

## Appendix B - e-Administration Requirements Specification

### Introduction and Scope

The e-Admin project is a collaboration between CARET and the faculties of Engineering and Physics in Cambridge University. The project is funded by JISC under the Institutional Innovation programme<sup>1</sup>

The project aimed to develop two software modules already produced or specified by members of staff in Engineering and Physics, so that they could be adapted and used by other departments at Cambridge and beyond.

The **teaching duty allocations** module was based on an existing system developed in the Department of Engineering by Prof. Richard Prager. The **student choices** module was, at the outset, partially specified as a combination of several functions already present in course management systems at both the Departments of Physics and Engineering. In both cases, existing implementation served as an effective specification of requirements. The present document is a record of those requirements and is intended for anyone contemplating the provision of similar functionality.

For more information of the software including a discussion of the software development approach readers are referred to the final report of the e-Administration of Teaching project.

### Shared Requirements

1. All involved departments have existing systems for teaching duty allocation and student enrolment. The modules must support these systems, not impose their own.
2. Precursor systems represent significant development and testing effort, and should be allowed to serve as specifications by example, both for process and UI.
3. The sustainability plan called for Departments adopting the software to take full subsequent responsibility for it. Computer officers taking on this duty are unlikely to be professional developers and the software design must take this into account.
4. Deployment should be swift and simple, not requiring expert knowledge on the part of the users.
5. Performance should be quick, i.e. less than 0.5s page load for typical actions in the web application, even for a large department.
6. Data schemas should be as simple and modular as possible, to ease maintainability.
7. Existing functionality in each department should be preserved and easy to get to.
8. Extensive reporting facilities should accommodate the various types of report used by each department.

### Teaching Duties Allocation

1. The module should be a database driven web application, with three main tables: jobs, staff and curriculum elements.
2. Referential integrity should not be enforced in the database tables. Problems should be flagged for attention, and queries should use 'fuzzy' logic as far as reasonable.

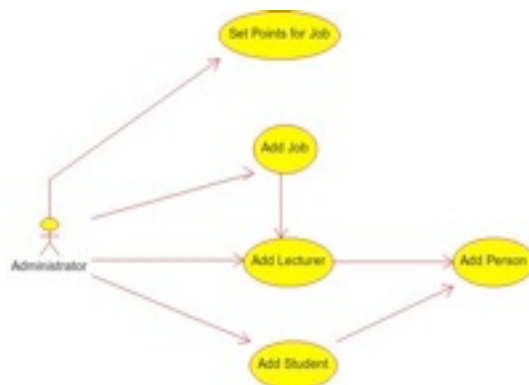
---

<sup>1</sup> <http://www.jisc.ac.uk/whatwedo/programmes/institutionalinnovation.aspx>

3. Users should be able to enter jobs, curriculum elements and staff lists by CSV file upload, or enter or edit them one by one in web views.
4. Jobs should be typed, and each type have an associated point score representing expected workload
5. Provision should be made for restricting access to data that are sensitive, e.g., the teaching load of people other than the user. Anyone should be able to search by curriculum element to see who is responsible for it. Staff should have a specific access policy, determined by departmental heads as appropriate to their context. Jobs and corresponding points assignments may be private to management and the individual concerned, or shared within the department. Administrators should have full access and be able to assign access levels.
6. The web interface should give the user options to run a selection of queries useful in their context, and present the results in report format, e.g. points assigned per curriculum element, points assigned per departmental sub-group, points assigned per staff member, jobs per staff member.
7. Provision should be made for instances in different departments to query eachother's databases, for the purposes of sharing staff.
8. A mechanism should be provided for archiving (and subsequently viewing) allocations for previous years

Generic use cases are given below. They capture the main activities covered by the teaching duties allocation module, rather than trying to give an exact description.

**Data entry:** various data entry tasks performed by the administrator



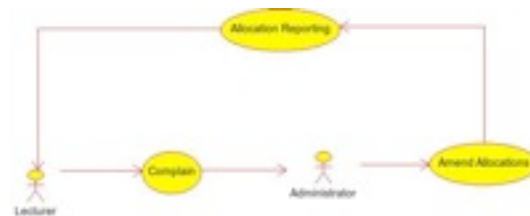
**Iterate Allocations:** make allocations, receive feedback from teaching committee and amend



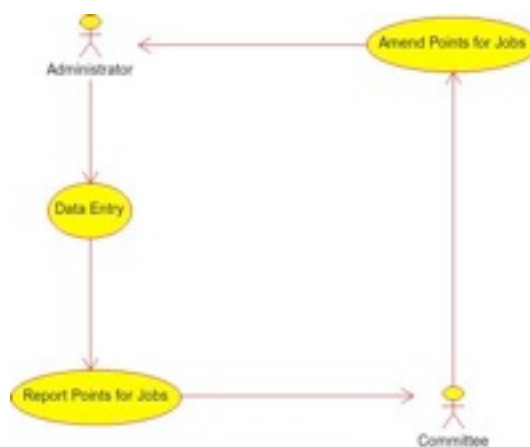
**Allocation Reporting:** produce reports of teaching allocations



**Special pleading:** individual lecturers request changes to allocations



**Points for Jobs:** assignment of points for new or changed kinds of jobs



## Student Choices

1. The module should be a database driven web application, with two main tables: examination papers and students. Both will have at least unique key values, preferably taken batch-wise from the University student information system. The student table should also contain information about programme registrations and previously-taken papers, for checking of pre-requisites, and, for Cambridge, their College affiliation. A mechanism for defining rules about allowed and necessary combinations of papers, and checking student selections against them, should also be provided.
2. A mechanism for administrators importing lists of students as CSV files (usually sourced from the student information system).
3. Students wishing to register their choices but who find they are unknown to the system should be given information on who to contact to resolve the problem.
4. The module should allow initial student choices as starting values, up to an early deadline. Later modifications made in the department should be incorporated into a final report made available to the examinations board (or local equivalent responsible for HEFCE returns) for final approval.
5. A mechanism should be provided for importing lists of papers as CSV files, or rolling a previous year's list forward. Papers should be editable one by one within the web application.
6. A mechanism for exporting examination registrations to CamSIS should be provided.
7. Sensitive data, e.g., student grades, should be secure and have restricted access.
8. Students should be able to log in and see registration deadlines and previously-made choices. They should be able to change and make choices which are still open to them. The system should remind them of approaching deadlines by email, and should provide appropriate guidance onscreen if a student wants to know what combinations are necessary or permissible, or if a student has made an impermissible or non-recommended selection.
9. Administrators should be able to see at least lists of students, papers, students enrolled for each paper, students choosing project options, students with incomplete or non-recommended selections, numbers of students enrolled for each paper, selection rules, and deadlines. Each view should allow manual editing and export as CSV file. If a student's choices are edited, the administrator should be given the option to notify the student by email.
10. Emails from the system should contain text explaining that they come from internal Departmental processes, not the central SIS.

The use context addresses by Student Choices is as follows. The student registers for exams (with central University administration) separately from registering for courses (with the Departments running them). Departments need to know what courses students intend to take, so that they can arrange lab sessions and supervisions, and contact them in case of changes. Departments with large student cohorts have particular need of IT support for these processes. The University's central administration meanwhile needs to know what examinations students plan to take, so that they can report to HEFCE and arrange examination venues. For some year groups in some departments, the department submits 'generic' examination registrations on behalf of students, and arranges examination venues internally - this is usually done to reconcile the need to report to HEFCE with a curriculum structure which allows choices after the HEFCE reporting deadline, however the central SIS still needs exact transcript information. The Student Choices module facilitates the actions carried out by the student and the department.