EchoText

DEVELOPMENT PLAN

TEAM MEMBERS: AHMAD GHADDAR, KEVIN XING, HARSH BHAGAT, ANDY HUANG

Project Overview

This project aims to develop a cutting-edge Text-to-Speech (TTS) application called EchoText, designed to enhance the accessibility and convenience of consuming written content. EchoText will empower users to listen to any text-based material on the go, offering a seamless alternative to traditional reading. The focus of EchoText is to use the latest advancements in transformer-based TTS models (AI voices) which can produce more natural-sounding, human-like voices for the user and advancement in modern smartphones to run everything locally. By integrating those advancements in our application, we can deliver high-quality audio suitable for many use cases like reading articles, books, emails, and other written content without the user worrying about user privacy or data security.

Project Purpose

This project's purpose is to develop a mobile application that allows the user to seamlessly convert text to speech using an advanced transformer-based TTS model, creating a more natural and human-like sounding voice. The app will enable the user to consume written content more efficiently, especially in situations where reading is not convenient or possible while ensuring that all processing happens locally on the user's device to maintain privacy and security.

Project Scope

The Scope of the project includes the development of a user-friendly mobile application (initially focusing on IOS devices), integration with a transformer-based TTS engine, and implementations of key features like voice customization, exporting of audio files, importing PDF files, and on-device processing. This project will also include comprehensive testing and optimization to ensure that the app performs well across many mobile devices.

Project Objectives

- All processing will be done locally to ensure data privacy and no reliance on external servers or services.
- Complete Minimum Viable Product by 9/24
    - Includes text input, TTS processing, and audio playback features
- Complete Final Product by 12/3
    - Includes all planned features

Team Organization (Roles and Responsibilities)

Team Lead / Meeting Minutes Taker – Kevin Xing
Presentation Lead / Frontend & UI/UX Lead – Ahmad Ghaddar

Documentation Lead / Testing & QA Lead – Andy Huang
GitHub Lead / Backend & Data Management Lead – Harsh Bhagat

Team Lead

The team lead is responsible for managing the project timeline by setting both soft and hard deadlines and ensuring smooth communication among team members. This role involves acting as the main point of contact with the professor and teaching assistant (TA). The team lead oversees the distribution of responsibilities and ensures that the team works efficiently to meet goals. In cases of unresolved disagreements or hardware distribution decisions, the team lead has the final say.

Meeting Minutes Taker

This meeting minutes lead is responsible for recording and distributing meeting minutes. This ensures that key discussions, decisions, and actions are documented and shared with the team to keep everyone informed and aligned throughout the project.

Presentation Lead

The presentation lead is responsible for creating and managing the presentation framework, ensuring consistency throughout the slides, and assigning sections to team members. This role ensures that all presentation elements are completed on time and meet the project's quality standards.

Frontend & UI/UX Lead

The frontend & UI/UX lead oversees the design and user experience of the project, makes final decisions regarding the architecture and design of the interface, and distributes frontend development tasks. The primary goal is to ensure the front-end is both functional and aesthetically pleasing.

Documentation Lead

The documentation lead sets up and shares the project documentation, including creating section headers and assigning tasks related to document writing. This role ensures all documents are properly structured, completed on time, and reviewed for grammar and formatting.

Testing & QA Lead

The testing and QA lead focuses on testing and quality assurance by creating and implementing a testing plan, reviewing code quality, and ensuring that all components of the project function as

intended. This role helps guarantee that the final product meets all technical and user requirements before submission.

GitHub Lead

The GitHub lead manages the Version Control Managment, Code Reviews and Pull Requests, Merge Management, Backup and Repository Management, Issue Tracking and Task Assignments, Documentation and Git workflow, Continuous Integration and Testing, Repository Access and Permissions. This role ensures that contributions from team members are merged correctly and efficiently.

