Real-Time Call Translation

Complete 10-Week Work Plan
18.11.2025 → 26.1.2026

Project Code:	25-2-D-5
Team Members:	Amir Mishayev, Daniel Fraimovich
Advisors:	Dr. Dan Lemberg, Mrs. Elena Kramer
Institution:	Braude College - Software Engineering
Duration:	10 Weeks (66 Days)
Final Presentation:	January 26, 2026

Project Summary: Development of a cross-platform mobile application enabling real-time multilingual voice communication with AI-powered voice cloning. Supports Hebrew, English, and Russian for 2-4 simultaneous participants.

1. Overview & Timeline

Project Goals:

- Develop a real-time voice translation system with <1s latency
- Support 2-4 simultaneous participants in multilingual calls
- Implement voice cloning to preserve speaker identity
- Create cross-platform mobile app (Android & iOS) using Flutter
- Design user-friendly interface for non-technical users
- Deploy using Docker for consistency and portability
- Integrate with Google Cloud APIs (STT, Translate, TTS)
- Produce comprehensive documentation and demonstration materials

10-Week Timeline Overview:

Week	Dates	Focus	Deliverables
1-2	18.11-29.11	Setup & Infrastructure	Docker, Database, Google APIs
3-4	2.12-15.12	Real-Time Core	WebSocket, Translation Pipeline
5	16.12-22.12	Flutter UI	Complete Mobile Interface
6	23.12-29.12	Voice Cloning	xTTS Integration
7	30.12-5.1	Testing	System Validation
8	6.1-12.1	Documentation	Project Book, Guides
9	13.1-19.1	Presentation	Poster, Video
10	20.1-26.1	Final Prep	Submission & Demo

Week 1: Project Setup & Infrastructure

18.11 - 29.11

Day 1: Monday 18.11 - Environment Setup

Responsible: Both Team Members | **Hours:** 8

✓ **GitHub Repository Setup**

Create repository with branch structure (main, develop, staging). Setup branch protection rules and Git workflow documentation.

✓ **Development Tools Installation**

Install Python 3.10+, Docker, Docker Compose, Flutter SDK. Verify installations with version checks.

✓ **Project Structure Creation**

Create backend/ and flutter-app/ directories with initial structure. Setup requirements.txt and .env.example files.

■ **Deliverable: Repository created, all tools installed, project structure ready**

Day 2: Tuesday 19.11 - Docker Setup & Database

Responsible: Amir (Backend Lead) | **Hours:** 8

✓ **Docker Compose Configuration**

Create docker-compose.yml with PostgreSQL, Redis, and backend services. Configure healthchecks and restart policies.

✓ **Database Schema Design**

Create SQLAlchemy models for User, Call, CallParticipant, Contact, VoiceModel. Setup database initialization scripts.

✓ **Test Docker Environment**

Build containers, verify all services start correctly. Test database connections and Redis cache.

■ **Deliverable: Docker environment running, database models created**

Day 3: Wednesday 20.11 - Database Models & Basic API

Responsible: Amir | **Hours:** 8

✓ **Complete All Database Models**

Finish User, Call, and Contact models with relationships. Add validation and helper methods.

✓ **FastAPI Application Setup**

Create main FastAPI app with CORS, middleware, and basic endpoints. Setup Swagger documentation.

✓ **Health Check Endpoints**

Implement /health and /api endpoints for monitoring.

■ **Deliverable: FastAPI app running with health checks**

Day 4: Thursday 21.11 - Flutter Project Setup

Responsible: Daniel (Frontend Lead) | **Hours:** 8

✓ **Create Flutter Project**

Initialize Flutter project with proper organization. Setup pubspec.yaml with all required dependencies.

✓ **Project Structure & Navigation**

Create lib/ folder structure: models/, services/, screens/, widgets/, config/. Setup navigation framework.

✓ **Basic Models**

Create Dart models for User, Call, Participant matching backend schema.

■ **Deliverable: Flutter project structure complete, app running**

Day 5: Friday 22.11 - Google Cloud Setup

Responsible: Both | **Hours:** 8

✓ **Google Cloud Project Creation**

Create GCP project, enable Speech-to-Text, Translation, and Text-to-Speech APIs. Create service account with appropriate roles.

✓ **Credentials Management**

Download JSON credentials, configure in backend. Update .gitignore to protect sensitive files.

✓ **Weekly Review**

Review all code written, test entire setup, commit to Git with proper PR.

■ **Deliverable: Week 1 complete, all infrastructure ready**

Day 6-7: Sunday-Monday 24-25.11 - Google Speech-to-Text Integration

Responsible: Daniel | **Hours:** 16

✓ **Google STT Service**

Create GoogleSTTService class with async transcribe method. Support Hebrew, English, Russian with proper language codes.

