CSRS - Customizable assessment forms

Engineering Design Review

Author: Diego Sebastian Garcia Minjares
Date: 30 October 2024
Status: Pending

Introduction

Feedback is the core of any student learning path, as it allows students to learn from their mistakes and improve future work. According to university protocols, every student is entitled to receive feedback for every submitted assignment. Currently, the CSRS feedback system is limited: markers only have two feedback options, a numeric grade and a single free-form text field. This limited approach prevents the markers from offering a detailed, organized feedback that could help students improve using it as a basis. Markers have a hard time expressing being specific, and students receive feedback as a single text covering an entire assignment, making understanding feedback aimed at specific sections more confusing. As a result, students may ignore crucial advice, limiting the purpose and effectiveness of feedback as a tool for academic success.

This document proposes the creation of a *enhanced feedback system* within the CSRS ecosystem, providing module organizers two choices regarding feedback customization for assignments:

- **Assignment-Specific Forms**: Creation of custom feedback forms specifically for each assignment.
- **Reusable Templates:** Option of saving and reusing feedback forms on similar assignments, saving time and simplifying feedback due to similar structures, resulting in consistency.

This feature will enable multi-sectioned feedback, including instructions and multiple free-form text fields that cover different aspects of the feedback, component-based numeric marks, and tagging options for both markers and module organisers. This feature goes hand in hand with CSRS objective of aiding students improve academically.

Goals and non-goals

- **Goal:** Enable module organisers to create detailed and structured feedback forms within multiple sections, component-based numeric marks, and multiple free-form text fields that cover different aspects of the feedback.
- **Goal:** Ability to label submissions with tags, allowing to categorize submissions and filter them.
- **Goal**: Save feedback forms as templates, adding the possibility of reapplying to different assignments.
- **Non-goal:** The system will not create any automated grading or feedback in any form.

Design overview

Enhanced feedback system adds functionality to CSRS, this enables the creation of custom feedback forms that can also be used as reusable templates for assignments. This requires new data structures that correctly integrate with the existing CSRS to guarantee the correct application of custom forms and templates.

The new feature will consist of four main changes: module organisers will have the option of adding a tag and creating the multi-sectioned feedback, forms. When grading, markers will be able to see all the relevant feedback for the assignment and will be able to filter submissions using tags. Finally students will be able to see a rubric with all the information from the feedback when checking their graded assignments.

Current system

The existing system provide markers with only two fields for each submission, a free-form text field and a numeric input. Feedback data is stored in submitted_assignments table which currently lacks the means for sectioned feedback, tagging and reusable templates. This are the core tables which need changes for the new feature implementation:

Table	Purpose	Key Fields	Limitations
assignments	Stores information created y the module organisers.	assignment_id, title, description, created_by, module_id, due_date, start_date	Hasn’t connections to any kind of structured feedback form or template.
submitted_assignments	Has information about each student submission given an assignment.	-submission_id, assignment_id, student_id, marker_id, grade, feedback,late	Has a single grade restriction and doesn’t support detailed sectioned feedback nor tags.

New tables

Enhanced feedback system needs the creation of new tables, and the extension of the above tables.

- A table that stores the configuration for each of the custom feedback and is linked to an assignment is needed. A JSON based structure will be used to store the correct structure of the feedback form. This includes text-fields, instructions and sections within it.
- A table that manages ands stores reusable templates that module organisers can later apply across other assignments. Each template is scored with an unique identifier, a name, timestamps and a JSON based structre as well.
- A new table linked to *submitted_assignments* needs to be created to store the feedback entries of each student. This table will also use a JSON based structure in one of its attributes as well to store the data from the feedback form.
- Tags will be stored in a separate table that can later be linked to submitted_assignments, this will allow markers to filter graded submissions.

New data and work flow

Module organisers use the costumization interface to create or select templates, this are saved on their corresponding tables. When create is mentioned I refer to both templates and individual feedback forms. *Markers* access feedback forms linked to the assignment they are currrently grading, they are able to enter scores, comments and apply tags to each of their graded assignments. *Students* will be able to see their structured feedback for each of their submitted assignments. Role based access restricts visibility so that students only have access to their feedback.

Integration with existing system

New endpoints need to be designed, implemented and tested for the development of the feature.