Responsibilities of the GitHub Lead:

1. **Version Control Managment:**
    - GitHub Lead is responsible for overseeing the entire version control process using Git and GitHub
    - This involved setting up repository structure, coordinating the creation of branches, and enforcing best practices for version control such as feature branching and proper commit messages.
2. **Code Reviews and Pull Requests:**
    - They are responsible for assigning reviewers to pull requests and ensuring that any comments or issues raised during reviews are addressed before mergers.
    - At least two pull request approvals, including at least one from either the Team Lead or the Lead of the affected area, are required before merging.
3. **Merge Management:**
    - The GitHub Lead will handle merge conflicts that arise during development, assisting team members in resolving conflicts efficiently.
    - They will ensure that mergers into the main branch are made only after proper testing and verification of the code's stability.
4. **Documentation and Git Workflow:**
    - The GitHub Lead will ensure that the team follows a consistent Git workflow such as Git Flow or Feature Branch workflow to maintain code quality and streamline collaboration.
    - They will provide guidance training if any team member is unfamiliar with Git or GitHub processes, ensuring that everyone adheres to version control best practices.
5. **Repository Access and Permissions:**

- The GitHub Lead is responsible for managing access to the repository, ensuring that only authorized members have the correct permissions such as read, write or admin rights.
- They will coordinate with the team lead to update permissions as needed, ensuring the right balance between access and security.

By fulfilling these duties, the GitHub Lead ensures that the EchoText development proceeds smoothly, with proper version control, timely resolution of merge conflicts, and a stable, well-organized codebase.

Backend & Data Management Lead

The backend and data management lead manages the backend logic and database systems of the project, ensuring that data is stored securely and that the backend services are integrated seamlessly with the front-end. Proper management of backend operations and data is crucial for the stability and success of the project.

Problem Resolution Policies

A structured three-strike system is in place to handle conflicts or issues that arise during the project. This approach ensures that problems are addressed step-by-step, giving the individual multiple opportunities to resolve the matter before taking more serious actions.

1. **First Strike – Internal Team Discussion:**
   If a problem arises, such as missed deadlines, lack of communication, or disagreements, the issue will first be addressed internally within the team. The team lead will initiate a conversation with the individual involved to understand the situation and work toward a solution. This allows the team to resolve the issue collaboratively without outside intervention. The goal at this stage is to reach an agreement or adjust responsibilities as needed to ensure the project stays on track.

2. **Second Strike – Involvement of the Teaching Assistant (TA):**
   If the issue persists after the internal discussion or cannot be resolved within the team, the next step is to escalate the matter to the teaching assistant (TA). The team lead will communicate the problem to the TA, providing a summary of the steps already taken to resolve the issue. The TA will offer guidance, mediate the situation if necessary, and help facilitate a resolution that benefits both the individual and the team.

3. **Third Strike – Escalation to the Professor:**
   Should the issue persist despite the involvement of the TA, the final step is to escalate the

problem to the professor. At this point, the professor will be asked to review the situation and all previous resolution efforts. If the issue continues to hinder the team's progress, the professor will be requested to consider excluding the individual from being part of the team's final grade. This action would remove the individual's contributions from the group evaluation, making them responsible for their own outcome.

This process ensures that conflicts are handled fairly, with ample opportunities for resolution before taking more serious measures.

Project Plan (iterations, project Schedule)

This section will outline the project plan for the development of our iOS-based on-device transformer text-to-speech (TTS) application. The project plan is written to ensure smooth communication, efficient task management, and timely delivery of project milestones and documentations. To help with this goal, this section will discuss the following: meeting times and communication channels, development methodology, task schedule, and contingency plans.

Meetings and Communication

The project's meetings will mainly consist of team meetings. Each meeting will have a specific focus, such as strategy planning, problem-solving, or progress evaluations and updates. These meetings will help the team stay on track, identify any issues early on, and ensure they are resolved promptly. There will also be meetings with our assigned GTA for progress reports and general concerns.

Weekly Meeting Schedule:

- Tuesday 6:50 PM ~ 8:00 PM: Team meeting at Wayne State University campus.
- Wednesday 5:00 PM ~ 6:00 PM: GTA meeting via Zoom. In-person meetings can be requested.
- Wednesday 6:00 PM ~ 7:00 PM: Team meeting via Discord.
- Friday 9:30 PM ~ 10:30 PM: Team meeting via Discord.

Communication within the team will mainly happen in-person and in Discord, which is an instant messaging and social platform that will help facilitate the team in basic text communications, voice calls, keeping notes, text discussions, as well as sharing small files and media management.

Document sharing will be conducted via the web version of the Microsoft suite, as provided by Wayne State University. If there are large files that exceed Discord's file size limit that need to be shared, the files will be shared via Microsoft OneDrive.

Methodology

For our project's development, we will use a structured development strategy. This approach will clearly outline the development plan and milestones, making it easy to determine if the project is on track.

