TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE 492 - Senior Project Test Plan Report

Submission Date

27.12.2024

Group Members:

Bartu Özen

Elif Nazlı Böke

Gülce Ayşe Döker

Toygar Yurt

Table of Contents

1 Introduction

The goal of this Test Plan is to establish a structured approach for testing the real-time communication application, ensuring it meets both functional and non-functional requirements. This application leverages advanced technologies such as speech recognition, machine translation, and text-to-speech systems to provide seamless real-time voice and text translation during communication.

The Test Plan includes methodologies, responsibilities, entry/exit criteria, and quality objectives, facilitating clear communication among team members. This document will help ensure that the application aligns with user needs and maintains high quality in a competitive market.

2 Scope

This Test Plan applies to all stages of testing for the real-time communication application.

2.1 Functions to be Tested

Core Functionalities:

- Real-time voice translation during calls.
- Real-time text translation during text-based communication.
- User management, including adding and managing a friend list.
- Notifications and synchronization across platforms.

Technologies:

- Integration of speech recognition, machine translation, and text-to-speech systems.

User Interface (UI/UX):

- Testing for multilingual support, intuitive navigation, and user satisfaction.

Performance:

- Ensuring low-latency operation under diverse network conditions.

2.2 Functions Not to be Tested

- Customization or optimization of core AI models (speech recognition, translation, and text-to-speech).
- Hardware-specific features outside the scope of standard mobile devices and headphones.

3 Quality Objectives

3.1 Primary Objectives

The principal objective of the testing phase is to guarantee that the system adheres to all stated requirements, encompassing both functional and non-functional elements, while meeting the specified quality metrics and anticipated use case scenarios. The objective of this process is to maintain the quality standards of the product. By the conclusion of the development cycle, it is expected that users will perceive that the project has either met or exceeded their initial expectations as outlined in the requirements.

Any modifications, additions, or removals in the requirements will be carefully documented and tested to the highest possible quality level within the available time and resources of the testing team.

3.2 Secondary Objectives

The secondary objectives of testing are to identify and uncover all potential issues and associated risks, communicate these findings to the project team, and ensure that every issue is appropriately resolved before the application's release. Achieving this requires a thorough and systematic approach to testing, ensuring all system components are rigorously examined and any identified bugs are effectively addressed.

4 Test Approach

Exploratory Testing: Testers will use their expertise and the high-level design to explore and identify edge cases not explicitly covered in the requirements. It will focus on discovering edge cases and unexpected behaviors in subsystems like Friend Management, Notification, and Offline Functionality.

Integration Testing: Focused on subsystem interactions (e.g., User Management and Translation Subsystems).

Performance Testing: The system's ability to handle real-time translation requests and concurrent user interactions will be evaluated.

Automation Testing: The software's functionality will be validated and ensured that it meets the system requirements. (Ex: Unit test, smoke test)

Requirement-Based Testing: Test cases will be decided and designed based on detailed requirements provided in the high-level design, ensuring comprehensive validation of system functionality and alignment with user needs.

4.1 Test Automation

Unit Test: It will be done for key services such as translation accuracy, latency, and scalability, alongside smoke tests for the UI.

Smoke Test: Basic functionality of the application’s UI and server endpoints during each build will be tested.

Performance Metrics: Translation speed performance metrics will be captured ensuring minimal latency and scalability under concurrent user loads.

5 Entry and Exit Criteria

5.1 Entry Criteria

- High-level design documentation and requirements are finalized and approved.
- All development components for the test phase have been deployed in the QA environment.
- Necessary test data has been created and validated.
- QA resources are trained and have access to the required tools and environments.

5.2 Exit Criteria

- Performance benchmarks for translation accuracy and response time are met.
- No critical or high-severity bugs remain unresolved.
- All functional and non-functional requirements outlined are tested with the critical functionalities meeting acceptance criteria.

6 QA Role in Test Process

Each developer on the team is responsible for testing their own code with a focus on core functionality: chat message delivery and language translation accuracy. Basic automated tests check critical features like message sending and receiving, while translation quality is verified through manual spot checks. Bugs are tracked in GitHub Issues, and before deployment, the team does a quick run-through of the main features.

7 Resource and Environment Needs

7.1 Testing Tools

- Test case creation: Microsoft Excel
- Test case tracking: Microsoft Excel
- Test case execution: Manual, Postman
- Test case management: Microsoft Excel
- Defect management: Microsoft Word
- Test reporting: PDF
- Check list creating: Microsoft Excel

7.2 Configuration Management

Configuration management will be essential to maintain consistency and control during the testing lifecycle:

- Documents Configuration Management: Google Docs is used to version control testing-related documentation.
- Code Configuration Management: Git is used to track code changes and synchronize testing scripts with application updates.

7.3 Test Environment

- Android Emulator (Android 5)
- Android Emulator (Android 15)
- Xiaomi Redmi Note 12 (Android 14)
- Samsung Galaxy Tab S10+ (Android 14)