



**TED UNIVERSITY**

Faculty of Engineering

Department of Computer Engineering

**CMPE 491 - Senior Project High-Level Design Report**

Submission Date

25.05.2024

**Group Members:**

Bartu Özen **22804498074**

Elif Nazlı Böke **11863076002**

Gülce Ayşe Döker **16483072168**

Toygar Yurt **4012670916**

# Table of Contents

<b>1. Introduction</b>	<b>2</b>
1.1 Purpose of the system	2
1.2 Design goals	2
1.3 Definitions, acronyms, and abbreviations	2
1.4 Overview	3
<b>2. Proposed Software Architecture</b>	<b>3</b>
2.1 Overview	3
2.2 Subsystem Decomposition	3
2.2.1 User Management Subsystem	3
2.2.2 Friend Management Subsystem	3
2.2.3 Translation Subsystem	3
2.2.4 Communication Subsystem	3
2.2.5 Notification Subsystem	4
2.2.6 Offline Functionality Subsystem	4
2.3 Hardware / Software Mapping	4
2.4 Persistent Data Management	5
2.5 Access Control and Security	5
2.6 Global Software Control	5
2.7 Boundary Conditions	5
2.7.1 Initialization	5
2.7.2 Termination	6
2.7.3 Failure	6
<b>3. Subsystem services</b>	<b>6</b>
3.1 User Management Subsystem Services	7
3.2 Friend Management Subsystem Services	8
3.3 Translation Subsystem Services	8
3.4 Communication Subsystem Services	9
3.5 Notification Subsystem Services	9
3.6 Offline Functionality Subsystem Services	10
<b>4. References</b>	<b>10</b>

# 1. Introduction

## 1.1 Purpose of the system

The main purpose of the real-time translator application is to enhance communication by removing language barriers, allowing users to make voice calls and text messaging with simultaneous translation.

## 1.2 Design goals

- **Performance:** Ensures high translation accuracy and minimal latency in voice and text translations.
- **Scalability:** Ensures the app can support a growing user base.
- **Accessibility:** Ensures to provide a user-friendly interface for effortless navigation and operation.
- **User Privacy:** Ensures to implement strong privacy protections to ensure that user data is handled responsibly. This includes anonymizing data where possible and giving users control over their data.

## 1.3 Definitions, acronyms, and abbreviations

- **LLM:** Large Language Model
- **API:** Application Programming Interface
- **UI:** User Interface
- **NLP:** Natural Language Processing
- **SQL:** Structured Query Language
- **Microservices:** A software development approach where applications are composed of small, independently deployable services that work together to form a larger system.
- **Authentication:** The process of verifying the identity of a user or system attempting to access a resource.
- **Encryption:** The process of encoding information in such a way that only authorized parties can access it.
- **Real-Time Translation:** Instantaneous conversion of text, speech, or data from one language to another.

- **Caching:** Storing frequently accessed data temporarily to improve performance by reducing the need to fetch it from the original source repeatedly.

## **1.4 Overview**

This report outlines the proposed architecture and subsystems of the real-time translator app, detailing both hardware and software components and their interactions.

## **2. Proposed Software Architecture**

### **2.1 Overview**

The following sections discuss the subsystem interactions, hardware software mapping to clarify how different parts of the system will interact, persistent data management explains how the data will be kept, how the security and access control will be managed is explained in access control and security part, global software control clarifies the general flow of the system and lastly in case of any exceptions how the system will react is explained in boundary conditions part.

### **2.2 Subsystem Decomposition**

#### **2.2.1 User Management Subsystem**

Handles user authentication, registration, profile management, and account deletion.

#### **2.2.2 Friend Management Subsystem**

Manages the addition, removal, and listing of friends, including friend requests and search functionalities.

#### **2.2.3 Translation Subsystem**

Provides real-time translation for text and voice communication, leveraging advanced NLP and LLMs.

#### **2.2.4 Communication Subsystem**

Manages text messaging and voice calls, integrating real-time translation features for seamless communication.