We have gathered and broken down the project features, which are subject to change, into 4 major milestones. For each milestone, the application will be built and tested, ensuring all the features for that milestone are functioning before moving on to the next milestone. During each milestone, depending on the focus of the feature, the corresponding team leader will break down and delegate the features or functionalities to teammates completely, with weekly updates and re-evaluations of the work and assignments.

First milestone:

- Deploy a local TTS model on a mobile device and have it inference a pre-defined sentence. This will serve as the foundation of our development, ensuring that the core functionality of TTS works on the device.

Second milestone:

- Implement basic functionalities and UI, including copy-paste text as input. Develop the ability to save, browse, play, and manage audio files.

Third milestone:

- Implement more advanced and useful features such as importing documents, basic playback control,model/voice selection, file encryption, and model adjustments.

Final milestone:

- Conduct testing of the application and improve UI and UX, and any other last minute changes if needed.

Task Schedule

The following table will outline our key activities, sorted by their due date. Each task will be assigned a task leader(s) based on its scope. The schedule includes buffers in the form of soft deadlines, to ensure we have enough time to deal with delays and give time to edit and refine the tasks.

Task	Assigned Leader	Start Date	Soft Deadline	Due Date
Project proposal	Presentation	9/2	9/4	9/5
TTS model deployment research	Model	9/2	-	TBD
Development plan document	Documentation	9/5	9/8	9/12
First prototype (milestone 1)	Model, Front end	9/11	9/20	9/24
First prototype presentation	Presentation	9/18	9/22	9/24
Requirements document	Documentation	9/11	9/25	10/1
Design specifications document	Documentation	9/20	10/11	10/17
Second prototype (milestone 2)	Front end, Back end, QA	9/25	10/20	10/24
Third prototype (milestone 3)	Front end, Back end, QA	10/25	11/8	11/12
Test plan document	Documentation	10/11	11/13	11/19
Final product (milestone 4)	Front end, Model Back end, QA	11/13	11/27	12/3
Peer evaluation	-	-	-	12/10

Risk Management

Risk management will involve regular assessments of potential technical and non-technical challenges and risks. Each milestone will include one or multiple risk review sessions or informal discussions to proactively address any issues that could affect the timely delivery of project prototypes. The task leaders and the team leader will be responsible for spotting potential risks and coordinating contingency actions if they are considered necessary.

Below are some identified risks and their contingency plans:

- iOS deployment challenges:
  If during the development of milestone one, deploying on iOS proves to be infeasible, we will discuss as a group, as well as with the GTA and professor to change to focus to either Android devices or using a client-server configuration.

- Feature scope adjustments:
  If during any of the milestones, the feature set and scopes are evaluated to be too large or small, features will be added or removed after discussing as a group and with the GTA.
- Team availability issues:
  If any team member cannot contribute long-term due to medical or other unavoidable circumstances, the team leader, or if unavailable, the current task leader will promptly contact the GTA to discuss immediate actions, which could include decreasing scope and reducing features, modifying the task schedule, and re-assigning team roles.

Configuration Management Plan

We will use Git and GitHub for version control as a primary tool and collaboration. This will allow us to track and manage changes to our code throughout the development cycle. By using Git' branching; features, new features or bug fixes will be developed in isolated branches to avoid disrupting the main codebase. Developers will submit pull requests once their work is ready to be reviewed, enabling team members to provide feedback and ensure code quality before merging changes into the main branch. This approach will enhance collaboration, reduce the risk for errors, and help maintain the stability of the project.

Technologies

Frontend

The mobile application will be developed using the latest versions of Swift 6 and SwiftUI 6 to build a highly responsive and modern mobile application. SwiftUI provides a robust framework for creating native iOS applications with declarative syntax, enabling seamless integration with Apple's ecosystem. The primary focus will be to ensure the app is fully responsive, adapting seamlessly to various screen sizes and orientations, offering an optimal user experience across devices.

The goal is to create a polished, responsive mobile app with the latest technologies while ensuring efficient performance and maintaining user interface consistency across iOS devices.

Backend

The backend of the application will be powered by the Sherpa-Onnx framework (1.10.24) to handle text data processing and transformer-based Text-to-Speech (TTS) models. The project will exclusively use the VITS model for high-quality, natural-sounding audio generation.

Additionally, SQLite (3.46.1) will be used to manage metadata, and NSUserDefaults will be leveraged for storing small pieces of data like user preferences/settings.

Testing

Testing will primarily be done by building and running the app on devices/emulators. Additionally, XCode's built-in testing tools will also be leveraged for UI and unit testing.