# Grade prediction and performance application

Part A – Requirements

## Functional requirements summary

<R1>The software allows the user to build a course.</R1>

<description>The user is able to build a course in order to find out the outcome of his degree, real or by predicting marks.</description>

<R2>The software shall allow the user to record the name of the course.</R2>

<description>The student is able to add the course name. The user should be able to change the name of the course, for example, when a student decides to switch to a different course.</description>

<R3>The software shall allow the user to add modules to the course.</R3>

<description>The student can create modules and add them to the course.</description>

<R4>The software shall allow the user to add a credit value to each module.</R4>

<description>Each module has a credit value. Most of them are 15 credits but there are also modules with 30 credits. There is also a final year project worth 45 credits. Therefore, the user should be able to add a credit value to the each module.</description>

<R5>The software shall allow the user to input assessments(tutorial, coursework, exam) for each module.</R5>

<description>The user can add assessments to each module, including a tutorial exercise, coursework, exam et al. Each assessment has name and weight.</description>

<R6>The software should show the new module added.</R6>

<description>After each module entry, if everything is correct, the module will be shown in the software. However, some information can be deleted in case some of errors. For example if the assessments do not add up to 100% or invalid weight for assessments.</description>

<R7>The software shall allow the user to add modules until the course is complete.</R7>

<description>A course should be complete when the sum of 120 credits per year has been reached in all three years of study.</description>

<R8>The software shall allow the user to save the status of the course.</R8>

<description>The user should be able to save manually the status of the course using a save button. The user can also save automatically any changes on exit. A dialog should appear when exiting and asking if the user wants to save any changes.</description>

<R9>The software shall continue from the last edit version.</R9>

<description>If the software is not at its first run, then it should use the existing data to build the interface and resume to the last edit stage.</description>

<R10>The software shall save the file changes in the background.</R10>

<description>The software will save changes in the background to simulate writing to a database.</description>

<R11>The software shall update any changes in the summary tab (level summary and main one) as the user edits any module details for the purpose of predicting the outcome of his degree.</R11>

<description></description>

<R12>The software shall allow the user to add modules for 3 years of study.</R12>

<description>The user can add modules for 3 years of study. Each module will have a credit amount. All modules should add to 120 credits per year.</description>

<R13>The software shall show results on each level summary.</R13>

<description>On a level summary the user can see the outcome of each module. For example: User Experience Design: 45% Pass or 35% Referral or 25% Fail.</description>

<R14>The software shall have a main Summary tab where the user can see the final outcome for each level of study and the final outcome of their degree.</R14>

<description>The main summary tab will cover all three levels of study with each percentage and the final outcome for the user’s degree. For example: level 4: 44%, level 5: 60% , level 6: 75%. Outcome: second upper class degree.</description>

<R15>The software shall allow the user to delete a course.</R15>

<description>The user is able to delete a course. The user might want to change the course. Or to create the final outcome for a friends’ degree without deleting his own data.</description>

<R16>The software shall allow the user to add the name of the student taking the course.</R16>

<description>The student or a friend of the student might want to find the outcome of their degree. They need to have a different name so that it could become distinctive on the table.</description>

<R17>The software shall allow the user to reset a year of study.</R17>

<description>When changing the course or for any other reason, the user shall be allowed to reset the year of study to make the data input easier. This will reset the year of study to 0 modules added.</description>

<R18>The software shall allow the user to delete the module and its assessments.</R18>

<description>If the student decided to drop a module, the student might want to delete the module and all it contains.</description>

<R19>The software shall allow the user to delete an assessment.</R19>

<description>The user might input data by mistake or might have added too many assessments. The user shall be allowed to delete an assessment. </description>

## Non functional requirements

<NF1>The software should not crash under any user input.</NF1>

<description>The user is asked to input lots of data. The software should be aware of the possibility of having negative numbers, instead of positive, strings instead of integers, et al.</description>

<NF2>The software should be able to recover from previous versions in case of a crash.</NF2>

<description>In case the software doesn’t manage all the errors, it should recover from the previous version without having to input all the data again.</description>

<NF3>The capacity of the app should not be limited.</NF3>

<description>The user can create more than one course entry. In this case it can be used by lecturers and university staff as well.</description>

<NF4>The application should be secure.</NF4>

<description>The application holds personal information given by the user, so it should be secure in order to conform with the data protection act.</description>

<NF5>The response time should be minimum.</NF5>

<description>The software should be optimised in order to minimize the time response.</description>

<NF6>The software should be managed easily.</NF6>

<description>The software should be easily managed by any user with no prior experience with it. Should also be intuitive.</description>