TED UNIVERSITY

CMPE 491 / SENG 491 Senior Project

Local Generative AI Services Super App

Software Requirements Specification Report

22.11.2024

Team Members:

Deniz ÖZCAN, 33577146512, Software Engineering

Betül Ülkü YURT, 11056264926, Software Engineering

Umut ŞAHİN, 11597931646, Software Engineering

Supervisor:

Emin Kuğu

Jury Members:

Tansel Dökeroğlu

Kasım Murat Karakaya

Table of Contents:

1 Introduction

This document presents a detailed analysis of the project that our team will deliver, the project of which is a comprehensive web and mobile application platform that has a suite of AI powered mini apps and services, what sets it apart from the market is our platforms ability to implement users private API key from different AI models to provide users with a wide range of functionalities, promoting accessibility, creativity, and personalized experiences across diverse domains.

We plan to use various grounding methods to combine the power of several Large Language Models to provide more accurate and advanced results, to improve our results even further, our mini apps will use specialized and exclusively tailored prompts to provide users with better responses.

Our platform will be a one-stop-shop for unique instruments that help users on a daily basis with the mini apps. It addresses practical problems like privacy and security, adds value and builds diversity with solutions like video language translation, audiobooks of documents, customized night tales and media on demand. Incorporating the latest technologies in AI, since the prompts for our mini apps are developed and set by our team, the project ultimately aims to eliminate obstacles so that hard work is no longer necessary and creativity as well as content creation can be taken to new levels, even for people who are not familiar with AI services.

Here is a list of key mini services that we plan to implement to our platform as mini apps:

- **Video Language Translation** to enable seamless global communication by translating spoken language in real-time.
- **Multilingual Audio Documents** to make written content accessible to visually impaired users and those who prefer auditory experiences.
- **Bedtime Story Creator** encourages imagination and literacy by generating engaging, personalized stories for children.
- **Video Auto-Captions and Creation** to simplify video production and enhance accessibility through automated captioning and multimedia creation tools.
- **Daily Recap** to deliver personalized, podcast-style news summaries and topics tailored to user interests.
- **Text-to-Image Generator** to translate written descriptions into high-quality, visually striking images.
- **Image-to-Video Generator** to transform image sequences into captivating videos, enabling advanced storytelling and presentation capabilities.

Our comprehensive review of these features establishes a clear blueprint that bridges user needs with our technical capabilities. We've outlined specific requirements, operating parameters, and system architecture to guide development toward a solution that prioritizes both performance and usability.
This document serves as the cornerstone for all stakeholders, providing essential guidance throughout the development cycle while ensuring we maintain consistent direction and purpose across all project phases.

2 Current System

Currently, there are very few AI applications present which are scattered across various platforms, lacking a unified system that combines different features. Existing tools often don't work well together and lack standardization, making it difficult for users to navigate and use them. Also, users face challenges like fragmented workflows, reliance on external cloud services, and limited customization options. These issues hinder personalized AI experiences, reduce efficiency, and increase user frustration. Also, they are very expensive. Most AI tools rely on central servers, raising privacy and security concerns due to the transfer and storage of user data on external systems. This centralized approach limits user control and increases risks like data breaches or unauthorized access. Additionally, the absence of advanced techniques like multi-agent systems and real-time grounding reduces the ability to handle complex tasks and provide relevant, up-to-date responses.

The lack of local hosting options further increases these challenges, excluding users in areas with poor internet connectivity, or the users who choose not to give out their data. There are a few examples of good implementations of some of these services that include multi-agents, grounding etc. at the desktops. But according to our research, we couldn’t find any mobile application that let’s users just use their own generative AI key to create content.

A centralized, adaptable, and secure platform that integrates AI mini-apps, incorporates advanced techniques, and provides real-time capabilities is crucial to address these limitations. Such a system would improve user experience by enabling smooth interactions, promoting data privacy through decentralized models, and supporting advanced features tailored to individual needs. Also, users would have control and knowledge on whether to share their data, whom they are sharing fully.

3 Proposed System

3.1 Overview

The proposed system is a unified web and mobile application platform that combines the power of several advanced AI models to host a suite of mini-apps, each mini-app is designed to address a specific need, this helps the services be more specialized thus providing better, more accurate results. This system will use pre-made, openly accessible LLMs to deliver accessible, personalized, and innovative solutions, enabling users to overcome challenges in language translation, content creation, digital accessibility, and storytelling.

