**Virtual Advisor**

This interactive application is expected to advise student to identify the optimum combination of courses and grades in order to achieve a target CGPA or to come out of the probation situation.

It’s very important to have a strategy to combat probation in the early stage of this situation. This tool could certainly provide a platform to achieve just that.

The Virtual Advisor can analyze the current state of a student grades and can project the combination of courses to be taken and the grades to be earned to achieve a target CGPA. There could be many ways to reach out to a specific goal however this application can offer the most **optimum** path initially that could require **least effort** from the student. The student can interact with the application by providing their input iteratively to determine the course and grade combination of their likings.

This analysis is almost impossible by an actual advisor to perform when dealing with hundreds of students in a very short span of time. Most importantly, the student can ask the application to display his potential CGPA when a certain course is added, retaken, excluded or when a grade is changed - any permutation and combination imaginable.

**Here are the requirements of the mobile application to be developed in Android Platform:**

Design and develop three CGPA projection/forecasting **strategies** as follows: CGPA forecasting based on:

1. Course Retake Only
2. Combination of Retake and New Course.
3. New Course Only.

Consider all courses have 3 credits and each credit can have maximum 4 grade points. Example: A 3 credit course can have maximum 3X4 = 12 grade points. (A Grade)

**User input interface requirement: Score: 20**

1. The student must choose the strategy first. These strategies will be listed in the check box. It’s mandatory to choose one strategy before the calculation starts. Score: 5
2. The user should be able to enter the target CGPA in a mandatory input field. Score:5
3. The user should be able to add new course which may be taken by the student in the future.

Score:5

1. A button to start the calculation and to find the grades to satisfy the CGPA. Score: 5
2. The student should be able to see their current actual grades by hitting a link. This will help them check their current situation before using with the application Score: 5

**Output interface Requirement:** **Score: 40**

1. Display the target CGPA at the very beginning. The field is editable. Score: 5
2. The course and the target projected grade should be displayed side by side in a row. There will be a new row for each course and grade combination being displayed. The list should display only those courses which are new (not completed yet by the student) or when the grades are changed for the existing courses to satisfy the target CGPA. Score: 5
3. The grades should be editable and the row should be delete-able for each row of record. The user should be able to change the grades proposed by the system and have it rerun the calculation again. When the user modifies the grade, the system must not change it during calculation. Score: 10

1. An add button at the end of the display so that the student can bring new course in the mix optionally with a potential grade. Score: 5
2. The user must be able to choose one of the three strategies explained earlier in this page also.
3. A calculate button to start the calculation again. Display message if the target CGPA cannot be achieved. Score: 5
4. There could be multiple paths or projection to reach out to a target CGPA. There should be a button to view the rest of the CGPA projections/paths if any. If not, show the message saying that there is no other projection available. Score: 10

Processing Requirement: **Score: 25**

1. Retake is prohibited on a course which has already B+ or more. Score: 5
2. Retake strategy is applicable when the student has already completed some courses: Score: 5
3. The forecasted grade must be the minimum to satisfy the CGPA targeted at the beginning of the process.
4. There are potentially multiple paths to reach to a target CGPA. The application should be able to keep all those combinations in the memory so that they can be traversed when toggled by the user in the front end. Score: 15
5. Maximum grade point of 3 credit course is 12 (A)
6. Minimum grade point of 3 credit course is 0 (F)
7. A combination of more course with average low grade is preferable than the combination of less course but higher average grade.

Application Input:

1. The application should be provided with the following input from the database:

student number, semester, course code, course title, grade, credit etc. Score: 15

The participants should create the database at the beginning and design and populate the tables with random student and data.

Platform Requirements:

1. Should be a mobile application developed using native platform.
2. It is essentially developed by using android platform - Android Studio.
3. Testing can be done with Android Studio emulator or actual devices

Deliverable:

1. The complete executable (apk) with data source must be uploaded to a Google shared folder. (Will be provided later)
2. Export all source code to a file called: teamname.zip Need to make sure that source code can be imported to Android studio.
3. A readme file to run and to test the application by the judges later.

Evaluation Criteria:

1. The APK must be installable and runnable in the emulator preferably in Android studio.
2. Ten finalists will be selected from the initial evaluation of the executable and the maximum number scored
3. Team will be called to explain their developed code and the strategy taken
4. The application will also be evaluated based on the following principals:

* User friendly UI
* Overall architecture of the application. Is it layered? (MVC). Separation of concerns etc.
* Is there any convention used to design the various classes, methods, constants, packages etc.
* Database design. Normalized. Why or why not.
* How easy to add a widget or a feature in the application? Easily maintainable, extendable
* The degree of Object Oriented methodology used.

Hint: Calculate the deficit grade point first and then start distributing the deficit points across the old and new courses to satisfy the required CGPA.

Grade and Grade Points distribution:

|  |  |
| --- | --- |
| A | 4.0 |
| A- | 3.7 |
| B+ | 3.3 |
| B- | 2.7 |
| C+ | 2.3 |
| C | 2.0 |
| C- | 1.7 |
| D+ | 1.3 |
| D | 1.0 |
| F | 0.0 |
| I | 0.0 |