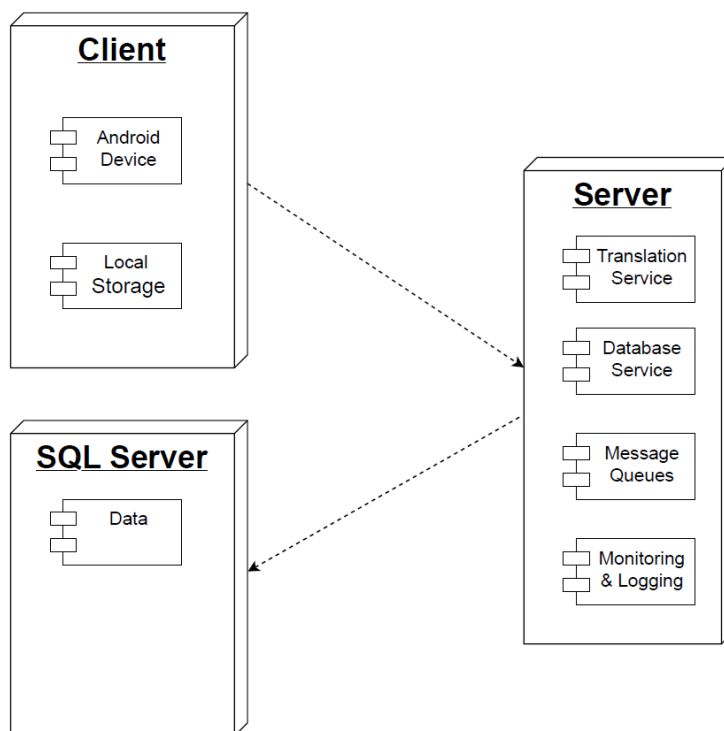
### 2.2.5 Notification Subsystem

Delivers notifications for friend requests, messages, and call invitations.

### 2.2.6 Offline Functionality Subsystem

Ensures access to cached messages and translations even when offline.

## 2.3 Hardware / Software Mapping



**Figure 1:** Hardware/Software mapping representation using UML deployment diagram

The Android devices act as client hardware, running UI applications for user interaction and communication with server components. Server hardware includes dedicated resources like translation servers, databases, and message queues, each running specialized software. The SQL Server, another piece of hardware, manages data storage and retrieval with its software, ensuring data integrity.

## 2.4 Persistent Data Management

Data persistence is managed through an SQL database. This database stores all data required to be stored in the long term such as user profiles, friend lists, and messages.

## 2.5 Access Control and Security

**Authentication:** Implemented with username and password authentication to obtain a token from the server.

**Data Encryption:** All sensitive data is encrypted at rest and in transit.

**Regular Security Audits:** Conducted to identify and mitigate vulnerabilities in the system.

## 2.6 Global Software Control

The real-time translator uses an event-driven software control. Users will make requests to the server by log in, send text and make calls. The server handles the requests and returns the processed result to the client. Also, it utilizes message queues for asynchronous processing of tasks like sending notifications and handling background translation jobs.

## 2.7 Boundary Conditions

### 2.7.1 Initialization

**System Boot-Up:** All servers (web, translation, database) are initialized and verified to be running properly. This includes starting up all necessary services and establishing connections to databases and message queues.

**Service Initialization:** Microservices responsible for authentication, user management, translation, and communication are started in a predefined sequence to ensure dependencies are met.

**Data Initialization:** Load essential data such as configuration settings, user sessions, and cached data from previous sessions into memory.

## 2.7.2 Termination

**Graceful Shutdown:** Allow current operations to complete before shutting down services. This involves completing ongoing translations, messaging, and calls to avoid abrupt disruptions.

**User Session Management:** Inform active users about the impending shutdown and save their session data to the database for future recovery.

**Service Termination:** Shut down services in an orderly manner, ensuring that dependent services are terminated after their dependencies have been safely closed.

**Data Preservation:** Save state data, logs, and important metrics to persistent storage to ensure no critical information is lost.

**Resource Cleanup:** Release allocated resources, close database connections, and clear temporary files or caches.

## 2.7.3 Failure

**Error Detection:** Implement monitoring and alerting systems to detect failures in real-time. This includes monitoring for server crashes, service unavailability, and performance degradation.

