Right Pane): This pane shows context-aware details of the selected item, James Kirk. It includes information such as GPA (2.33), Rank (Postgrad), CV (View), Hours Offered (50), Hours Allocated (45), and Notes. It also lists Preferred / Preliminary Classes and an AUTO-ALLOCATE section.

Global functions (Top): This section includes buttons for 'ADD CLASS' and 'Assign / Recommend to new class'.

GUI in three main segments: Global functions, Overview pane, and Details pane

- Overview: the left pane. Shows all classes and markers as potentially-draggable boxes, for easy graphical assignment.
- Details pane: Right, shows context-aware details of selected items.
- Global functions such as 'Send email to __ regarding' or 'Export __' are provided along the top (not shown).
 - Export list of markers per class, for all classes
 - Email all incompletely-allocated class' coordinators.
 - Total student hours needed, allocated

Class representations

Classes are each row, and markers are positioned on the same row, indicating their assignment.

- Class workload information is displayed in the overview (left) in terms of hours (when available).
- Each class will show its workload requirements
- Clicking a class will provide a detailed breakdown in a details-pane, showing:
 - Detailed class information (coordinator details etc)
 - All the Markers allocated (or requested/offered) to that class.
 - Automatic-assignment facilities.
 - Free text boxes for any purpose.
 - Pending client review: checkboxes for the status of each class (done, incomplete, etc).
 - Export list of allocated markers in that class to CSV. (optional)
- Class colors in overview reflect workload fulfillment status (mix of colors).
- Classes can be sorted vertically by needs in any arbitrary order. Usability only feature - implies no functionality.
- Classes can be horizontally compressed visually (toggle) (optional)

Marker representations

Markers (*Students* when unallocated) are represented as boxes. A marker appears on the rows belonging to classes of which they are allocated to.

- Dragging a marker to a different row will reassign the marker to a different class. Also can create duplicates of Marker to assign a marker to multiple classes.
- When a student is selected and shown in the Details Pane
 - All the marker boxes in the Overview are 50% opacity, except for the selected marker's box which remains 100% opacity, indicating it is selected.
 - The marker's top choices will be visible as buttons to allocate the marker to that class.
 - A class can be manually typed-in (or drop-down selected) to allocate the marker to that class.
- Markers are be colorized, indicating their allocation status
 - Allocation status is a multi-boolean value for each class:
 - Course-coordinator requested
 - Offered by Marker Coordinator
 - Assigned to course (only selectable by Marker Coordinator)
- Markers can have their box (in the overview) predominantly display their name or relevant information.
- Markers dragged leftward are highly-requested (by convention, not enforced).
- An 'Unallocated Students' row in the overview will show all unallocated students, which when clicked, causes the info-pane to provide detailed sorting information or even operations to perform on unallocated students. ***Shown in demo picture***
- When Unallocated Students is selected and shown in the Details Pane
 - Text input box search for a Student's name
 - Sort by GPA
 - Sort by study level (postgrad, phd, etc)
 - ***Experimental auto-allocation function if possible***

Web interface for course coordinators:

This section is also available to the Marker Coordinator for all classes. It is the exact same UI as the Marker Coordinator, however constrained to their single class and the unallocated students.

- Info-pane has edit fields for the course title and Relevant information (plaintext)
- One preliminary marker workload's quantity (to mark ONE student)
- Field to specify Marker workload types (possibly text-only, not systematized)
- Skill requirements for the Markers
- Whitelist / Blacklist of Students for specific courses: plaintext fields for each student, or selectable from a list of students if registered with MARC.
- Number of students enrolled in the class: **this computes the total hours (estimate) needed for the class**

Web interface for students to apply to be a marker:

A separate, single page only - basically a single online form.

- Provides a selection of available courses, and roles available for each course.
- Ability to select preferences in order
- Ability to input GPA, CV, relevant skills, study level (phd, post grad etc) (subject to client conformation of systemization)
- Relevant overseas information - Visa, Residency, Experience etc.

