LASSONDE SCHOOL OF ENGINEERING | YORK U

EECS 2311 Z

Evalio

Group 11

Shivam Gupta, 219923309

Shadi Karimpour, 218183525

Himanshi Verma, 219830306

Bardiya Ameri, 220108684

Rima Ramcharan, 219522986

Kartik Sharma, 220760146

Evalio Vision Statement

Summary:

Evalio is a web app that models real course grading rules. A student uploads a course outline; the system extracts assessments, weights and rules. The student reviews/edits the structure for accuracy. As grades are entered, Evalio recalculates feasibility for a target grade and the minimum scores needed on remaining assessments.

Project Description:

Once multiple tasks are submitted, understanding their impact on an overall score often causes confusion among students. Because course guidelines may assign varied importance, drop the weakest result, select the best outcomes within groups, or require passing thresholds, clarity fades quickly. Standard calculators do not handle such complexity well, relying solely on averages without flexibility. As new grades appear over time, uncertainty rises. Through Evalio, individuals build custom evaluation models matching real class designs. With every entry, current standings refresh instantly, maintaining accurate insight.

Target Users:

Far from being designed with specialists in mind, Evalio supports students in universities who want to see progress more clearly. Through clean layouts, results appear without reliance on grading formulas or statistical knowledge.

Value and Benefits:

At the start of the period, Evalio provides straightforward information on student progress. Instead of guessing outcomes, students get meaningful reports linked to actual work. Because responses come quickly, development is simpler to follow. This method permits corrections early, preventing issues from expanding. With steady data movement, meaningful updates become possible. Understand their current standing in a course, check whether a target grade is achievable and plan realistic strategies for remaining assessments with grading rules made clear via Evalio. Uncertainty fades as students rely on comprehension when choosing paths.

Success Criteria:

Evalio will be considered successful if students can clearly understand how their final grade is calculated. Learner growth is observed carefully and measured without interruption over time. Finding out whether the target grade is achievable hinges on how well work has gone so far, alongside what lies ahead in upcoming evaluations. Performance up to now sets a foundation, while future tasks bring potential shifts. As objectives appear achievable, assurance naturally increases. Goals take form based on immediate priorities. Clarity shapes how users view their academic standing after working with Evalio. When data moves freely, oversight follows naturally. Uncertainty lifts - so does tension tied to results, with understanding comes responsibility.

Big User Stories
Group 11: Evalio

Course Evaluation Modeling

As a student, I want to upload a course outline and build a grading structure that matches my real course rules, so that my progress is calculated accurately.

Priority: High

Cost: 8 days

Track Progress and Grade Feasibility

As a student, I want to enter my completed grades and see whether my target final grade is achievable, so that I can understand my current standing.

Priority: High

Cost: 7 days

Big User Stories
Group 11: Evalio

Minimum Score Requirement Calculation

As a student, I want to see the minimum scores I need on remaining assessments based on my course rules, so that I can plan realistically for future work.

Priority: High

Cost: 8 days

What-If Scenario and Risk Analysis

As a student, I want to simulate different grade outcomes and see safe and risky score ranges, so that I can make informed academic decisions.

Priority: Medium

Cost: 7 days

Evalio - Iteration 1 User Stories

EECS 2311 - ITR1 (6 detailed stories)

ITR1-1: Manual Course Setup (Assessments + Weights)

As a student

I want to create a course and manually enter assessment components and weights

So that I can start planning even before outline extraction is implemented

Details

- Create course (course name; term optional).
- Add assessments with: name and weight (%).
- Show running total weight while editing.
- Block invalid inputs (empty assessment name, negative weights).

Acceptance criteria

- User can create a course and add at least 1 assessment.
- System displays total weight and updates it immediately after edits.
- Invalid values are prevented and user sees a clear error message.
- Course can be saved in stub storage (or in-memory) for ITR1.

Priority: High **Story points (ideal dev days):** 2

Open questions

- Do we allow bonus components in ITR1 or leave for ITR2?

ITR1-2: Edit & Validate Grading Structure

As a student

I want to edit, delete, and reorder assessments with weight validation feedback

So that I can correct mistakes and keep my grading structure consistent

Details

- Edit assessment name/weight; delete assessments.
- Optional: reorder assessments for a clean UI.
- Validation: warn if total weight is not 100% (but allow saving in ITR1).

Acceptance criteria

- User can edit/delete assessments and see updates immediately.
- System warns (non-blocking) if total weight is not 100%.
- System blocks negative weights and empty names.
- No crashes when assessments are removed or reordered.

Priority: High **Story points (ideal dev days):** 1.5

Open questions

- Should saving be blocked when total weight != 100% (or only warn) in ITR1?

ITR1-3: Grade Entry + Current Standing Dashboard

As a student

I want to enter received grades and view my current standing

So that I can understand where I am before planning for a target grade

Details

• For each assessment, user can enter a percentage score (0-100) (keep ITR1 simple).

• Treat blank as 'not graded yet' (not counted).

• Dashboard summary shows: completed weight, remaining weight, and grade-to-date.

Acceptance criteria

• User can enter/update grades for any assessment.

• Grade-to-date uses only completed components (ungraded excluded).

• Remaining weight is shown clearly.

• Inputs validated within 0-100.

Priority: High **Story points (ideal dev days):** 2

Open questions

• In ITR1, do we support points (e.g., 17/20) or percentage only?

ITR1-4: Target Grade Feasibility + Explanation (MVP)

As a student

I want to set a target final grade and see if it is achievable, risky, or impossible

So that I can avoid unrealistic targets and adjust my plan early

Details

- User enters target final grade (e.g., 80%).
- Compute required remaining average based on completed contribution and remaining weight.
- Display label + explanation: 'You need X% average on remaining Y%.'

Acceptance criteria

- System shows Achievable/Risky/Impossible with a short explanation.
- Updates instantly when grades or target changes.
- Handles edge case where remaining weight is 0 (course complete).

Priority: High **Story points (ideal dev days):** 1.5

Open questions

- What threshold defines 'Risky' for ITR1 (e.g., required >= 90%)?

ITR1-5: Minimum Required Score Calculator (Remaining Average)

As a student

I want to see the minimum average score needed on remaining work to hit my target

So that I can plan the level of performance I must maintain from now on

Details

- MVP output: a single number for required remaining average (%).
- If required remaining average > 100%, mark as impossible and explain why.
- Keep rule handling simple in ITR1 (no best-of/drop rules yet).

Acceptance criteria

- System outputs required remaining average for the current target.
- If impossible, system clearly states that 100% on remaining is not enough.
- Recomputes immediately after any grade/weight/target change.

Priority: High **Story points (ideal dev days):** 2

Open questions

- Later, should we compute per-assessment minimums or keep overall remaining average as the main view?

ITR1-6: What-If Scenario (Single Scenario MVP) + Save Snapshot

As a student

I want to test a simple what-if scenario and save it

So that I can compare different plans and see how feasibility changes

Details

• User enters hypothetical scores for remaining assessments OR a single 'remaining average' field (choose MVP).

• System recalculates projected final grade + feasibility + required minimum.

• Save one named scenario snapshot using stub persistence (real DB later).

Acceptance criteria

• User can create a scenario, view results, and save it.

• Saved scenario can be reloaded (within ITR1 stub storage).

• Scenario results are consistent with the same calculation logic as the main view.

Priority: Medium **Story points (ideal dev days):** 2

Open questions

• For ITR1, should scenario input be per-assessment hypotheticals or a single 'remaining average' input?