The prompts for each mini-app will be carefully crafted and pre-set by the development team. This will ensure accessibility for users who may not be familiar with AI technologies. This simplifies interaction and makes sophisticated AI capabilities available to a broader audience without requiring extensive technical knowledge.

For advanced users or those prioritizing privacy, the system will also offer the option to use their own API keys. This feature ensures greater security and control over data while providing personalized and private results. By supporting both built-in and custom API configurations, the platform caters to a wide range of user preferences and requirements.

The primary objectives of the system are:

- **Accessibility:** Empowering users with preset, intuitive tools while allowing customization for experienced users.
- **Personalization:** Delivering tailored outputs, such as custom bedtime stories or curated daily news recaps, based on user input and preferences.
- **Innovation:** Harnessing multiple AI models to unlock creative possibilities, such as generating images from text or transforming sequences of images into videos.
- **Efficiency and Security:** Simplifying complex processes and providing private, secure options for users who choose to manage their own API integrations.

The system will feature a scalable architecture that combines AI powered backend with a user-friendly, responsive and intuitive frontend. Security, data privacy, and user-centric design will be foundational aspects, ensuring the platform is both reliable and trustworthy.
By uniting various AI-driven mini-apps under one roof and offering customizable options, the proposed system aims to expand access to advanced AI capabilities while prioritizing user convenience, creativity, and security.

3.2 Functional Requirements

1. **Modular Accessibility**
   - **i.** Each mini-app must be accessible from a central dashboard with a clear interface that displays available applications.
   - **ii.** Users should be able to open, close, and switch between mini-apps seamlessly from the dashboard.
   - **iii.** The system should enable individual mini-apps to operate independently, ensuring users can use one app without needing others.

2. **Personalization Options**
   - **i.** Some mini-app should have customization settings based on user profiles, preferences, and usage history.
   - **ii.** For the Daily Recap feature, users should be able to select topics of interest and specify frequency (daily, weekly, etc.).
   - **iii.** For the Bedtime Story Creator, users can select themes, character names, story length, and language preferences.
   - **iv.** The platform should save personalization settings and apply them automatically for returning users.

3. **User Input Handling**
   - **i.** The system should allow users to input various data types (text, images, audio, and video) through an intuitive upload mechanism.
   - **ii.** For apps like Video Language Translation, users should be able to upload video files directly.
   - **iii.** Text-to-Image and Image-to-Video mini-apps should provide a description input box where users can detail desired image or video elements.
   - **iv.** Error handling must guide users if unsupported file types or data formats are uploaded.

4. **Data Collection and Storage**
    i. The platform must securely collect and store user data, following data protection and GDPR compliance standards.
    ii. User data retention policies must include clear options for data deletion and account removal upon user request.
    iii. Data should be encrypted at rest and in transit, ensuring the security of personal and generated content.

5. **Content Generation**
    i. Each mini-app should generate downloadable/copyable outputs in formats suitable for sharing (e.g., MP4 for video, PNG/JPG for images, MP3 for audio and text).
    ii. Ensure content generation adheres to content moderation policies, particularly for user-generated text in apps like the Bedtime Story Creator.

6. **API Key Management**
    i. Secure storage for API keys, including encryption and access controls, must be implemented.
    ii. Users should be able to add, remove, and update API keys for third-party services like OpenAI or Gemini.
    iii. The platform should verify and validate API keys before use, alerting users if their keys are invalid or require reconfiguration.
    iv. Permissions for accessing different APIs should be clearly outlined, with granular control to limit data usage according to user settings.

7. **Multi-Model Support**
    i. Users should have the option to select different AI models (e.g., Gemini, OpenAI, platform-hosted open-source models) based on available API keys.
    ii. Support for platform-hosted open-source Llama should be integrated as fallback options for users without third-party API keys.
    iii. Support for users who have their own service on their system can use the app with that service.

8. **Session Management**
    i. User sessions should be managed to allow for persistence across sessions, with options to remember or forget login credentials as chosen by the user.
    ii. Secure password management, account recovery, and session timeout policies should be implemented.

