Revised Problem Statement

Grade Management System

The primary purpose of the system is to input, store and edit grades for student’s homework, projects assignments and tests (mid / final).

The system to be built should allow the professor (privileged user) and his TAs (non-privileged users) to

a) create a database for every class e.g. one for CS328;
b) every test consists of several assignments which are graded on a scale of 0 .. 100 points.
c) every test is to be entered on a separate Excel sheet - this will be the only input of student scores into the grading system.
d) for every student and for each assignment the points (0..100) for each individual problem, are entered into the Excel sheet;
e) points range typically from 0 (not done at all) through 75 (essentially correct) to 100 (absolutely correct), but bonus points for e.g. an exceptionally good solution are also possible, raising the upper limit to, say 110 or even 120; it might be a good idea the do plausibility checks on these fields and flag non-numeric entries or values below zero, and to warn when values above 125 are found.
f) it must be possible to access the entries in the database for selective updating i.e. entering selective changes to individual records - but only through the one and only Excel sheet.
g) the professor can run a “grading” program which computes the total score for each assignment/test and for the whole course so far.
h) the professor can request a printout of the grades plus a statistical evaluation of the course.

The Excel sheet shall permit the user to enter the name, id-number and scores for each test or assignment and each student; the user must be able to change these values before they are used for the grade computation. Also in this Excel sheet shall be one entry for the “Reference Candidate” who sets the required score value for an A; this will usually be 100, but it could also be lower (e.g. 85) or even higher (e.g. 110).

Every class is to be treated as a separate database (file). Every test (e.g. mid term, final, homework assignments) are to be treated as separate files; the final score (grade) for the class is computed from the results of these individual test, following the same calculation outlined below.

All entries should be marked with a time stamp and a sequence number for documentation purposes. Access to individual records based on the name or the sequence number and subse-
quent updating and editing must be possible. This functionality must use its own graphical user interface and must allow to display individual entries selectively as well as collectively i.e. the whole test. Editing is basically changing individual fields of the retrieved record and subsequently writing this information back to the mass storage.

This output report generator is controled by its own graphical user interface and allows to run a “grading process” which computes grades from the individual scores. This must be possible for individual tests per student and for all students and collectively for all tests and all students. The scores have to be displayed on the screen and also entered in a file formatted for subsequent printing.

Outline of grade calculation:
every test is rated with 0 .. 100 points $p_i$.
every test $[i]$ also has a “weight” factor of $w_i$ indicating the difficulty.
the sum over all $i$ weighted tests (the score) is calculated: $(\sum_i (w_i \cdot p_i))$

the score of the Reference Candidate (RC) sets the 100% upper limit, i.e. the individual student scores will range from 0 .. 100% of the reference score determined by the RC.

Every test is calculated separately from the individual problems (referred to as assignments) and will result in one final column with the individual intermediate scores.
The final grade is computed from these intermediate scores using exactly the same formula as above, permitting to set e.g. the weight of the final test to 25%, the midterm to 20%, the homeworks to 20% and the project to e.g. 35% (must add up to 100, though).