# Debate Feedback iOS App

A comprehensive iOS application for recording debate speeches with integrated timing, automatic transcription, and AI-powered feedback generation.

## Features

### ✅ Implemented (Phase 1-6)

* **Authentication System**
  + Teacher login with device-based authentication
  + Guest mode for quick access
  + Persistent sessions
* **Debate Setup**
  + 3-step wizard: Basic Info → Students → Team Assignment
  + Support for multiple formats: WSDC, BP, AP, Australs
  + Drag & drop team assignment
  + Customizable speech and reply times
  + Student level selection (Primary/Secondary)
* **Timer & Recording**
  + Precision CADisplayLink-based timer (60fps)
  + Simultaneous audio recording (M4A @ 128kbps)
  + Bell notifications at 1:00, time-1:00, time, and +15s overtime
  + Visual progress bar and REC indicator
  + Speaker navigation (Previous/Next)
  + Real-time recording status
* **Upload & Processing**
  + Background upload with progress tracking
  + Automatic retry with exponential backoff (3 attempts)
  + SwiftData persistence for offline capability
  + Upload status badges per speech
* **Feedback Viewing**
  + Grid layout of all speeches
  + Processing status indicators
  + Direct link to Google Docs feedback
  + Share functionality
  + Summary statistics

### Core Architecture

MVVM + Coordinator Pattern

├── Models: SwiftData (@Model)

├── Services: Business Logic Layer

│ ├── AudioRecordingService (AVAudioRecorder)

│ ├── TimerService (CADisplayLink)

│ ├── UploadService (URLSession)

│ └── AuthenticationService

├── Networking: APIClient with Mock Mode

└── Views: SwiftUI

## Getting Started

### Prerequisites

* Xcode 15.0+
* iOS 17.0+
* Swift 5.9+

### Installation

1. **Clone or open the project**

cd /Users/tikaram/Downloads/iOS/DebateFeedback

open DebateFeedback.xcodeproj

1. **Add Microphone Permission**
   * Open the project in Xcode
   * Select the DebateFeedback target
   * Go to the "Info" tab
   * Add a new entry:
     + Key: NSMicrophoneUsageDescription
     + Value: We need access to the microphone to record debate speeches.
2. **Configure Backend URL** (When Ready)
   * Open Utilities/Constants.swift
   * Update Constants.API.baseURL with your backend URL
   * Set Constants.API.useMockData = false
3. **Add Bell Sound Files** (Optional)
   * Create folder: Resources/Sounds
   * Add three audio files:
     + bell\_1.mp3 (single ding)
     + bell\_2.mp3 (double ding)
     + bell\_3.mp3 (triple ding)
   * Add to Xcode project and target membership
4. **Build and Run**
   * Select a simulator or device
   * Press ⌘R to build and run

## Project Structure

DebateFeedback/

├── App/

│ ├── DebateFeedbackApp.swift # App entry point

│ └── AppCoordinator.swift # Navigation coordinator

│

├── Core/

│ ├── Models/ # SwiftData models

│ │ ├── DebateSession.swift

│ │ ├── SpeechRecording.swift

│ │ ├── Student.swift

│ │ └── Teacher.swift

│ ├── Services/ # Business logic

│ │ ├── AudioRecordingService.swift

│ │ ├── TimerService.swift

│ │ ├── UploadService.swift

│ │ └── AuthenticationService.swift

│ ├── Networking/

│ │ ├── APIClient.swift

│ │ ├── Endpoints.swift

│ │ └── NetworkError.swift

│ └── Persistence/

│ ├── DataController.swift

│ └── FileManager+Audio.swift

│

├── Features/

│ ├── Authentication/

│ │ ├── AuthView.swift

│ │ └── AuthViewModel.swift

│ ├── DebateSetup/

│ │ ├── DebateSetupView.swift

│ │ └── SetupViewModel.swift

│ ├── DebateTimer/

│ │ ├── TimerMainView.swift

│ │ └── TimerViewModel.swift

│ └── Feedback/

│ └── FeedbackListView.swift

│

├── Resources/

│ ├── Assets.xcassets/

│ └── Sounds/ # Add bell audio files here

│

└── Utilities/

├── Constants.swift

├── AudioSessionHelper.swift

└── Extensions/

├── String+Sanitize.swift

└── Date+ISO8601.swift

## Usage Guide

### 1. Login

**Teacher Mode:**

* Enter your name
* Tap "Login as Teacher"
* Benefits: History access, auto-population (future), cloud sync

**Guest Mode:**

* Tap "Continue as Guest"
* Limitations: No history, feedback expires on next session

### 2. Setup Debate

**Step 1: Basic Information**

* Enter debate motion
* Select format (WSDC, BP, AP, Australs)
* Choose student level (Primary/Secondary)
* Adjust speech/reply times

**Step 2: Add Students**

* Type student names
* Press Enter or tap + to add
* Remove with X button

**Step 3: Assign Teams**

* Drag students from top row into team zones
* Position order auto-assigned by drop order
* All students must be assigned

Tap "Start Debate" when ready.