✓ **Google Translate Service**

Implement GoogleTranslateService with caching support. Handle language detection and translation for all language pairs.

✓ **Testing**

Write unit tests for both services. Test with audio samples.

■ **Deliverable: STT and Translation services working**

Day 8-9: Tuesday-Wednesday 26-27.11 - Google TTS & API Endpoints

Responsible: Amir + Daniel | **Hours:** 16

✓ **Google TTS Service**

Create GoogleTTSService with voice selection for each language. Support different speaking rates and voice parameters.

✓ **API Endpoints**

Create /api/google/speech-to-text, /api/google/translate, /api/google/text-to-speech endpoints with proper error handling.

✓ Integration Testing

Test complete STT → Translate → TTS pipeline end-to-end.

■ Deliverable: Week 2 complete, all Google APIs integrated

Week 3: WebSocket & Real-Time Core

1.12 - 14.12

Day 11-12: Sunday-Monday 1-2.12 - WebSocket Handler

Responsible: Amir | **Hours:** 16

✓ **ConnectionManager Class**

Build WebSocket connection manager to handle multiple calls. Implement connect, disconnect, and broadcast methods.

✓ **WebSocket Endpoint**

Create /ws/call/{call_id} endpoint with user authentication. Handle audio messages and control messages (mute, ping).

✓ **Message Routing**

Implement logic to route messages to correct participants.

■ **Deliverable: WebSocket infrastructure working**

Day 13-14: Tuesday-Wednesday 3-4.12 - Call Manager

Responsible: Amir | **Hours:** 16

✓ **CallSession Class**

Create CallSession to manage active calls with participants. Handle join/leave events and state synchronization.

✓ **CallManager Service**

Global call manager to handle multiple concurrent calls. Integrate with database for persistence.

✓ **Call API Endpoints**

Create /api/calls/create and /api/calls/{id}/end endpoints.

■ **Deliverable: Call management system complete**

Day 15-17: Thursday-Saturday 5-7.12 - Flutter WebSocket & Audio Services

Responsible: Daniel | **Hours:** 24

✓ **WebSocketService**

Create Flutter WebSocket client with connection handling. Implement send/receive with proper error handling.

✓ **AudioService**

Implement audio recording and playback using flutter_sound. Handle permissions and audio chunking.

✓ **ApiService**

HTTP client for REST API calls to backend.

■ **Deliverable: Flutter services ready**

Day 18-19: Sunday-Monday 8-9.12 - Translation Pipeline

Responsible: Amir | **Hours:** 16

✓ **TranslationPipeline Class**

Orchestrate STT → Translate → TTS for multiple participants. Implement parallel processing for efficiency.

✓ **Streaming Processing**

Handle audio chunks with 200ms buffers for low latency.

✓ **Performance Optimization**

Add caching, connection pooling, and parallel execution.

■ **Deliverable: Translation pipeline working end-to-end**

Day 20-21: Tuesday-Wednesday 10-11.12 - Audio Processing

Responsible: Daniel | **Hours:** 16

✓ **Audio Recording**

Implement chunked audio recording at 16kHz mono. Handle microphone permissions properly.

✓ **Audio Playback**

Create audio playback queue to prevent overlapping. Support multiple audio sources.

■ **Deliverable: Audio services complete**

Day 22-24: Thursday-Saturday 12-14.12 - Integration & Testing

Responsible: Both | **Hours:** 24

✓ **End-to-End Integration**

Connect all components: Flutter → WebSocket → Pipeline → Back to Flutter

✓ **Two-Party Testing**

Test basic 2-party call with Hebrew ↔ English translation

✓ **Bug Fixes**

Identify and fix critical issues. Performance tuning.

■ **Deliverable: Week 3-4 complete, basic translation working**

Week 5: Flutter UI & User Experience

15.12 - 21.12

Day 25-26: Sunday-Monday 15-16.12 - Login & Home Screens

Responsible: Daniel | **Hours:** 16

✓ **Login Screen**

Email/password authentication with validation. Register link and error handling.

✓ **Home Screen**

Bottom navigation with Contacts, Recents, Settings tabs.

■ **Deliverable: Login and Home screens complete**

Day 27-28: Tuesday-Wednesday 17-18.12 - Contacts & Settings

Responsible: Daniel | **Hours:** 16

✓ **Contacts Screen**

Display contacts list with call buttons. Add/remove contacts functionality.

✓ **Settings Screen**

Language preferences, voice sample management, notifications settings.

■ **Deliverable: Contacts and Settings complete**

Day 29-31: Thursday-Saturday 19-21.12 - Call Screen UI

Responsible: Daniel | **Hours:** 24

✓ **Call Screen Layout**

2x2 participant grid, live transcription view, call controls.

✓ **Participant Cards**

Visual indicators for speaking, language badges, names.

