Minutes of the First Client Meeting

Group 14B

Tuesday 12nd March 2024

Client: Cruz Izu
Chair: Jingqi Wang
Secretary: Shiwen Cao
Participants: Zhangchi Pan, Lanxi Zhang, Zhige Zhang

1 Time and Place

The first client meeting for the MCI Group Project was held in IW 4.21 at 1pm on Tuesday 12 March 2024.

2 Agenda:

I. Discuss the main goals and functions of the project

II. Understand the process of code submission and review

III. Clarify the roles and involvement of students and instructors

IV. Decide on the technology stack and user interface design

3 Discussion:

1) **Project Goals and Functions:**

Client firstly identify that the main purpose of product is Instruction. She expected the products not like Gradescope, but be similar like the dictionary project(project12).

The main goal of this project is to create a platform where only instructors can submit and discuss the unqualified code examples written by students(from their exams). These examples will showcase common coding mistakes or inefficiencies that students often make, even though their code may pass the tests or assignments. Thus to the educators will get to know how to teach students better

“Code smells” code is not clean and good structure.

The platform will serve as a "code dictionary" and a Forum where students can refer to and learn from these examples. They can understand why certain coding practices are considered bad and how to improve their code quality.

2) The key functions include:

- Allowing instructors to submit student code snippets along with context (course, language, assignment details)

- Providing a multi-stage review process for instructors to analyze, discuss, and reach consensus on issues and improvements

- Maintaining a validated collection of annotated code examples categorized by issue types

- Offering a user-friendly interface for students to browse and learn from the "code dictionary"

3) Code Submission and Review Process:

- Instructors will submit student code examples along with metadata like course level, programming language, and a brief explanation of the issue(s) they identify.(When submitting code, there will be tick box to select language and error and so on.)

  - Start with broad category and the instructor can fill the category inside.
  - We do not give to many "Class". When we give to many choice, we may drive moderators' thinking and need to figure out if it is correct.

- Instructors /Moderators will review the submitted examples and write their opinions and approve them for discussion if deemed suitable.

- Approved examples will be open for review by other instructors, who can provide their opinions, suggestions for improvement, and classify the issues.

- The other Moderators must submit their opinions first and then able to vote for others comments.

- We are collecting different info from instruction. (For example: Should be for or while loop.)The instruction will decide it. The discussion phase aims to reach a consensus on the identified issues and their appropriate solutions.

- Once a consensus is reached, the example will be validated and added to the "code dictionary" for student reference.

4) Submitted Code file Size

- For code samples with many errors, the interface should provide room for diverse opinions

- The code will be displayed with line numbers, allowing reviewers to specify the range of lines containing issues (e.g., line 1, or lines 1-3).

- The platform focuses on small programs or code snippets, typically around 20 lines of

code, targeting basic programming concepts and data structures.(no more than 50 lines)

- If a program is larger, instructors should submit only the relevant method/Structure or part of the solution, rather than the entire codebase

- Instructor can Upload file and copy paste.

5) Roles and Involvement:

- Students will not directly submit code examples or participate in the discussions. The focus is on instructors contributing and evaluating the examples.

- Instructors will be the primary users, submitting code examples and engaging in discussions to reach a consensus on coding best practices.

- Instructors/Moderator will decide the way to rate. Maybe buttons: loops (2 examples)manage the approval process for submitted examples and facilitate the discussions.

6) User Authentication(university login API):

- We do not use university API as this product may be used in different universities.

- We do not need to focus on Authentication at first

- The Interface need security for the Moderator-only instruction can submit.

- logins may be added later if the platform scales across multiple institutions. It will be like shopping website, moderator can choose to log in on the last minute- log in or not. Or we can always put the login on top

7) Technology Stack and UI Design:

- The project may leverage GitHub Pages for hosting and generating the website, as it can provide a straightforward publishing solution.

- The code display should include line numbers to facilitate specifying the location of issues.

- The interface should adapt based on the size and complexity of the submitted code, showing fewer classification options for samples with many errors

- Initial designs may include a submission form with metadata fields, a discussion board for instructors, and a filterable "code dictionary" for students.

- The UI should be iterative, with refinements based on feedback and evolving requirements.

4 Next Steps:

- Decide on a regular meeting schedule (monthly meeting on Mondays 3:10PM) for progress updates and discussions.

- Client will provide us some code examples next week

- First milestone: Collect data and how we are going to save it.

- Begin prototyping the user interface and setting up the necessary infrastructure (e.g., GitHub repository, database). Next meeting we will make the file and let her check.

- Explore potential third-party tools or libraries that can simplify development and enhance functionality. Use as many thing we want. Its good to use tool and template. Any tool we use , we understand how to use and extend it.