3.3 Non-Functional Requirements

1. **Performance Benchmarks**
   - **i.** Each mini-app should load within 10 seconds on a stable internet connection (5 Mbps or higher) for an optimal user experience.
   - **ii.** Media generation time for mini-apps like Text-to-Image or Video Auto-Captions should not exceed 1 minute for 10MB of input data. Larger files should maintain proportionate performance scaling.
   - **iii.** Regular performance testing and profiling will be conducted to identify and resolve bottlenecks, ensuring consistent performance under varying loads.

2. **Scalability**
   - **i.** Each mini-app must be designed to function independently, ensuring the addition of new mini-apps or updates to existing ones does not impact platform performance.
   - **ii.** Backend servers should handle peak usage efficiently, maintaining responsiveness even under high user load.

3. **User-Friendly Interface**
   - **i.** The platform should feature a simple, intuitive layout, allowing users to navigate easily between mini-apps and settings without extensive training or instructions.
   - **ii.** Clear instructions and tooltips must be provided for complex functionalities, ensuring users of all technical backgrounds can utilize the platform effectively.

4. **Security and Compliance**
   - **i.** The system must ensure the protection of user data through secure storage, encryption, and proper handling practices.
   - **ii.** Reliable authentication and authorization mechanisms should be implemented to control access securely.
   - **iii.** Communication channels must be safeguarded to prevent unauthorized interception of data.
   - **iv.** The platform is required to comply with industry standards and legal regulations for data protection and privacy.

5. **Cross-Platform Compatibility**
   - **i.** The platform should provide a consistent experience across web browsers and mobile devices.
   - **ii.** Responsive design principles will ensure the platform automatically adjusts to various screen sizes, from desktops to tablets and mobile devices.
   - **iii.** Browser compatibility testing will be conducted to confirm the platform's functionality on commonly used versions of each browser.

6. **Reliability**
   - **i.** The platform should maintain a 99.5% uptime over any rolling 30-day period, ensuring availability for users at all times.
   - **ii.** Error handling should provide clear feedback to users in case of an issue, with automatic logging of errors for troubleshooting and continuous improvement.

7. **Usability and Learnability**
    i. The onboarding experience should be streamlined, guiding new users through key features and functionalities to promote ease of use.
    ii. User feedback will be regularly collected to identify areas for improvement, with iterative updates to optimize the overall user experience.

8. **Sustainability**
    i. Data retention policies should encourage the removal of unnecessary data to minimize storage needs, conserving resources and reducing environmental impact.

9. **Maintainability and Extensibility**
    i. The platform should be developed with a modular architecture, enabling efficient maintenance, updates, and independent deployment of mini-apps.
    ii. A well-documented codebase with consistent coding standards and comments will support maintainability and ease of debugging.
    iii. API design and modular development practices should enable straightforward integration of future AI services or third-party tools without disrupting current functionality.

3.4 Pseudo Requirements

- Detailed specifications for each mini-app will be deployed separately, including which models are available to use and the estimated cost of it.
- The integration process for new mini-apps will utilize a scalable workflow, minimizing development time and ensuring rapid deployment.
- Step by step instructions will be provided for the users who would like to host their generative AI models by themselves and use these models in our application.
- The platform will provide a robust and reliable infrastructure for hosting and managing AI mini-apps.
- The platform will provide a secure and private environment for users and developers.
- The mini-apps in the platform will follow industry best practices and will be kept up-to-date.

3.5 System Models

3.5.1 Scenarios

Scenario 1: From Frustration to Precision with Multi-Agent Assistance:
Actors: Novice user, ChatGPT, Multi-Agent System, Platform
Description:
Alice, a student, needs a summary of her lecture notes to prepare for an exam. She opens ChatGPT and types, "Summarize my notes," along with uploading her notes file. However, the response she receives is overly generic, missing key concepts and details she wanted highlighted. Frustrated, Alice decides to try the platform's **Multi-Agent Assistance** mini-app.
She provides the same input, "Summarize my notes," and uploads the file. The system's **Requirement Analysis Agent** steps in, asking clarifying questions such as:

- "Should the summary focus on key topics or include examples?"
- "What length of summary do you prefer?"
- "Do you want any specific sections emphasized?"