✓ **Call Controls**

Mute, speaker, end call buttons with proper state management.

■ **Deliverable: Week 5 complete - Full Flutter UI ready**

Week 6: Voice Cloning Integration

22.12 - 28.12

Day 32-33: Sunday-Monday 22-23.12 - Coqui xTTS Setup

Responsible: Amir | **Hours:** 16

✓ **Install Coqui TTS**

Add TTS library to requirements. Update Dockerfile with dependencies.

✓ **VoiceClonerService**

Create service to handle voice cloning. Support voice sample upload and synthesis.

✓ **Testing**

Test voice quality for English, Russian, Hebrew.

■ **Deliverable: Voice cloning service working**

Day 34-35: Tuesday-Wednesday 24-25.12 - Integration with Pipeline

Responsible: Both | **Hours:** 16

✓ **Enhanced Translation Pipeline**

Integrate voice cloning into pipeline. Smart fallback to Google TTS when needed.

✓ **Voice Sample API**

Create endpoints for uploading and managing voice samples.

■ **Deliverable: Voice cloning integrated**

Day 36-38: Thursday-Saturday 26-28.12 - Multi-Party & Testing

Responsible: Both | **Hours:** 24

✓ **3-4 Participant Testing**

Test with multiple users. Verify voice quality and latency.

✓ **Performance Metrics**

Measure latency with/without voice cloning. Optimize where needed.

✓ **Bug Fixes**

Fix issues found during testing.

■ **Deliverable: Week 6 complete - Voice cloning working**

Week 7: Testing & Optimization

29.12 - 4.1

Day 39-41: Sunday-Tuesday 29.12-31.12 - Comprehensive Testing

Responsible: Both | **Hours:** 24

✓ **Test Suite Creation**

Write unit tests, integration tests for all components.

✓ **System Testing**

Test all scenarios: 2-party, 3-party, 4-party calls.

✓ **Edge Cases**

Network interruptions, short speech, background noise.

■ **Deliverable: Comprehensive test suite**

Day 42-43: Wednesday-Thursday 1-2.1 - Performance Optimization

Responsible: Amir (Backend) + Daniel (Flutter) | **Hours:** 16

✓ **Backend Optimization**

Add Redis caching, connection pooling, parallel processing.

✓ **Flutter Optimization**

Audio chunking, state management optimization.

■ **Deliverable: Optimized system**

Day 44-45: Friday-Saturday 3-4.1 - Final Testing & Validation

Responsible: Both | **Hours:** 16

✓ **Real Device Testing**

Test on multiple phones, different OS versions.

✓ **Performance Validation**

Verify all metrics meet targets: latency < 500ms without cloning, < 1s with cloning.

■ **Deliverable: Week 7 complete - System stable and tested**

Week 8: Documentation - Project Book

5.1 - 11.1

Day 46-47: Sunday-Monday 5-6.1 - Project Book - Part A (Hebrew)

Responsible: Amir | **Hours:** 16

✓ **Cover Page & Introduction**

Team info, abstract, problem overview (2 pages in Hebrew).

✓ **Solution Description**

Architecture diagrams, activity diagrams, component description (4 pages in Hebrew).

✓ **Development Process**

Work phases, tools used, methodology (3 pages in Hebrew).

■ **Deliverable: Project book pages 1-10 (Hebrew)**

Day 48-49: Tuesday-Wednesday 7-8.1 - Project Book - Part B + User Guide

Responsible: Daniel | **Hours:** 16

✓ **Challenges & Solutions**

Technical and engineering challenges faced (4 pages in Hebrew).

✓ **Results & Conclusions**

Goals achievement, metrics, lessons learned (3 pages in Hebrew).

✓ **User Guide**

Installation, usage, troubleshooting (5 pages in English).

■ **Deliverable: Project book complete + User guide**

Day 50-51: Thursday-Friday 9-10.1 - Maintenance Guide

Responsible: Amir | **Hours:** 16

✓ **System Requirements**

Hardware, software requirements (English).

✓ **Installation Instructions**

Step-by-step setup for backend and Flutter (English).

✓ **Maintenance & Troubleshooting**

Monitoring, logs, common issues (English).

■ **Deliverable: Complete maintenance guide**

Day 52: Saturday 11.1 - Documentation Review

Responsible: Both | **Hours:** 8

✓ Final Review

Proofread all documents, check formatting, generate PDFs.

■ Deliverable: Week 8 complete - All documentation ready

Week 9: Poster & Video Production

12.1 - 18.1

Day 53-55: Sunday-Tuesday 12-14.1 - Poster Design

Responsible: Both | **Hours:** 24

✓ **Poster Layout**

Design A0 poster (120×80cm) with all sections: problem, solution, tech, results.

