



THE UNIVERSITY OF
**WESTERN
AUSTRALIA**

CITS5206 Information Technology Capstone Project

Deliverable 1 Project Specification and Plans (DialogCoder)

Group Member	Student Number
Aadil Vagh	22791631
Xudong Chen	23778972
Icy Chen	23965953
Yifan Gao	23966753
Amit Dinesh Bhudiya	24032869
Prisca Wang	24041794

Table of Contents

1. Problem Statement	2
1.1 Current Problem	2
1.2 Current Solution and Limitations	2
2. Client Communication and MVP Agreement	4
2.1 Client Communication Details	4
2.2 Minimum Viable Product (MVP)	5
3. Project Planning and Management	6
3.1 Project & Workflow Management Methodology	6
3.2 Timeline and Milestones	6
3.3 Roles and Responsibilities	7
3.4 Collaboration Tools and Practices	8
4. Risk and Technology Assessments	10
4.1 Group Skills Audit Survey	10
4.2 Technical Risks	10
4.3 Operational Risks	11
4.4 Technology Assessment	11
Reference	13
Appendix	13
Appendix A: Project Resources	13
Appendix B: Client Communication Records and Meeting Minutes	13

1. Problem Statement

1.1 Current Problem

The project focuses on automating the coding of transcripts recording the dialogue of participants in submarine simulator team exercises. Currently, researchers manually analyze 64 one-hour voice recordings that have been transcribed into CSV files. Each file contains between 400-500 rows of dialogue, requiring two independent raters to manually code each utterance according to a qualitative coding frame.

1.2 Current Solution and Limitations

The Existing Solution

- Manual review of each dialogue row by two independent human raters
- Assignment of qualitative codes based on an established coding frame
- Manual compilation of frequency data for analysis

Limitations

- Extremely time-intensive
- Prone to human error and inconsistency
- Limited scalability for future experiments
- Delayed insights due to processing time
- Monotonous work leading to potential coder fatigue and reduced accuracy

Our Solution

We will develop an automated system to qualitatively code transcripts from submarine simulator team exercises. The system will:

- Classify timestamped dialogue rows into discrete communication types based on an existing qualitative coding frame
- Flag edge cases for human review to ensure accuracy
- Support bulk data upload for efficient processing of the 64 one-hour voice recordings
- Append coded classifications to CSV files by adding an additional column with the assigned code for each dialogue row
- Generate summary reports of communication type frequencies across all uploaded data

Benefits Delivered to Users

By automating this process, we will deliver the following benefits:

- Time Efficiency: Reduce coding time from hours to minutes per transcript
- Consistency: Ensure uniform application of coding criteria across all transcripts
- Scalability: Create a solution that can handle increasing volumes of data from future experiments
- Resource Optimisation: Free up researcher time for higher-value analytical work
- Rapid Insights: Enable quicker analysis of communication patterns across teams
- Reduced Fatigue: Eliminate monotonous manual coding work

Link to GitHub README.md File

https://github.com/amitbhudiya/CITS5206_DialogCoder/blob/main/README.md

2. Client Communication and MVP Agreement

2.1 Client Communication Details

Key Stakeholders

- **Client Representative: Dr. Owen Carter**
- Project Facilitator: Mr. Pavneet Bedi
- Development Team: Group 3

Communication Channels

- Microsoft Teams: For project ongoing discussions, document collaboration, meeting minutes, and facilitator connection.
- Microsoft Outlook (UWA Email): For client communications with formal updates.
- Github: For code repository, issue and milestone tracking, and project planning.

Link to Meeting Agenda and Minutes Document (Under MS Teams Group 3 Channel)

<https://uniwa.sharepoint.com/:w:/t/CITS5206SEM-12025-Group3/EfDEhnhJR7pFnPrzwaALRpoBL3Rn7CDffZVqt5pgSXLKTQ?e=noNDRH>

Link to Client Communication Records (Under MS Teams Group 3 Channel)

https://uniwa.sharepoint.com/:f:/t/CITS5206SEM-12025-Group3/EI8k5el7j0ZEpwXdSYbsZk0BVhQbou1-hITJYe1Tk_BGdg?e=Li08we

Initial Client Consultation

Following our initial meeting with the client, we established the following key requirements:

The product should be a qualitative coding application using purely deterministic rules that can be manually modified at any time.

- Could be run on a local, isolated computer (no Internet required)
- Where multiple transcriptions (CSV files) could be uploaded
- Where users could define a 'dictionary' of keywords and phrases and assign a number code to each, and for this dictionary to be saved and loaded again for future use.
- Where the application would process the transcriptions and add additional columns in the CSV files populated with numbers defined in the dictionary

2.2 Minimum Viable Product (MVP)

The agreed MVP will include following functional requirements. This was confirmed by the client on 26th March 2025 as per the client communications in Appendix B.

1. Automated Transcript Coding

- Process CSV transcripts by mapping each dialogue row to a communication category based on a user-defined keyword dictionary.
- Append a new column to each CSV file with the assigned communication code.

