Project Requirement Form

Project title

|  |
| --- |
| Module Delivery Plans: Assessment Data Collection and Visualisation |

Customer name

|  |
| --- |
| The University of Sheffield, Department of Computer Science |

Project advisor

|  |
| --- |
| ?? |

Project description

|  |
| --- |
| Every year, the Learning and Teaching Manager (LTM) sends round a Google Form to collect details of every assessment academics plan to hold for their modules. Academics select their module and which semesters it will run in (1, 2 or both). They then enter details of each assessment – the kind of assessment it is, what its name is, how much it is worth and when it will take place.  Once the LTM has chased all of the academics for their responses, year tutors check the assessment dates for bunching. If there are lots of assessments happening in a short space of time, the year tutors will try to move them so that student workload isn’t excessive.  Once this is done, the LTM turns the data into a list of assessments per module and prints this off as a booklet – this hard copy is the “single source of truth”, and any changes throughout the year (which are supposed to be kept at a minimum) are noted in this booklet when they have been approved.  This process is hugely manual and very user-unfriendly. The Google Form sent to academics does not provide adequate validation, so the LTM has to clean the information after it is collected. The LTM must manually find missing entries and chase academics for their responses by email. Finding bunching is a case of looking down a list of dates, and could be much easier if it were visualised somehow.  Finally, year tutors would benefit from new insights, such as determining how many students might be affected by suspected bunching (by checking against which students take which modules) and seeing how much bunched assessments are worth (three 2.5% quizzes from three 10-credit modules in the same week would be less of an issue than two 50% assignments from two well-subscribed 20-credit modules due on the same day). |

Features list

|  |
| --- |
| What are the key features of the software? Please list at least 8   * Collect responses from academics about the assessments they have planned   + Who is teaching the modules   + When do the modules run (year-long, semester 1, semester 2)   + What assessments are planned     - What format (MOLE quiz, assignment, presentation, assessed lab, etc.)     - Title     - What percentage of the whole module it is worth     - When it will be released (if applicable)     - When it will be handed in or submitted * Validate the content of fields   + All assessments in a given module must add up to 100%   + Assessments (except exams) cannot be planned within an exam period   + Hand-in dates must not be on weekends or public holidays   + Hand-in dates must be within the academic year (sometimes dates are entered for the previous year, for example)   + Academic-year module assessments can span two semesters but others cannot   + In the case of assignments, the hand-in date must come after the hand-out date * **Produce a list of all assessments by module** * Produce an A4 “booklet” of module information:   + Academic lead   + Which semesters it runs on   + What each assessment is and when it happens * **Display a timeline of all assessments** * Display a heat map of quiz, assessed lab and hand-in dates * Allow users (like the LTM and year tutors) to filter the above visualisations by level (first year, PGT, etc.) and by modules   + It would be extra useful if these visualisations could be adjusted by how many people they affect * Automatically prompt (by email) those academics who have not submitted their responses within a given timeframe (to be set by the LTM). |

Application users

|  |
| --- |
| Who are the primary users of the application?  For management and visualisation: COM teaching support staff (administrators) and year tutors. For data entry: academics leading or teaching modules in COM. |

Programming language required

|  |
| --- |
| What programming language does the customer require students to use?  No requirement, but Ruby, Python or PHP would be ideal. If Java is all they know, then Java would be fine. |

Database required

|  |
| --- |
| Does the system require a database?  What database does the customer require students to use?  Flat-file CSV or MySQL (the latter preferred), unless they have a creative reason to use something else ☺ |

Data provided

|  |
| --- |
| What data (or document) is the customer providing?  I can provide an example set of MDP data for this year, and a copy of the Google Form we use to collect it. |

Out of scope

|  |
| --- |
| What is considered out of scope?  Not sure |