Development Plan
Bridge

Buddies

Table 1: Revision History

Date	**Developer(s)**	**Change**
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

This report outlines VoiceBridge's project details, including an overview of the licensing, team norms, proof of concept plan, and the expected technology that'll be utilized throughout the development phase.

1 Confidential Information?

[State whether your project has confidential information from industry, or not. If there is confidential information, point to the agreement you have in place. —SS]

[For most teams this section will just state that there is no confidential information to protect. —SS]

2 IP to Protect

[State whether there is IP to protect. If there is, point to the agreement. All students who are working on a project that requires an IP agreement are also required to sign the "Intellectual Property Guide Acknowledgement." —SS]

3 Copyright License

[What copyright license is your team adopting. Point to the license in your repo. —SS]

4 Team Meeting Plan

[How often will you meet? where? —SS]

[If the meeting is a physical location (not virtual), out of an abundance of caution for safety reasons you shouldn't put the location online —SS]

[How often will you meet with your industry advisor? when? where? —SS]

[Will meetings be virtual? At least some meetings should likely be in-person. —SS]

[How will the meetings be structured? There should be a chair for all meetings. There should be an agenda for all meetings. —SS]

5 Team Communication Plan

[Issues on GitHub should be part of your communication plan. —SS]

6 Team Member Roles

[You should identify the types of roles you anticipate, like notetaker, leader, meeting chair, reviewer. Assigning specific people to those roles is not necessary at this stage. In a student team the role of the individuals will likely change throughout the year. —SS]

7 Workflow Plan

The team will use Git and GitHub for version control, to manage issues, and to handle pull requests. During the weekly meetings, the team will create a list of issues to complete. This will follow an issue template with a description that includes the Definition of Done (DoD), an assignee, a reviewer, as well as the source/purpose of the issue and any related dependencies or issues. Issues will follow a naming convention which will be included in the associated development branch. Commits will also include the issue number for traceability purposes.

To keep an organized project, PRs will be merged to main after a reviewer approves that the DoD has been met. Tests will be required where applicable in the DoD. The issue will be linked in the PR description.

Issues will be used to track all tasks, such as:

- Development
- Tests
- Team meetings
- Deliverable Submissions
- Infrastructure Setup

- Resolving Bugs
- Addressing Supervisor Feedback

The branch names will include a prefix to specify the type of issue worked on, including:

- feature/*
- bug/*
- infra/*
- docs/*

Once the assignee is confident with their completed issue, they will notify the reviewer and address potential feedback/concerns the reviewer may raise. After the assignee and reviewer agree on the issue completion, the issue owner will notify the team that it is complete.

Our team will use GitHub Actions to automate testing and maintain code quality standards. The CI pipeline will run unit tests on core functionality to ensure that bugs aren't introduced, and existing components are working as expected. A linting tool will be added to maintain consistent styling and to review code for potential errors.

PRs must pass all tests and lint checks before developers request review. Branch protection rules will prevent merging until a minimum of 1 PR approval is added.

8 Project Decomposition and Scheduling

The project will be organized on GitHub projects, with a linked Kanban-board to visualize issue completion. Swimlanes will be used to show the stage of the issue, including:

- **Backlog:** issues which have been identified and created
- **To Do:** lists issues which need to be done before the next meeting, have been groomed by 1 or more team members
- **In Progress:** issues that have been started
- **In Review:** development is complete, awaiting feedback (assigned to reviewer)

- **Done:** DoD is met, and team is informed of completion

Link to GitHub project: https://github.com/speech-buddies/VoiceBridge

9 Proof of Concept Demonstration Plan

What is the main risk, or risks, for the success of your project? What will you demonstrate during your proof of concept demonstration to convince yourself that you will be able to overcome this risk?

10 Expected Technology

We expect to use **Python** as the programming language.

We plan to use **PyTorch** or **TensorFlow** for model fine-tuning. We will extend and fine-tune existing models trained on standard speech, adapting them to accurately process dysarthric speech.

We will use the **PyLint** linter to ensure pull requests meet coding standards.

Testing will be done with **Pytest**.

For code coverage, we will use **coverage.py**.

Environment: For model training, proof of concepts, and performance testing, we will use Google Colab for convenience and ease of use. The final application will likely require a managed Python environment, such as **Conda**, to handle custom dependencies and allow easy local integration of project components.

Compute Resources: Since the project may involve handling large amounts of data during training and tuning, we plan to use McMaster CAS GPU clusters. Alternatively, we may consider Google Colab Pro, which offers pay-as-you-go compute units.

11 Coding Standard

We will follow the PEP 8 coding standard.

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. Why is it important to create a development plan prior to starting the project?

2. In your opinion, what are the advantages and disadvantages of using CI/CD?

   Luna:

   CI/CD simplifies the integration of changes by automating the repetitive tasks of building and testing code, reducing the manual effort needed. It provides insights on the project's health, making it easier to track and resolve conflicts early.

   However, using CI/CD is ineffective without maintaining an up-to-date testing suite (95

3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

   We did not have disagreements in this deliverable. We were able to delegate tasks, and complete them ahead of our meetings, which allowed us to review the work when we met. This helped us stay on track, and avoid any coflicts.

Appendix — Team Charter

[borrows from University of Portland Team Charter —SS]

External Goals

[What are your team's external goals for this project? These are not the goals related to the functionality or quality fo the project. These are the goals on what the team wishes to achieve with the project. Potential goals are to win a prize at the Capstone EXPO, or to have something to talk about in interviews, or to get an A+, etc. —SS]

Attendance

Expectations

[What are your team's expectations regarding meeting attendance (being on time, leaving early, missing meetings, etc.)? —SS]

Acceptable Excuse

[What constitutes an acceptable excuse for missing a meeting or a deadline? What types of excuses will not be considered acceptable? —SS]

In Case of Emergency

[What process will team members follow if they have an emergency and cannot attend a team meeting or complete their individual work promised for a team deliverable? —SS]

Accountability and Teamwork

Quality

[What are your team's expectations regarding the quality of team members' preparation for team meetings and the quality of the deliverables that members bring to the team? —SS]

Attitude

[What are your team's expectations regarding team members' ideas, interactions with the team, cooperation, attitudes, and anything else regarding team member contributions? Do you want to introduce a code of conduct? Do you want a conflict resolution plan? Can adopt existing codes of conduct. —SS]

Stay on Track

[What methods will be used to keep the team on track? How will your team ensure that members contribute as expected to the team and that the team performs as expected? How will your team reward members who do well and manage members whose performance is below expectations? What are the consequences for someone not contributing their fair share? —SS]

[You may wish to use the project management metrics collected for the TA and instructor for this. —SS]

[You can set target metrics for attendance, commits, etc. What are the consequences if someone doesn't hit their targets? Do they need to bring the coffee to the next team meeting? Does the team need to make an appointment with their TA, or the instructor? Are there incentives for reaching targets early? —SS]

Team Building

[How will you build team cohesion (fun time, group rituals, etc.)? —SS]

Decision Making

[How will you make decisions in your group? Consensus? Vote? How will you handle disagreements? —SS]