2. Bulk Data Processing

- Allow simultaneous processing of multiple CSV files to manage large data sets effectively.

3. Frequency Summary Report

- Generate summary reports providing frequency counts for each communication code across the processed datasets.

4. Offline Operation

- Ensure the application functions fully offline, maintaining data confidentiality and eliminating reliance on external services.

Detailed Storyboard for the GUI

https://uniwa.sharepoint.com/:b:/t/CITS5206SEM-12025-Group3/ERQFASwMOJFEnX_DTkvjVxIByiZNeAffYVrRFsq59X8nZw?e=aTcFP1

High Level UML User and System Interaction Diagram

[https://github.com/amitbhudiya/CITS5206_DialogCoder/blob/main/Project%20Files/Deliverable 1/Deliverable 1 UML.png](https://github.com/amitbhudiya/CITS5206_DialogCoder/blob/main/Project%20Files/Deliverable%201/Deliverable%201%20UML.png)

3. Project Planning and Management

3.1 Project & Workflow Management Methodology

Based on our Skills Audit Survey as outlined in section 4.1 we have chose Kanban as our project & workflow management methodology.

Kanban was chosen as the preferred methodology due to the team's varied availability and skillsets. Its visual, flexible task-tracking system allows team members to contribute at their own pace while maintaining clear visibility of project progress [1]. This approach supports continuous delivery and smooth collaboration, making it well-suited to the team's working style and the iterative nature of the project. We have elected to use [GitHub's built-in Kanban board](#).

3.2 Timeline and Milestones

To ensure on-time delivery and clear progress tracking, our team has divided the project into distinct phases and milestones. Each phase corresponds to specific development activities and deliverables.

Week 1-2: Requirements & Initial Setup

- Confirm high-level scope and requirements with the client
- Create GitHub repository and MS Teams channels

Week 3: MVP & Offline LLM

- Worked on MVP (Minimal Viable Product)
- Researched about LLMs
- Hugging face transformer offline testing

Week 4: Requirement & New MVP

- Worked on new MVP
- Change of Requirement due to security issues
- Dividing topics to work between teammates for D1
- Created Git issues for project plan
- Conduct a small test on deterministic algorithm

Week 5: Deliverable 1 Submission

- Refining Problem statement and MVP
- Client feedback on MVP and scope

- Finalize Project Specification and Plans (this document)

Week 6-8: Core Development & UI Implementation

- Build out the user interface for file uploads, dictionary editing, and flagged case display
- Enhance classification logic to handle edge cases and ambiguity
- Perform iterative testing with increasingly larger CSV data sets

Week 9: Software Feature Submission

- Provide a functioning version of the system (coding engine + UI)
- Submit partial documentation (user instructions, known issues)

Week 10-11: Refinement & Bulk Testing

- Gather user/client feedback; refine dictionary and classification rules
- Execute bulk processing tests across multiple transcripts
- Implement final UI touches (improved usability, error handling)

Week 12: Final Deliverable

- Complete and submit final report/documentation
- Deliver final, polished software system and demonstrate to client

3.3 Roles and Responsibilities

Our group is composed of six members, each focusing on complementary tasks to ensure efficient collaboration.

Role	Name	Responsibilities
Project Managers (PM)	Amit B & Aadil V	Coordinate overall timeline Manage weekly check-ins Ensure milestones are met
Backend Developer	Prisca W & Aadil V	Implements the classification engine, dictionary-based rules, and CSV processing logic Ensures robust code structure and integration with the UI
UI/UX Developer	Amit B	Designs and implements the user interface Focuses on usability, clarity for dictionary editing, and batch file handling

Data & Testing Specialist	Xudong C & Icy C	Create and execute test plans for accuracy, performance, and edge-case handling Ensure flagged/ambiguous dialogues are captured and tested thoroughly
Documentation & Reporting Lead	Yifan G	Prepares and updates project documentation (Deliverable Report, Meeting Minutes, Client Communication Records) Coordinates final deliverables, including reports and demonstration slides

3.4 Collaboration Tools and Practices

GitHub Repository

All code is stored in a private GitHub repository:
https://github.com/amitbhudiya/CITS5206_DialogCoder

We use GitHub Issues to track tasks, features, and bugs. Each issue is assigned to a relevant milestone (e.g., "Deliverable D1," "UI Implementation," "Final Polish").

Kanban Board

We have been using a Kanban board under the Github project to track issues that need actioning, those in progress and those completed.

<https://github.com/users/amitbhudiya/projects/1/views/2>

MS Teams

Weekly team meetings and real-time discussions occur in MS Teams. Meeting notes and action items are documented in a dedicated GitHub Issue, ensuring all members have access to ongoing progress updates.

Issue Tracking & Milestones

- o Each task is logged as a GitHub Issue with a clear title and acceptance criteria.
- o Milestones (e.g., "Deliverable 1: Project Spec and Plans," "Deliverable 2: Software Feature Submission") aggregate related issues.
- o Pull requests reference the issue they resolve and are peer-reviewed by at least one other team member before merging.