Architecture, Authentication and security

- Security is a first-class principle and enforced server-side.
- All users of the system must be authenticated, and authorized against each endpoint.
- Students have no access to any API endpoints with the exception of the registration of their interest.
- ***Changes made by one entity should ensure consistency with other entities.***

Deliverables

- Node.js back end server application, with Sequelize ORM middleware
- SQLite database or MySQL database server.
- REST API
- Web interface in React, with Axios middleware.
- Installation scripts
- Documentation detailing the installation & requirements.
- Source code via Github.
- Demo installation, pre-installed.

Project Approach

Roles

The MARC project team consists of five members collaborating in different locations and across different time zones. The team has a collective understanding of ~~the~~ MARC's requirements, and such will all contribute to the ideas, coding, and/or testing of multiple components of the project.

Our small number necessitates a customized development model.

Rather than **fixed roles which then get assigned tasks**, upcoming tasks are grouped & allocated to members as they become apparent. This ensures that the roles available in our group are always a **function of the required work**. For example, a **Database Admin** may be required only for a brief moment in time. We will aim to have consistency with which areas each member works on.

Members are assigned to lead either the **development** of an aspect, or **testing** that aspect. As testing is required on all aspects, we each are responsible for testing a **different aspect** from what our development role is. This ensures each aspect has some external form of testing from its developer.

Finally, there will be the role of **Service & Delivery Manager**. This role is responsible for the communication between the client and the team, aiming to best deliver the client goals for the project. This role includes **oversight** of the integration testing, though integration tests are actually written by the respective team members.

Preliminary Roles

- Service delivery
- Team lead

Each development role has an accompanying Test role which is fulfilled by a different person.

- Back-end database development
- Back-end database API development
- Back-end REST API development
- Front-end GUI & Navigation development

Development Orchestration

Using an Agile methodology allows us to develop the project in iterations: with core functionality and then features in successive iterations. Agile also allows us to be more adaptable later on during iterations - as we can make improvements and changes to features and the design throughout each iteration.

Due to both the short-timeframe and long-distance setup of our team (as well as work responsibilities), coordinating iterations and having rigid temporal structure (a.k.a scrum methodology) provides its own challenges & inefficiencies.

Which is why we are opting for Kanban.

Kanban allows for development iterations based on continuous flow and continuous delivery. With the sub-tasks and their lead times pulled through the kanban board, it can facilitate how much work is required for each task & how to allocate development resources.

The similarity of our collaboration challenges to the actual problem case of MARC wasn't lost on us. Discrete moments of time as milestones and time-estimates are difficult for the Marker Coordinator as it is for us. This is key to designing MARC to function asynchronously.

Equipment

Our team will use a virtual kanboard with Jira as our main tool for our development orchestration. Viewing the kanban board frequently will allow for the team to visualise the tasks we have to complete for each iteration, which we will view daily as to upkeep focus, motivation, and morale throughout the duration of the project.

Version control will be orchestrated with Github.

This will allow for the team members to collaborate while working on individual remote depositories and to push changes to the central repository, which is particularly beneficial due to the long-distance setup our team has due to COVID-19.

Additionally, Github is an industry standard and will be beneficial for our team to become more familiar with professional software development tools and how teams cooperate towards larger scale projects.

Project Plan

Taking the above project priorities of the client into account, the team will first design the database and its base intractability for the target users (Marker co-ordinator, course coordinator, and students) such that the tool is capable of assigning markers to courses.

Once the base technical functionality & interconnections are built, the team will add features and functionalities for the purpose of the end-use application.

It is expected that the web front end will become the dominant workload towards the end of the project.

