Jimmy Peleha (12230830)

Requirements Notes

Here are some rough notes I’ve made in class. I’ve extrapolated upon what I understand.

Required:

A marking and mark management system that provides a simplified and convenient interface for lecturers to allocate marks, markers to award and record marks and students to view and query their marks. This system will use a database to accumulate and store the marks of every student within the CS department. This incorporates all courses and modules available to the students. The emphasis is on the management and design of this database because the rest of the system concerns actions on this database. It provides multiple interfaces through which designated users may interact with this database.

Firstly, we’ll use the university’s Computer Science database to import information and create our own database. This can be done through the use of csv documents for efficiency. The csv documents need not be parsed to capture the data; there are predefined tools available to developers and designers to do so.

In the design of the database, privacy and access settings need to be modified. For example, students should not have rights to make any changes to their marks on the database.

The lecturer should be able to specify the mark allocation for individual tasks in every course. This means that every assignment/practical/test for every module should be able to have a unique number of questions and a variable amount of marks for each question. This can be achieved through refining entities within the database. This avoids having entities with a fixed number of columns for marks that could be.

All the changes within the database should be accounted for by markers and lecturers. This requires the use of audit trails, reports. The interface should incorporate a field in which the marker can specify the reason for any modifications on the database whatsoever.

The Web interface

* Must use Jango and Python
* Must be secure
* Must generate audit trails
* Must be testable
* Must be scalable such that speed will not be an issue
* Aldap?
* After updating the database, the server returns immediately after ussuing the UPDATE command

I also wrote down “daily batch systems”. I can’t remember why.

We have to use UML diagrams to model the system:

* Use use-case diagrams to illustrate what happens when the user interacts with the system
* Abstract models for scoping.