Communication Cadence

- o Weekly Team Meetings: Status updates, identification of blockers, assignment of new tasks.
- o Ad-Hoc Check-ins: Quick syncs via MS Teams for urgent issues or clarifications.
- o Client Updates: Summaries of progress shared bi-weekly, ensuring continuous feedback and alignment on MVP features.

4. Risk and Technology Assessments

4.1 Group Skills Audit Survey

We conducted a survey of all group members to assess the team's technical abilities, resource availability, and potential risks. It covers programming experience, familiarity with web development tools, CSV handling, and collaboration platforms like Git. It also assesses each member's weekly availability, access to development tools, and confidence in key areas, helping identify challenges early and plan effectively for project delivery.

Link to the survey questions, [click here](#).

Survey outcome summary:

The team has collectively demonstrated a strong foundation in Python, with nearly all members listing it as their primary language. Given its simplicity, versatility, and robust libraries for data handling, Python is the natural choice for this project. Most team members are also comfortable working with CSV files, making the use of the Pandas library ideal for efficient and reliable data processing. For the user interface, GUI tools like Tkinter were selected for their ease of integration with Python and suitability for desktop applications.

4.2 Technical Risks

Dictionary Management and Rule Complexity

Overly complex or poorly defined dictionaries can degrade coding accuracy.

Mitigation: Provide clear guidelines and validation checks within the application to ensure dictionary entries are correctly formatted and clearly defined.

Accuracy and Edge Case Handling

Deterministic rules may misclassify context-dependent or ambiguous utterances.

Mitigation: Implement automated flagging of ambiguous or borderline cases for manual human review, ensuring accuracy while maintaining efficiency.

Data Integrity

Processing multiple CSV files simultaneously might lead to errors in file handling, such as data overwriting or corruption.

Mitigation: Incorporate robust file-handling procedures with checksums and backup features to maintain data integrity during batch processing.

User Interface and Usability Issues

An overly complex or unintuitive interface could be cumbersome for the user

Mitigation: Develop a clear, intuitive interface based on standard UI design principles, supported by user feedback gathered through iterative prototyping.

4.3 Operational Risks

Project Timeline Delays

Unforeseen technical challenges or inadequate coordination may result in delayed project milestones.

Mitigation: Regular status updates, Kanban project management techniques, and contingency planning.

Team Coordination and Communication Issues

With multiple team members working concurrently, misunderstandings or overlapping tasks could occur.

Mitigation: Utilize structured communication platforms (GitHub Issues and MS Teams), and regular meetings to synchronize team activities and clarify responsibilities.

4.4 Technology Assessment

Development Platform and Tools Chosen

Building on the results of our Group Skills Audit Survey, Python was selected as the primary programming language for this project.

The choice was guided not only by the team's collective proficiency in Python but also by the client's preference for a language that is well-suited to data

handling and easy to adopt for future personal use. Python's simplicity, combined with its powerful libraries like Pandas for CSV processing and Tkinter for GUI development, makes it an ideal fit.

This alignment between the team's capabilities and the user's needs ensures both efficient development and long-term usability.

Programming Language: Python is chosen due to its versatility, ease of use, and extensive support for text-processing operations essential for this project.

User Interface: GUI libraries such as Tkinter will be used to create the user interface

CSV Processing: Pandas library will be utilized to reliably and efficiently handle large volumes of CSV data.

Technical Feasibility

The technology stack selected is mature, well-supported, and widely adopted, minimizing risks associated with software bugs and compatibility issues.

Offline execution guarantees data security, essential for sensitive or proprietary research data.

Reference

[1] "Kanban in Software Development – Kanban Tool," *Kanban Tool*, <https://kanbantool.com/kanban-software-development> (accessed Mar. 27, 2025).

Appendix

Appendix A: Project Resources

GitHub Repository

https://github.com/amitbhudiya/CITS5206_DialogCoder

Project Plan (GitHub Issues, Milestone 1)

https://github.com/amitbhudiya/CITS5206_DialogCoder/issues

https://github.com/amitbhudiya/CITS5206_DialogCoder/milestones

<https://github.com/users/amitbhudiya/projects/1>

Kanban Board (GitHub Projects)

<https://github.com/users/amitbhudiya/projects/1/views/2>

Group Skills Audit Survey Questions

https://github.com/amitbhudiya/CITS5206_DialogCoder/blob/main/Project%20Files/Deliverable_1/SkillsAssessmentSurveyQuestions.md

Appendix B: Client Communication Records and Meeting

Minutes

Client Communication Records

https://uniwa.sharepoint.com/:f/t/CITS5206SEM-12025-Group3/EI8k5eI7j0ZEpwXdSYbsZk0BVhQbou1-hITJYe1Tk_BGdg?e=Li08we

Meeting Agenda and Minutes

<https://uniwa.sharepoint.com/:w/t/CITS5206SEM-12025-Group3/EfDEhnhJR7pFnPrzwaALRp0BL3Rn7CDffZVqt5pgSXLKTQ?e=noNDRH>