After Alice provides her preferences, the Multi-Agent System refines the task and forwards the prompt to an LLM with specialized, pre-defined instructions. The output is a detailed, structured summary highlighting key concepts, examples, and critical sections from her notes.

Outcome:

Alice receives a precise and tailored summary that perfectly meets her needs. The experience demonstrates how the Multi-Agent System bridges the gap for users unfamiliar with effective prompt design, significantly improving the relevance and accuracy of the AI's response.

Scenario 2: Integrating Gemini API for Seamless Access:

Actors: User with Gemini API, Platform

Description:

Mark has purchased a subscription to Gemini API and wants to use it within the platform's mini-apps. He navigates to the **API Key Management** section, securely adds his Gemini API key, and links it to his account. Mark then accesses the **Video Language Translation** mini-app and uploads a video to be translated. The platform automatically uses the Gemini API to provide accurate and fast results. Mark uses the app on his mobile phone during a business trip, impressed by the seamless integration and convenience.

Outcome: Mark benefits from the advanced features of the Gemini API while enjoying the ease of use and functionality of the platform.

Scenario 3: Hosting Llama Model on a Private Server:

Actors: Experienced user, Private server, Platform

Description:

Sarah, an experienced developer, prefers to host an open-source Llama model on her private server to ensure full control over her data and costs. She sets up endpoints for her hosted model and provides these endpoints through the **Custom AI Configuration** section in the platform. Once configured, Sarah uses the **Daily Recap** mini-app to summarize her preferred news topics. The app communicates directly with Sarah's private server, utilizing her hosted model to generate content.

Outcome: Sarah successfully uses the platform's mini-apps with her custom model, combining advanced AI capabilities with full data sovereignty.

Scenario 4: Creating a Custom Mini-Service:

Actors: Advanced user, Platform

Description:

John, an advanced AI enthusiast, wants to create a custom mini-service for generating storyboards for films. He uses the platform's **Custom Mini-Service Builder**, a guided feature that allows him to set up a multi-agent LLM structure. John configures the Requirement Analysis Agent to break down user input into key storyboard elements (e.g., scenes, characters, dialogue) and links it to a Text-to-Image Generator for visual output. He then deploys the custom service within the platform, testing it with his own inputs. Once satisfied, he uses the service seamlessly from his mobile phone to create storyboards for his projects.

Outcome: John creates a fully functional, personalized mini-service, showcasing the platform's flexibility and support for user-driven innovation.