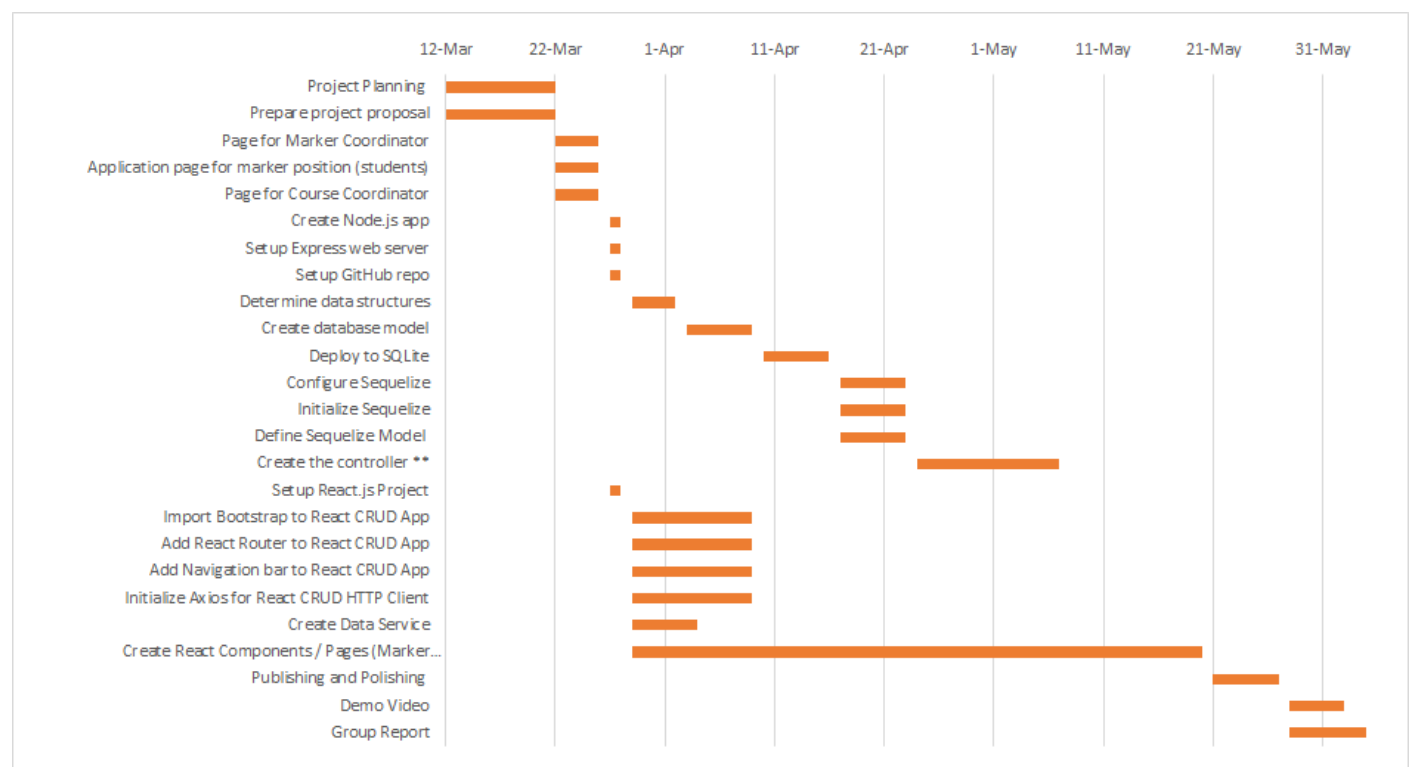




Table of Authorship

Team member	ID	Section	Signature
Sukrat Ubha	421843390	Background and Rationale	Sukrat S Ubha
Qiwon Kan	208565088	Executive Summary, Project Plan	Qiwon Kan
Dave Wright	1570128	Title, Executive Summary, Specific Aims, Design Methodologies, Project Approach	
Spencer Smith	988448770	Executive Summary, Project Approach	
Pak Wing Andrew Hung	535199951	Project Plan	Pak Wing Andrew Hung