**Data Integrity:** Ensure data integrity through transactional operations and database replication. Use backup systems to restore any lost data.

**Post-Failure Analysis:** Conduct a thorough analysis of the failure to identify the root cause and prevent future occurrences. Update the system with patches or improvements based on the findings.

## 3. Subsystem services

Each subsystem in the real-time translator application provides a set of services that enable the application to function as intended.

### **3.1 User Management Subsystem Services**

The User Management Subsystem is responsible for handling all aspects of user accounts, including authentication, registration, and profile management.

#### **Authentication Service**

**Login:** Allows users to log in to the application using their email and password.

**Logout:** Manages user logout.

**Password Reset:** Provides functionality for users to reset their passwords via email confirmation.

#### **Registration Service**

**Sign Up:** Facilitates new user registration by collecting necessary information (name, email, password) and storing it securely.

**Email Verification:** Sends a verification email to new users to confirm their email address before activating the account.

#### **Profile Management Service**

**View Profile:** Allows users to view their profile information, including personal details and profile picture.

**Edit Profile:** Enables users to update their profile information and upload a new profile picture.

**Delete Account:** Provides functionality for users to delete their account permanently, including all associated data.



### **3.2 Friend Management Subsystem Services**

The Friend Management Subsystem handles the addition, removal, and management of user friendships.

#### **Friend Search Service**

**Search User:** Enables users to search for other users by username.

**Send Friend Request:** Allows users to send friend requests to other users found through the search.

#### **Friend List Service**

**View Friends:** Displays a list of all friends associated with the user's account.

**View Friend Requests:** Shows incoming friend requests with options to accept or reject them.

#### **Friend Management Service**

**Accept Friend Request:** Allows users to accept incoming friend requests.

**Reject Friend Request:** Enables users to reject incoming friend requests.

**Remove Friend:** Provides functionality to remove a friend from the friend list.

### **3.3 Translation Subsystem Services**

The Translation Subsystem provides real-time translation of text and spoken language.

#### **Speech Recognition Service**

**Voice Input Capture:** Captures audio input from the user.

**Language Detection:** Identifies the language spoken by the user and prepares it for translation.

## **Language Translation Service**

**Text Translation:** Translates text from the source language to the target language using advanced Large Language Models (LLMs).

**Contextual Understanding:** Uses LLMs to understand the context and nuances of the text to provide accurate translations.

### **3.4 Communication Subsystem Services**

The Communication Subsystem manages the core communication features of the application, including text messaging and voice calls.

#### **Messaging Service**

**Send Message:** Allows users to send text messages to their friends within the app. Messages are automatically translated into the recipient's preferred language.

**Receive Message:** Handles incoming messages and displays them in the user's preferred language.

**Message History:** Provides access to the history of conversations, including translated text.

#### **Voice Call Service**

**Initiate Call:** Enables users to start a voice call with a friend from their friend list.

**Receive Call:** Manages incoming call requests and allows users to accept or decline calls.

**Real-Time Translation:** Translates spoken language in real-time during the call, enabling seamless communication between users speaking different languages.

### **3.5 Notification Subsystem Services**

The Notification Subsystem ensures users are promptly informed about important events within the app.

## Notification Delivery Service

**Friend Request Notifications:** Sends notifications for incoming friend requests.

**Message Notifications:** Alerts users of new messages from their friends.

**Call Invitations:** Notifies users of incoming voice call invitations.

## 3.6 Offline Functionality Subsystem Services

The Offline Functionality Subsystem ensures that users have access to certain features even when not connected to the internet.

### Offline Access Service

**Cached Messages:** Allows users to view previously sent and received messages that are cached locally on the device.

**Cached Translations:** Provides access to cached translations for offline reference.

## 4. References

- [https://www.researchgate.net/figure/Hardware-software-mapping-Deployment\\_fig7\\_353382691](https://www.researchgate.net/figure/Hardware-software-mapping-Deployment_fig7_353382691)
- <https://www.conceptdraw.com/examples/hardware-software-mapping-deployment>
- *Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition*, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.
- Designing Whatsapp Messenger | System Design:  
<https://www.geeksforgeeks.org/designing-whatsapp-messenger-system-design>