Handin

Handin is a python-based student grading program. Users of Handin are Lecturers, Students and the System Administrator.

System users:

System Administrator: Creates Modules upon request from lecturers and set the access permissions for that module.

Responsibilities:

* Retrieve information needed to create modules, which is module name and code, and the academic year and semester the module will be held in.
* Grant access to the lecturer of the modules and teaching assistants that will be aiding in assignment creation or grading.
* Ensure students of the module cannot access module information and assignment details. (class list, grades, solution, testing commands)

Use cases:

* As I system administrator, I want to create modules, so that lecturers can manage them.
* As I system administrator, I want to ensure module access is secure, so that assessments are fair.

Lecturer: Creates assessment for their modules to be undertaken by their students.

Responsibilities:

* Create a class list for the students who will be assessed.
* Create a defaults file with opening and closing dates for assessments, number of retries allowed, late submission closing date, and penalty for late submissions.
* Create an instructions file with the commands to be run to test the students submission and the awarded points relating to each command.

Use cases:

* As a lecturer, I want to set assessments for my students, so that I can monitor their progress.
* As a lecturer, I want to have my students assessments graded automatically, so that my workload is reduced.

Student: Undertakes assessment set by their lecturer and submits it through Handin.

Responsibilities:

* Undertake assessment within the assessment time frame.
* Submit the assessment for grading.

Use cases:

* As a student, I want to get instant results on my assessments, so that I know how I did and whether I should try again.

Handin functions:

Creating a module:

Only a system administrator can create a module.

Run h4a.py and click button “Create New Module”.

Then enter module code, academic year and semester for the new module.

Click “OK”.

In the .handin directory a /<module code>/ directory will be automatically created.

Creating module definitions:

After a module has been created by the system administrator, the lecturer must enter the definition for that module.

Run h4l.py and click “Module Definitions”.

Fill in the date for Monday of week01 and the time, open date, closing date and cut off date for assessments. Handin will automatically change these dates to reflect the following weeks.

Click “OK”.

These can later be found in definitions.yaml in the directory .handin/<module code>/<academic year>/

Creating assessments:

Submitting assessments:

When an assessment is due, the student will run the Handin thin client with a parameter containing that assessment code e.g. w01, and the location of the file.

Handin will search for the file for that week in the expected directory and run it using the commands specified by the lecturer when creating the assessment.

Grading assessments:

When a lecturer creates an assessment, they must include a parameter file with the commands to be run of the students submission.

They must associate a grade with each command and if the command output is as expected, the student is awarded those points.

The points are displayed on the students machine in the form:

Step 1: compiling program 2/2.

The total of marks awarded can be found by running h4l.py and clicking manage student marks.

Access granting:

Two different accesses need: lecturers access to modules to alter module information, create assessments and check students grades. Students access to modules to submit assessment only, shouldn’t be ble to see any module information.

Lecturers access: when creating the module the system administrator includes a permissions file with the user names of people allowed to acces the modules information.

Students access: upon registration for a module, students will receive a thin client with their student id’s and module code hard coded into it. They will then run this thin client every week with the week number as a parameter.

Storing data:

Date will all be stored in the directory .handin.

Each module will have its own directory /<module code> where all data relating to that module will be stored there.

Data that will need to be stored for each module in the directory /<module code>

* Academics years for when that module was taken.

Data that will need to be stored for each module academic year in the directory / <module code>/<academic year>/

* Permissions (who can alter module information).
* Class list (names and id’s of students who will be taking the modules that semester)
* A definitions file. (dates relating to assessment submission)
* A directory for each assessment.
* A students directory.

Data the will need to be stored for each assessment in the directory /<module code>/<academic year>/<assessment code>

* A parameter file. (commands to be run for this assessment and the grades relating to it)

Data that will need to be stored for students in the directory /<module code>/<academic year>/students

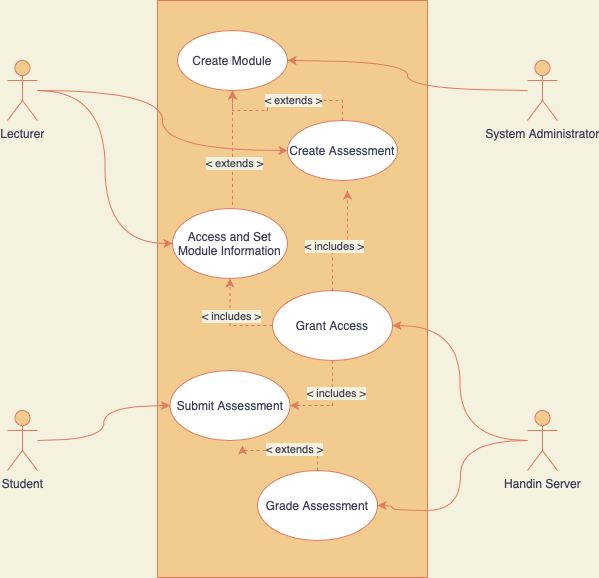
* A directory for each student created with their student id.

Data that will need to be stored for each student in the directory /<module code>/<academic year>/students/<student id>

* + A directory for each assessment the took.

Data that will need to be stored for each student in the directory /<module code>/<academic year>/students/<student id>/<assessment code>

* + - A copy of the students submission.
    - A variables file. (contains the date and time the student submitted their assessment, attempts, the marks awarded)

Use Case diagram. 

Handin sub-directory system

