Report 1.4

FITFORGE ALLOCATION SYSTEM

T039 | P062

28/7/2024

Table of Contents

Abstract

Due to the delays in the expected development, in addition with restarting communications with our industry partner and redefining our plans with them, the report for last week was skipped, having not many significant things of note. As such, this report will outline our new schedule, the minutes from the meetings over the past week, a risk assessment, and an outline of the overall system design and functionality, as it has changed in the past few weeks of planning.

Overall Goals

During our meeting with our Industry Partner on the Wednesday of our first week back, 24/7/24, it was indicated that due to their busy schedule they would only be available to test the system on the real situation and data on the Thursday of next week, the first of August, and the next available time would be several weeks afterwards. As such, we have discarded our previous sprint plan, and instead of redesigning our focus, we have decided to entirely commit to development and fix any bugs as we go. As of the end of this week, we are one week out from where we expected to finish the project. As such, we are still waiting on the sample data from our industry partner, so that we can refactor the code into working on the expected data structure. In addition, we will spend the remaining days finalising the last features that need to be merged into the main branch, reorganising the location some UI elements, and implementing the final navigational functionality.

State of the Project

As it stands, our project still holds to the initial outline defined in Report 1.0, to a very general degree. As we have developed, there have been several changes to the final vision as the feasibility of elements was discussed and reconsidered, and it would be worth highlighting how the system should function. This will be outlined in more detail in Report 2.0, which will incorporate a high level description of version 1.0 of the system as is ready for testing, in conjunction with the analysis of the testing and our plans from there.

The system incorporates three main sections of development. Significantly, the implemented algorithms have not changed in choice, nor been expanded as was once considered for this phase. Both Integer Linear Programming and Gale-Shapely have been implemented and are

functional as of this report, to the backend data implementation before being updated to the required standard. In the future, depending how useful they are in the testing, we will expand to incorporate more algorithms, particularly those that could provide multiple options.

The largest overhaul required once the data structure is obtained will be the CoreService class, which provides the functionality to the system. This class is responsible for storing the data in state as the program is used, calculating the b values as defined in the initial specifications, and serving any other required data that should not be calculated on the backend. This class is initialised as a React Context and is thus a persistent instance throughout the program life. It is from this function that listeners will pass updates when certain data is changed, allowing useEffect to be rerun on components. In addition, this class is able to save the state of the program, and then load the saved data to reestablish the previous state.

The UI element has mostly remained consistent in the overall component choices, with them becoming more detailed as we worked. Notably, the upload display has changed to properly handle multiple file uploads, and to properly identify the sheets being uploaded. Currently, it also has an option to specify the number of teams that can be assigned to a project. Based on the user flow as we work, this will be moved to its own separate view in the future, as that may be changed during usage.

The spreadsheet is still not perfect, having some issues with scrolling, however the major details have been implemented. Each cell is identified with the overall b value, in addition to a background colour, making it easy to quickly distinguish them from each other. The goal to make it possible to navigate using the left and topmost colum and row in the spreadsheet has not been implemented although that is desired.

When testing with large sample sizes on the list view, it was decided that to simplify the view, it would be easier to create tabs to separate into groups of ten, and then implement a second column of items to make it easier to view them. Scrolling is not optimal, and as the purpose of the list view is to easily compare as many pairings as possible in a simple manner, then it makes sense to separate the pairing divs across the screen as well as down. What this means is that the sorting function will need to be rewritten, to properly handle sorting across tabs, however, that is again not a priority for testing.

The navigation between components has been implemented as an extension on the CoreService Context, defining a type which contains a string of type Page, and a data parameter of type any. This is technically bad practice, but it will allow for switching between views, and for them to read the data needed when doing so.

Finally, one of the UI interaction options should be editing the values stored, not just comparing them. While this could be implemented as CoreService stands, because of the full rewrite required with the sample data it will be simplest to leave it as is.

Team Meeting Minutes

Wednesday 24/7/24 Team Meeting 7pm

Attendees:
Alex

Oliver
Raphael
Calum

Minutes:
sent approval to industry partner for work agreement contract
sent industry partner feedback form for our meeting today
Summarised meeting to member who missed it
- lacking minutes because we forgot and can't get the recording

Team meeting set on 10am sunday to finish tutor prep and last touches

discusses stuff that needs to be done

Raphael
- make the spreadsheet workable by fixing the css

Calum
- list view tabs and multiple columns

Alex
- algorithm
- input handle data properly and num teams to projects with input field

Oliver
- input handle data properly
- editing data
- navigation
- error handling

Sunday 28/7/24 Team Meeting 11am

Attendees:
Oliver Pinel
Calum Ingram
Raphael Fahey

Agenda:
Testing on Thursday preparation
Tutor meeting on Wednesday preparation

Minutes:
Check of progress on assigned tasks
- no one done but making progress
- all coding work done by Tuesday, give Oliver Wednesday to fix up anything before testing on Thursday
Covered all the implementation we want to
Oliver will go in to host laptop webserver - recorded
Tutor meeting
- overview of the required documents

- some done
- still need IP meeting minutes, risk assessment
- sprint plan has been thrown out
- team agreement draft submitted, will be confirmed or modified on wednesday

Industry Partner Meeting Minutes

Risk Assessment

Risk	Plans
Privacy of data	All team members have signed a confidentiality agreement with the university. In addition, we will never have the raw student data ourselves accessible in spreadsheet form. Our system is a local server, which performs all storage and calculations locally, so our computers will never have it. The only time we will see the data is in recordings of testing
Covid	Our team is fully capable to continue working solely online, as we have been doing for a majority of the time.
Team member dropping out	We have in place a schedule of our work that we can complete. In addition, we have close communication with our industry partner to allow us to adjust as needed quickly.

Team Agreement

As a team, we took the time to discuss the issues mentioned in Report 1.2, such as the delayed and substandard work completed. Through this discussion we defined the expectations of our team, noting that communication is the most essential, above even getting work done. Hopefully, with this new agreement we formulated together, we can avoid any future issues, or at least have a common understanding as to how we will complete this semester together.

Goals for this Week

For the upcoming week, we will be meeting with our industry partner to determine how the testing will work, and to confirm it is happening, as this is still somewhat ambiguous due to the nature of their timetable. In addition, to the leadup of this we will be finalising any code. Once the testing is done, we will begin work on Report 2.0, which will outline the system in detail, analyse the results of the recorded testing and determine our plans for the rest of the semester, whether that is a redesign or improvements, in the form of sprints, user stories and other agile artefacts.