JAPANESE PITCH ACCENT TRAINER

FUNCTIONAL SPECIFICATION VERSION 1

HAFZA ABDULLAHI
4TH YEAR SOFTWARE DEV STUDENT
SUBMISSION DATE: 23/10/2025

CONTENTS

SYSTEM OVERVIEW

PURPOSE

A mobile application that helps Japanese learners improve pronunciation accuracy through pitch accent visualization, correction and spaced repetition scheduling

SCOPE

- Mobile application for individual language learners
- Real-time pitch analysis and visual feedback
- Basic spaced repetition for word review scheduling
- Scoring system compared to native pronunciation

USER REQUIREMENTS

USER STORIES

USER MARY: RECORD PRONOUNCIATION

As a Japanese learner, I want to record my pronunciation attempts so that I can receive feedback on my pitch and its accuracy.

USER IVANA: RECEIVE AN ACCURATE SCORE

As a user I want to see a percentage score for my pronunciations so that I can see if I have improved since the last attempt and track any new progress

USER MESUT: VIEW PITCH ANALYSIS

As a user I want to see a graph comparing my pitch contour with the native pattern so that I can visually identify my errors

USER EMILY: SELF-ASSESS PERFORMANCE

As a user, I want to rate my own performance so that the system can schedule appropriate review intervals.

USER JASMINE: USE SPACED REPETITION

As a learner, I want the system to show me words at optimal intervals so that I can retain correct pronunciation long-term.

FUNCTIONAL REQUIREMENTS

MANDATORY FEATURES SPECIFICATION

AUDIO RECORDING AND PLAYBACK

Description: Users can record and playback their pronunciation attempts

- **Input**: Microphone audio capture (2-5 second duration)
- **Processing**:
    1. Record button initiates recording
    2. Stop button ends recording
    3. Play button replays captured audio
- **Output**: Stored audio file for processing
- **Constraints**:
    1. Maximum recording length: 10 seconds
    2. Supported format: WAV (16kHz, mono)
    3. File size limit: 1MB per recording

BACKEND PYTHON PITCH ANALYSIS

Description: Audio is sent to backend for pitch extraction and analysis

- **Input**: Wav File file via POST
- **Processing**:
    1. Parse audio file using Parselmouth praat
    2. Extract Fundamental frequency (F0) contour
    3. Apply smoothing algorithms (moving average, gaussian and savitky-Golay)
    4. Calculate average smoothed pitch graph
- **Output**: Json response containing contour, example sample data
  {

    "time": [0.0, 0.1, 0.2, ...],

    "pitch contour": [120, 125, 130, ...],

    "score": 75,

    "status": "success"

  }
- **Error**:
    1. No sound recorded, return error message
    2. Noisy audio, poor quality: return error message

PITCH VISUALIZATION

Description: Display native speaker graph vs user pitch graph

- **Input**: Pitch data from backend analysis
- **Processing**:
    1. Render line graph with two traces:
        1. User pitch contour (blue line)
        2. Reference pitch contour (red line)
    2. X-axis: Time (seconds)
    3. Y-axis: Frequency (Hz)
- **Output**: graph display
- **Constraints**:
    1. Graph updates within 5 seconds of analysis completion
    2. Clear legend distinguishing user/reference lines
    3. Legend on graph

ACCURACY SCORING USING DYNAMIC TIME WARPING

Description: Provide visual and numerical feedback on pronunciation accuracy

- **Input**: Processed pitch contour data
- **Processing**:
    1. Calculate similarity score using Dynamic Time Warping
    2. Convert to percentage (0-100%)
    3. Round to nearest integer
- **Output**: Numerical score display
- **Scoring**:
    1. Score < 50%: "Needs Practice"
    2. Score 50-79%: "Good"
    3. Score ≥ 80%: "Excellent"

SELF-RATING INTERFACE

Description: Users assess their own performance for SRS scheduling

- **Input**: User selection from four options
- **Processing**:
    1. Display four buttons: [Try Again] [Hard] [Good] [Easy]
    2. Capture user selection
    3. Store rating with timestamp
- **Output**: Rating stored for SRS calculation
- **UI Requirements**:
    1. Clear visual distinction between rating levels
    2. Immediate feedback on selection

SPACED REPETITION SCHEDULING

Description: Schedule word reviews based on user self-ratings

- **Input**: User rating + previous performance history
- **Processing**:
    - Calculate next review date using Anki-style algorithm:
        1. "Again": 1 day interval
        2. "Hard": 3 day interval
        3. "Good": 1 week interval
        4. "Easy": 2 week interval
    - Update word's due date in local database
- **Output**: Updated review schedule
- **Review Rules**:
    - Default initial interval: 1 day for new words
    - Successful reviews increase intervals exponentially
    - Failed reviews reset to minimum interval

TECHNICAL SPECIFICATIONS

FLASHCARD ENTRY

```
{
  "word_id": "string",
  "japanese_text": "string",
  "pitch_pattern": "string",
  "reference_contour": [number],
  "next_review": "timestamp",
  "review_count": number,
  "JLPT_level": number
}
```

USER SESSION

```
{
  "session_id": "string",
  "timestamp": "datetime",
  "word_id": "string",
  "user_audio": "file_path",
  "score": number,
  "user_rating": "string",
  "pitch_data": [number]
}
```

API ENDPOINTS

POST

- **Purpose**: Process audio and return pitch analysis
- **Input**: Form data with audio file and word_id
- **Response**: Json with pitch data and score
- **Timeout**: 10 seconds maximum processing time

USER INTERFACE SPECIFICATIONS

SCREEN FLOW

1. **Main Screen**: Display current word + record button
2. **Recording Screen**: Record/stop controls with visual feedback
3. **Results Screen**: Graph , accuracy score , rating buttons
4. **Next Word**: Automatic progression after rating

INTERFACE REQUIREMENTS

- Touch-friendly buttons
- Clear visual hierarchy with pronunciation feedback as primary focus
- Immediate feedback for all user actions

SUCCESS CRITERIA

RECORDING FUNCTIONALITY

- User can start/stop recording
- Audio playback matches recorded input
- Recording quality sufficient for pitch analysis

ANALYSIS ACCURACY

- Pitch graph displays within 5 seconds of recording
- Score calculation consistent across multiple similar attempts
- Graph clearly shows user vs reference comparison

SPACED REPETITION

- Words rated "Again" appear within 24 hours
- Words rated "Easy" don't reappear for at least 1 week
- Review schedule stays the same between app sessions

USER EXPERIENCE

- All mandatory features accessible within 3 taps from main screen
- Error states handled properly with user-friendly messages
- Performance maintained with 100+ word database

CONSTRAINTS AND LIMITATIONS

TECHNICAL CONSTRAINTS

- Audio processing requires network connection
- Local storage limited to 50MB for audio files
- Maximum 1000 words in local database
- Android 8.0+ / iOS 12.0+ compatibility