✓ **Visual Elements**

Architecture diagram, screenshots, graphs, QR code.

✓ **Print Preparation**

Export to PDF, test print, order from college print shop.

■ **Deliverable: Professional poster ready**

Day 56-57: Wednesday-Thursday 15-16.1 - Video Production

Responsible: Both | **Hours:** 16

✓ **Video Script**

Write 2-minute script covering intro, problem, solution, demo, conclusion.

✓ **Recording**

Record demo footage, voice narration, screen recordings.

✓ **Editing**

Edit video, add subtitles, music, transitions. Export to MP4.

■ **Deliverable: 2-minute video complete**

Day 58-59: Friday-Saturday 17-18.1 - Presentation Practice

Responsible: Both | **Hours:** 16

✓ **Presentation Script**

Write detailed script for 20-minute presentation with Q&A; preparation.

✓ **Practice Sessions**

Practice 10+ times, time each section, refine delivery.

✓ **Q&A; Preparation**

Prepare answers for expected questions.

■ **Deliverable: Ready for presentation**

Week 10: Final Preparations & Submission

19.1 - 26.1

Day 60-61: Sunday-Monday 19-20.1 - Final System Check

Responsible: Both | **Hours:** 16

✓ **Clean Build**

Clean rebuild of backend and Flutter. Test on fresh environment.

✓ **Multi-Device Testing**

Test on 3-4 devices with all scenarios.

✓ **Performance Validation**

Final latency measurements, verify all metrics.

■ **Deliverable: System verified working**

Day 62: Tuesday 21.1 - GitHub Final Prep

Responsible: Both | **Hours:** 8

✓ **Repository Organization**

Organize all files, update README, create docs/ folder.

✓ **Final Commit**

Commit all documents, ensure .gitignore is correct.

■ **Deliverable: GitHub repository complete**

Day 63: Wednesday 22.1 - Physical Preparation

Responsible: Both | **Hours:** 8

✓ **Print Poster**

Pick up poster from print shop, test mounting.

✓ **Equipment Check**

Charge all devices, test demo setup, prepare backup materials.

■ **Deliverable: All materials ready**

Day 64-65: Thursday-Friday 23-24.1 - Final Practice

Responsible: Both | **Hours:** 16

✓ **Final Practice**

Practice presentation 10+ times, test demo 20+ times.

✓ **Rest & Preparation**

Light review, ensure good rest before presentation day.

■ **Deliverable: Fully prepared**

Day 66: Sunday 26.1 - ■ PRESENTATION DAY

Responsible: Both | **Hours:** 8

✓ **Setup (08:00)**

Arrive early, set up poster, test equipment.

✓ **Submission (09:00)**

Submit GitHub link and all documents.

✓ **Presentation (10:00-16:00)**

Present project, demonstrate system, answer questions.

✓ **Celebrate! ■**

Project complete!

■ Deliverable: PROJECT COMPLETE! ■

Appendix A: Success Criteria & Performance Metrics

Technical Criteria:

Criterion	Target	Measurement Method
Translation Latency (without voice cloning)	< 500ms	End-to-end timing from speech to translated audio
Translation Latency (with voice cloning)	< 1000ms	Including xTTS processing time
Speech Recognition Accuracy	> 85%	Word Error Rate (WER) on test samples
Translation Quality	> 80% BLEU	BLEU score on standard test sets
Concurrent Participants	2-4 users	Successful multi-party calls
Language Support	HE, EN, RU	All language pairs working
Cross-Platform	Android + iOS	Apps running on both platforms
System Uptime	> 95%	Demo reliability during testing

Appendix B: Resources & Tools

Frontend:
- Flutter 3.16 - Cross-platform mobile framework
- Dart - Programming language
- Provider - State management
- WebSocket - Real-time communication
- flutter_sound, just_audio - Audio libraries

Backend:
- Python 3.10 - Primary language
- FastAPI - Web framework
- SQLAlchemy - ORM
- PostgreSQL 15 - Database
- Redis 7 - Caching and pub/sub
- Uvicorn - ASGI server

AI/ML Services:
- Google Speech-to-Text - Speech recognition
- Google Translate API - Machine translation
- Google Text-to-Speech - Speech synthesis
- Coqui xTTS v2 - Voice cloning

Infrastructure:
- Docker - Containerization
- Docker Compose - Orchestration
- Google Cloud Platform - External APIs

Project Contact Information:

Team Member	Email	Role
Amir Mishayev	Amir.mishayev@e.braude.ac.il	Backend Lead
Daniel Fraimovich	Daniel.fraimovich@e.braude.ac.il	Frontend Lead
Dr. Dan Lemberg	lembergdan@braude.ac.il	Academic Advisor
Mrs. Elena Kramer	elenak@braude.ac.il	Academic Advisor

Good luck with the project! ■