Category	Endpoint	Descirption
Feedback forms	/create,/update,/delete,/get	CRUD operations for feedback forms.
Templates	/create,/update,/delete,/get	CRUD operations for templates.
Tagging	/apply_tags,/get_tags	Categorizes submissions and retrieves available tags.
Feedback entries	/submit_assignment, /get_feedback	Allow markers to submit feedback and students to receive it.

In order to decrease database load, caching needs to be implemented on templates and tags.

Alternatives

Standardized feedback with limited customization

Develop universal templates from which module organisers can choose from, with limited to non existent customization.

- *Pro*: Module organisers will spend less time creating feedback forms
- Pro: Easier to implement, maintain and less storage is needed.
- *Pro:* Ensures consistency across many modules, has a more standarized approach.
- *Con:* No flexiblity when needed, decreasing quality and specificity of feedback.

Alternative template selection for assignments

Markers could be the ones selecting the feedback form template they wish for their graded assignment instead of module organisers which will result in more flexibility but in a possible lack of standardization across markers.

Relational tables instead of JSON for data storage

Use tables to store form layouts, templates, and feedback data instead of JSON fields.

- *Pro:* Complex queries are easier to implement, for further data analysis since each component has its own table, column or row.
- *Pro:* Improves data
- *Cons:* Increases complexity in database and table design, increasing maintenance and development time.
- *Cons:* Database load may increase due the new ables, affecting perfomance overall.

Milestones

1. Gather requirements from module organisers, markers and students to polish key features. Low effort prototypes also need to be created to confirm flow and interfaces within the feature. *2 weeks*
2. Creation of back-end controllers, use cases, and repositories for creating, saving and retrieving feedback forms, templates and tags. Conduct unit testing to ensure no errors and correct data flow. *4-5 weeks*
3. Allow UI/UX team to come with a design based on the results of the first milestone and develop the front-end for it, designs most be reviewed before being developed. *4-5 weeks*
4. Through testing ensure secure and efficient storage for saved templates, detailed feedback, and tags. Conduct integration testing for interaction between back-end modules, and exchange between front-end and back-end. *2 weeks*
5. Carry out end-to-end, performance and usability testing on all developed features. Upon approval, a demo should be conducted with some institutions that use the software. It's important to gather feedback for polishing and future refinements. *3-4 weeks*

Dependencies

- *API team:* Integration of new end points into CSRS existing API.
- *Test team:* Testing of new feature: end-to-end, integration, usability and unit testing.
- *Database team:* Four new tables need to be created, and some tables need to be expanded.
- *UI/UX team:* Need to design the changes to the module organisers assignment creation page, markers grading page, submissions for markers page, and student submitted assignment page.
- *Customer success team:* Talk to university to see if the feature is needed.

Cost

Increased storage requirements due to saved templates and detailed feedback data, costs are expected to be moderate, feedback data is and tags are compact but will grow overtime. Detailed feedback and tags can be erased every three years to reduce stacking deprecated information. Minimal costs are also to be expected for updates and new features related to templates and feedback forms.

Privacy and security concerns

Enhanced feedback system, doesn’t need the creation of new roles or permissions to CSRS, markers, and students as usual will have access to their feedback. It’s important to consider CSRS and universities policies on institutional record-keeping requirements, as well as to ensure that the integration of new end points follow all security guidelines, including auditing and testing procedures ensuring the detection and avoidance of any security vulnerability that may appear. It’s important to keep roles as they are, and don’t change permissions.

Risks

Risk	Mitigation(s)
Deleting or updating templates could remove importand feedback information, affecting past records.	Implement version control for the templates, ensuring that assignments aren’t affected by changes in templates overtime. Add a warning to module organisers when deleting a template.
Complexity to high for module organisers, this could lead to the feaure not being used or misconfigured and incomplete forms.	Implement an intuitive interface along with initial guidance, videos and documentation for the feature. Provide users with pre-configured templates that can be used as examples.
Database load may increase during grading periods reesulting in a slower perfomance in response across the CSRS.	Use caching for frequently accesed templates. Ensure integrity before grading periods through maintenance and testing.

Supporting material

(Note: first reference is fictitoinal and should not be taken seriously, its more of an example of how to cite documentation.)

- Smith, J. R., and Patel, A. K. "Comprehensive Student Record System (CSRS): Design, Implementation, and Best Practices," CSRS Technical Documentation, 2024.
- Wisniewski, B., Zierer, K., & Hattie, J. (2019). The Power of Feedback Revisited: A Meta-Analysis of Educational Feedback Research. *Frontiers in Psychology*. Retrieved from Frontiers .