3.5.2 Use Case Model

Figure [1]: Use case model

3.5.3 Object and Class Model

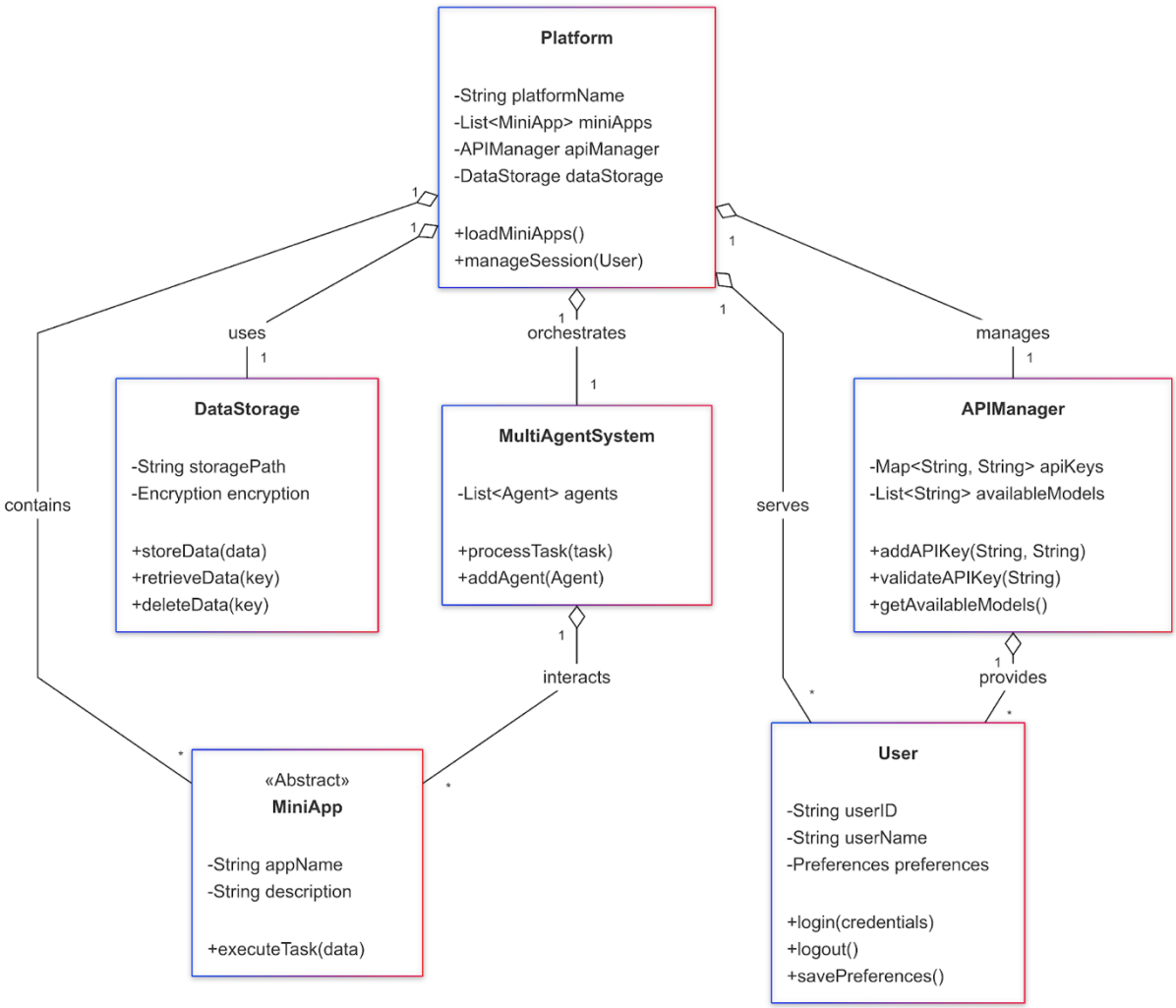

Figure [2]: Object Class Diagram

As you can see in the Figure 1, the system works obeying the principles of SOLID and OOP. It's planned to be a very scalable application that is easy to add any new MiniApp extending the MiniApp abstract class.

3.5.4 Dynamic Models

Activity Diagram:

