Minutes: Sprint 2 Client Review - Conversational AI System for Error Checking Submission Forms

---

Date: 18 September 2024

Time: 11:30 – 12:00

Attendees: Mark Castalanelli (Client), Josh Shipton, Camilo Lima Castillo, Sibi Moothedan, Samuel Kent

1. Demo Presentation and Functionality Review: The team demonstrated the current state of the project, focusing on the error-checking functionality for various XLS files.

Key features presented included:

- Ability to process and analyse different CSV file formats (converted from XLS)
- Detection of formatting issues, missing data, and incorrect entries
- Generation of summaries highlighting potential errors and data inconsistencies
- Display of both original and processed CSV data

Specific files demonstrated:

- A file with white space issues
- A file without sample ID
- A file with wrong headers
- A file with a wrong line

2. Client Feedback and Observations:

a) Positive Aspects:

- Mark expressed satisfaction with the summaries provided and the ability to identify problematic lines in the files.
- The client appreciated the system's capability to detect various types of errors, including white space issues, missing data, and formatting inconsistencies.

b) Areas for Improvement:

- The client identified a need for the system to recognize different types of Chain of Custody (COC) forms based on their headers (currently 20 different types).
- Mark suggested implementing a more flexible approach to error checking that can adapt to various COC types without relying solely on hard-coded rules.

3. Discussion on Future Developments:

a) COC Type Recognition:

- The team and client discussed the importance of implementing a feature to identify the specific type of COC based on the file's structure and content.
- Mark proposed using a database or dictionary to store expected formats for different COC types, which the system could reference for more accurate error checking.

b) Database Connection:

- The team mentioned difficulties in connecting to the provided database. Mark offered to assist with troubleshooting this issue in a follow-up call.

c) AI Integration:

- The client inquired about the extent of AI usage in the current system. The team clarified that while error detection is primarily done using Python code, the AI (GPT) is used to generate conversational summaries of the errors found.

d) Error Checking Approach:

- Current implementation uses hard-coded rules for error checking.
- Mark suggested a more flexible approach that can adapt to different COC types.
- Proposed solution: Use a database or dictionary to store expected formats for different COC types.

4. Action Items:

- The team will work on implementing COC type recognition and flexible error checking based on the identified type.
- A team member will schedule a call with Mark to discuss the structure and data types for different COCs.
- The team will attempt to resolve the database connection issues, with Mark offering additional support if needed.

5. Client Satisfaction:

Overall, Mark appeared satisfied with the progress made and the direction of the project. He found the error detection and summary generation particularly useful and was enthusiastic about the proposed improvements for the next sprint.

6. Additional Notes:

- The team decided against implementing a feature to download corrected CSV files due to potential inaccuracies in AI-generated corrections.
- Mark mentioned he needs to fill out a questionnaire about the sprint review for the unit coordinator.