### 3. Recording Speeches

1. Review current speaker info (name, position)
2. Tap **START** to begin timer and recording simultaneously
3. Bell notifications:
   * 1 bell at 1:00
   * 1 bell at (time - 1:00)
   * 2 bells at time
   * 3 bells every 15s overtime
4. Tap **STOP** to end recording
5. Recording auto-uploads in background
6. Use **Previous/Next** to navigate speakers

### 4. View Feedback

* After debate complete, tap "View Feedback"
* See all speeches with status indicators
* Tap "View" to open Google Docs feedback
* Tap share icon to share link
* Processing happens asynchronously

## Configuration

### Audio Settings (Constants.swift)

enum Audio {

static let fileExtension = "m4a"

static let sampleRate: Double = 44100.0

static let bitRate = 128000 // 128kbps

static let numberOfChannels = 1 // Mono

}

### API Configuration

enum API {

static let baseURL = "https://your-vps-domain.com/api"

static let useMockData = true // Set to false for production

static let maxRetryAttempts = 3

static let feedbackPollingInterval: TimeInterval = 5.0

}

### Debate Formats

* **WSDC**: Modified World Schools (3v3, 5min speeches, 3min reply)
* **BP**: British Parliamentary (4x2, 7min speeches)
* **AP**: Asian Parliamentary (2v2, 6min speeches)
* **Australs**: Australs (3v3, 8min speeches, 3min reply)

## API Integration

The app expects the following endpoints when useMockData = false:

### Authentication

POST /api/auth/login

Body: { teacher\_id: string, device\_id: string }

Response: { token: string, teacher: {...} }

### Schedule Auto-Population (Future)

GET /api/schedule/current?teacher\_id={id}&timestamp={ISO8601}

### Create Debate

POST /api/debates/create

Body: { motion, format, teams, speech\_time, student\_level }

Response: { debate\_id: UUID }

### Upload Speech

POST /api/debates/{debate\_id}/speeches

Content-Type: multipart/form-data

Fields: audio\_file, speaker\_name, speaker\_position, duration\_seconds

Response: { speech\_id: UUID, status: "uploaded", processing\_started: true }

### Check Feedback Status

GET /api/speeches/{speech\_id}/status

Response: { status: "processing|complete|failed", google\_doc\_url: string?, error\_message: string? }

## Testing

### Mock Mode Testing

The app includes a comprehensive mock mode for testing without a backend:

1. In Constants.swift, ensure useMockData = true
2. Run the app
3. All API calls return simulated data
4. Upload simulates progress and success
5. Feedback polling returns mock Google Docs URL

### Testing Checklist

* Timer accuracy across different speech lengths
* Bell notifications at correct intervals
* Simultaneous recording and timing
* Upload progress tracking
* Network failure recovery (toggle airplane mode)
* Microphone permission flow
* Drag & drop team assignment
* Multiple debate formats
* Guest vs authenticated mode differences
* iPad and iPhone layouts

## Known Limitations / Future Enhancements

### Not Yet Implemented

* Auto-population from schedule API (Phase 7)
* History view for past debates
* Offline queue for uploads
* iPad-optimized layout
* Actual bell sound files (currently using system sound)
* Swipe gestures for speaker navigation
* Real-time audio level monitoring
* Video recording option
* Multi-language support

### Future Roadmap

1. **Phase 7: Polish**
   * Auto-population flow
   * iPad side-by-side layouts
   * Bell audio files
   * Accessibility improvements
   * Haptic feedback
2. **Phase 8: Advanced Features**
   * Pattern analysis across debates
   * Student-facing app
   * Export feedback as PDF
   * Real-time feedback preview

## Troubleshooting

### Microphone Permission Denied

1. Go to iOS Settings → Privacy → Microphone
2. Enable permission for DebateFeedback
3. Restart the app

### Recording Fails to Start

* Check microphone permissions
* Ensure no other apps are using the microphone
* Try restarting the app

### Upload Fails

* Check internet connection
* App will auto-retry 3 times with exponential backoff
* Tap retry button if all attempts fail

### Feedback Not Processing

* Processing can take 2-5 minutes
* App polls every 5 seconds
* Check backend logs if status remains "processing"

### Drag & Drop Not Working

* Ensure you're dragging the student chip (not just tapping)
* Drop into the colored team zone
* Try on a real device if simulator is problematic

## File Storage

* **Audio Files**: Stored in Documents/Recordings/
* **Database**: SwiftData container in app's data directory
* **Cleanup**: Files older than 7 days are auto-deleted

## Performance

* **Timer Accuracy**: ±100ms (CADisplayLink @ 60fps)
* **Recording Start Delay**: <200ms
* **Upload Initiation**: <1s after STOP
* **App Launch**: <2s cold start

## Dependencies

All native iOS frameworks - no external dependencies:

* SwiftUI (UI)
* SwiftData (Persistence)
* AVFoundation (Audio)
* Combine (Reactive)
* URLSession (Networking)

## License

[Your License Here]

## Support

For issues or questions:

* Create an issue in the project repository
* Contact: [Your Contact Info]

**Built with SwiftUI + SwiftData | iOS 17+ | Xcode 15+**