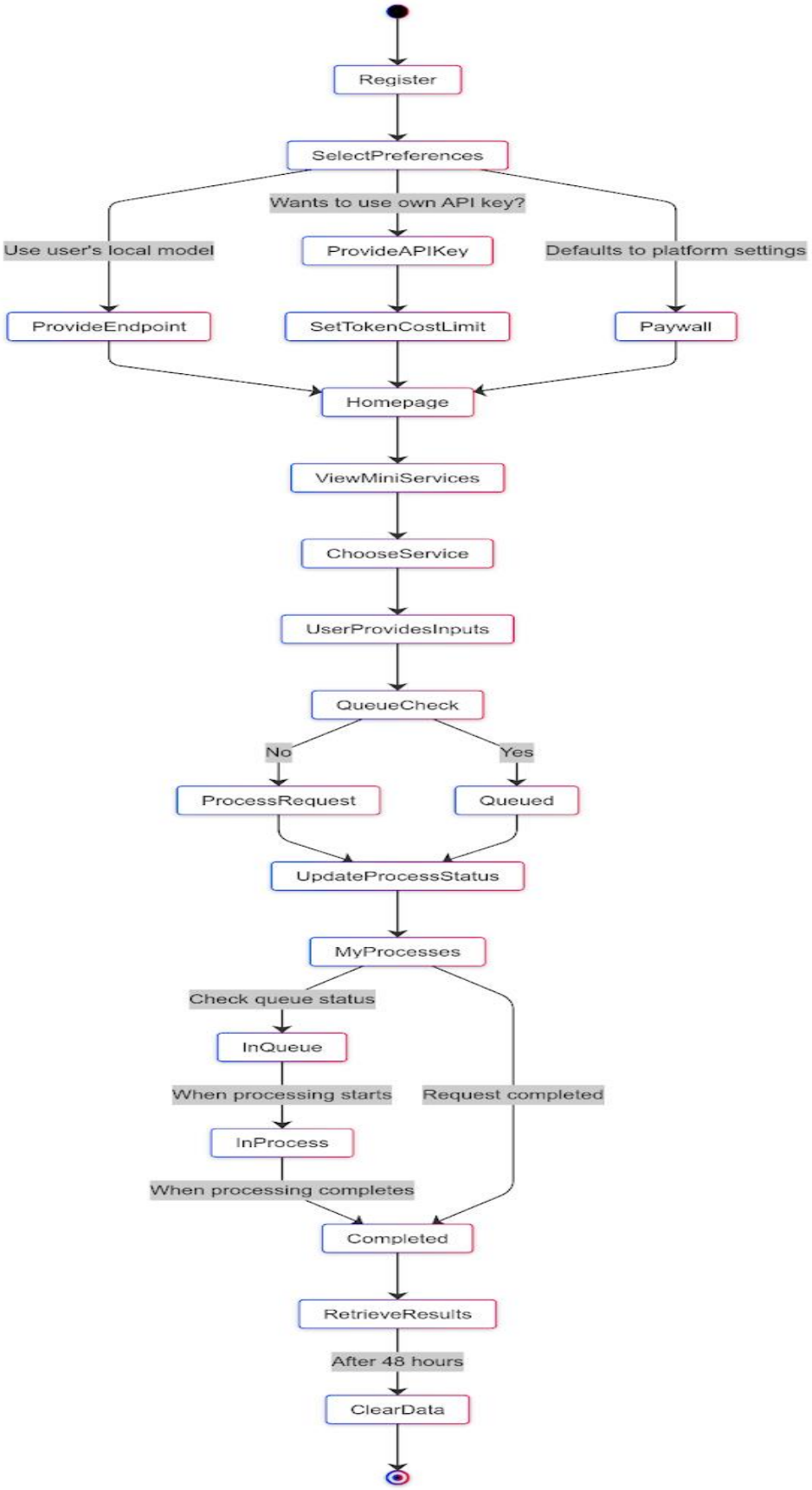

Figure [3]: Activity Diagram

Sequence Diagram for Login Process:

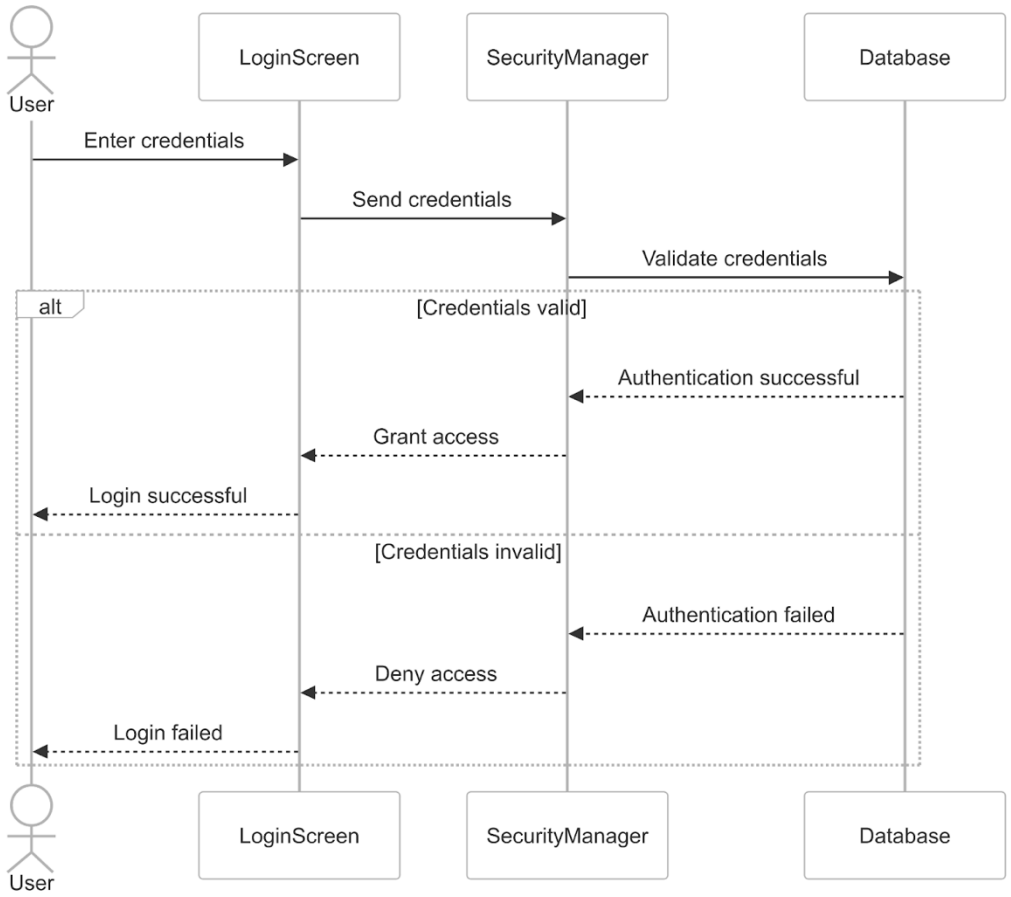

Figure [4]: Sequence Diagram

Sequence Diagram for Video Language Translation mini-app:

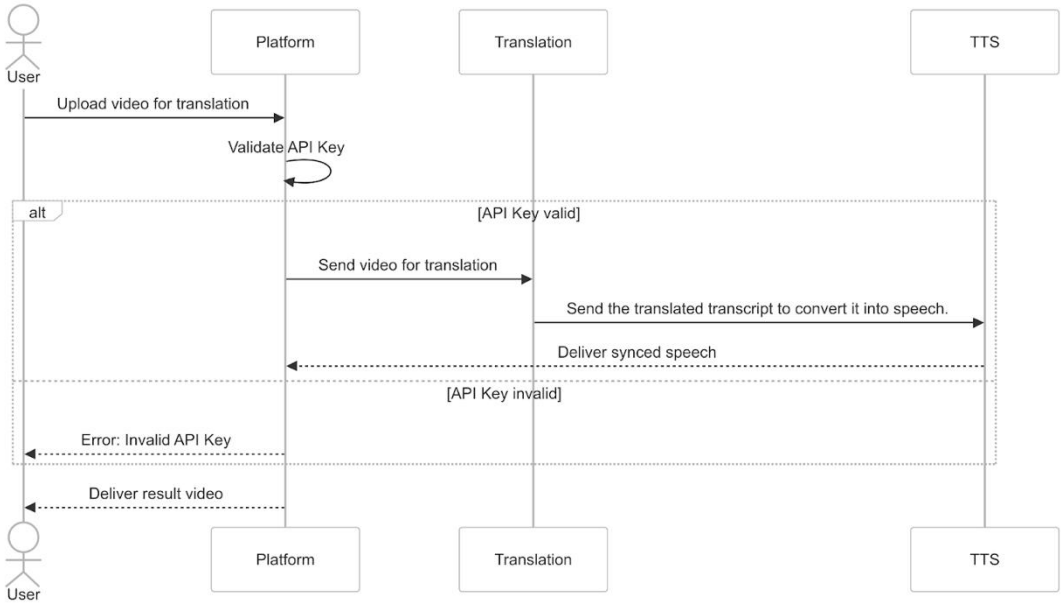

Figure [5]: Sequence Diagram

3.5.5 User Interface Mock-ups

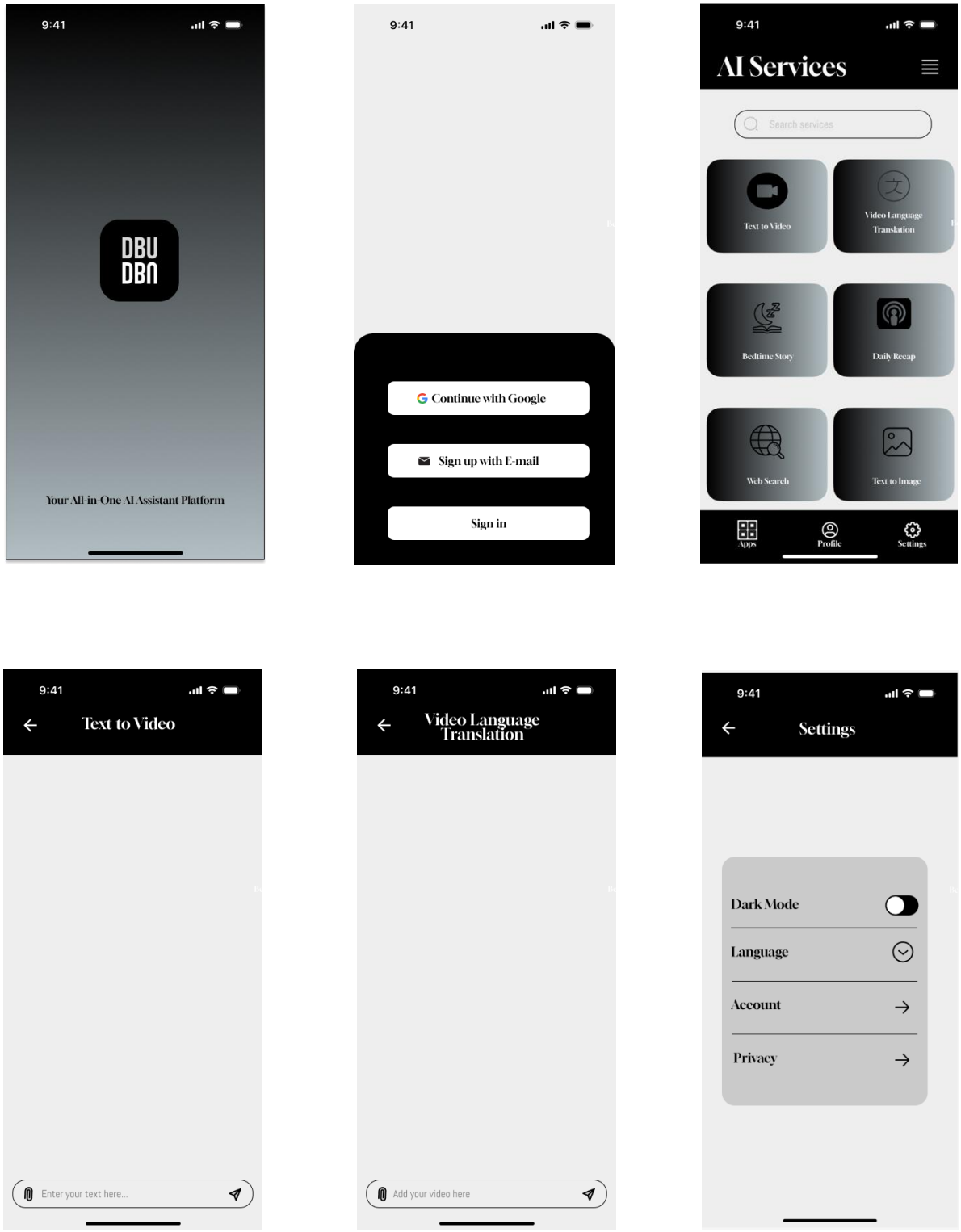

Figure [6]: User Interface mock-ups and navigational paths

4 Glossary

- **API (Application Programming Interface)**: A set of protocols, routines, and tools for building software applications that specifies how software components should interact.
- **GDPR (General Data Protection Regulation)**: A legal framework that sets guidelines for the collection and processing of personal information from individuals who live in the European Union.
- **LLM (Large Language Model)**: An advanced AI model trained on vast amounts of text data capable of understanding, generating, and processing human-like text.
- **Multi-Agent System**: An AI system comprising multiple intelligent agents that interact and collaborate to solve complex problems more effectively than a single agent.
- **Open-Source**: Software with source code that is freely available for modification and redistribution by anyone.
- **TTS**: Text to speech.
- **Prompt**: A specific instruction or input given to an AI model to guide its response or generate desired content.
- **SOLID Principles**: A set of object-oriented design principles aimed at making software designs more understandable, flexible, and maintainable:
    - Single Responsibility Principle
    - Open-Closed Principle
    - Liskov Substitution Principle
    - Interface Segregation Principle
    - Dependency Inversion Principle
- **Super App**: A mobile/web application that offers multiple services and functionalities within a single platform.
- **User Interface (UI)**: The point of human-computer interaction and communication in a device or software application.

5 References

Tundwal, D. (n.d.). *Mastering non-functional requirements for mobile application development*. Medium. Retrieved November 22, 2024, from https://dtundwal.medium.com/mastering-non-functional-requirements-for-mobile-application-development-d77e3235fc0d

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.