TED UNIVERSITY

CMPE 492

Senior Design Project II

MESAJLA

Final Report

26/05/2025

Group Members:

- **Ahmet Tunahan Küçükgökce**
- **Altar Gürsoy**
- **Burak Eren Birinci**
- **Serdar Kemal Topkaya**

Table of Content

Introduction 3

Brief Overview of the Project 3

Objectives and Motivation 3

Scope and Limitations 3

Scope: 3

Limitations: 3

Final Architecture and Design 5

System Overview 5

Architecture Diagrams 6

Design Decisions and Rationale 7

Final Project Status 8

Impact of Engineering Solutions 9

Contemporary Issues 10

Current Trends and Challenges 10

Legal, Ethical, and Privacy Concerns 10

Societal Implications 10

Tools and Technologies Used 11

Background and Resource Usage 12

Test Results 13

Conclusion and Future Work 21

References 22

Introduction

Brief Overview of the Project

Mesajla is a real-time, web-based chat application designed to bring together traditional messaging features with AI-enhanced communication. What sets Mesajla apart is its integration of AI-generated message suggestions during conversations, offering users a faster and more dynamic way to communicate. The app supports core features such as real-time messaging, group chats, profile management, and secure login, all powered through Firebase services.

Objectives and Motivation

The primary goal of Mesajla is to enhance the quality and efficiency of user interactions by offering smart AI-generated responses. With the increasing demand for AI-supported applications and secure digital communication, this project is driven by the need to create a platform that balances usability, performance, and user privacy. Mesajla aims to provide a chat experience that feels both intuitive and intelligent.

Scope and Limitations

Scope:

- Real-time messaging and persistent message history.
- AI-driven message suggestions and smart reply to functionality.
- User account and group chat management.
- Secure sign-up, login, and session management using Firebase.

Limitations:

- System performance may be affected by heavy concurrent usage due to resource constraints.
- AI suggestions are designed to protect user privacy, which may reduce the level of personalization.
- The app is currently optimized for desktop and responsive web use; there is no dedicated mobile application yet.

Final Architecture and Design

System Overview

Mesajla follows a modular architecture, separated into front-end and back-end components:

- **Front-End Modules:** They handle the user interface, including components like chat display, login/registration screens, user profile management, and AI-generated suggestions.
- **Back-End Modules**: They include Firebase services for authentication, data storage (Firestore), and media storage, along with a backend interface that communicates with the AI model and handles real-time message synchronization.

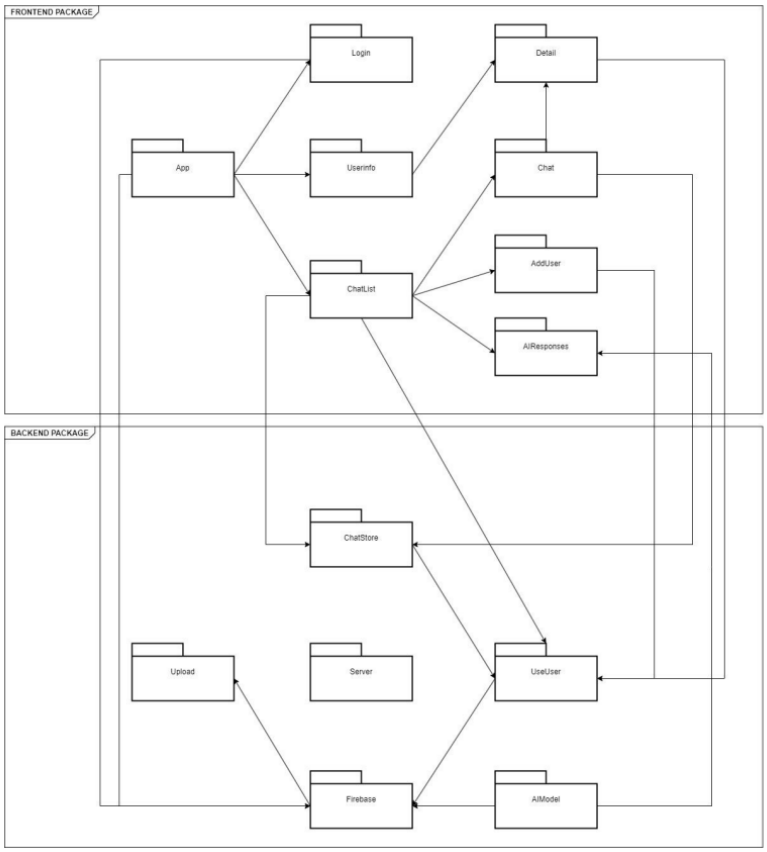

Figure 1: Package Diagram of the application

Architecture Diagrams

While the visuals are included in the Low-Level Design Report, here's an overview of the models used:

- **Package Diagram**: Shows how the front-end components (e.g., Chat.jsx, Login.jsx) interact with back-end services (Firebase.js, ChatStore.js, Upload.js).
- **Class Interfaces**: Each major class includes its variables and methods, defining its responsibilities within the system.
- **Use Case Models**: Cover user flows such as signing up, chatting, receiving AI suggestions, and managing group memberships.
- **Dynamic Models**: Represent the flow of data between user actions, the database, and AI-based responses.

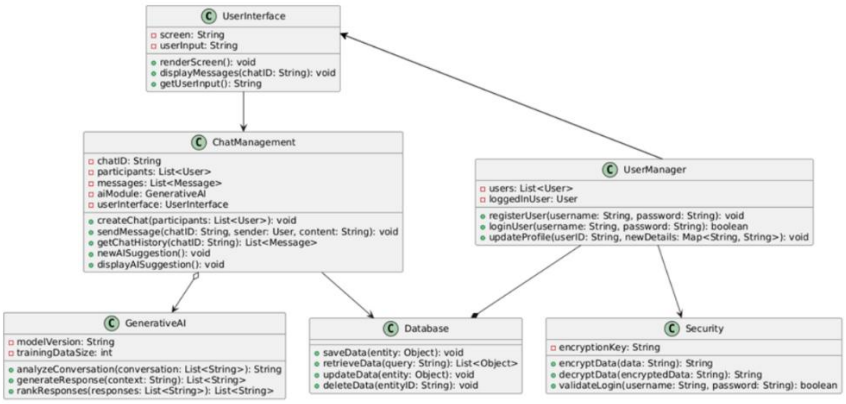

Figure 2: Class Diagram of the application

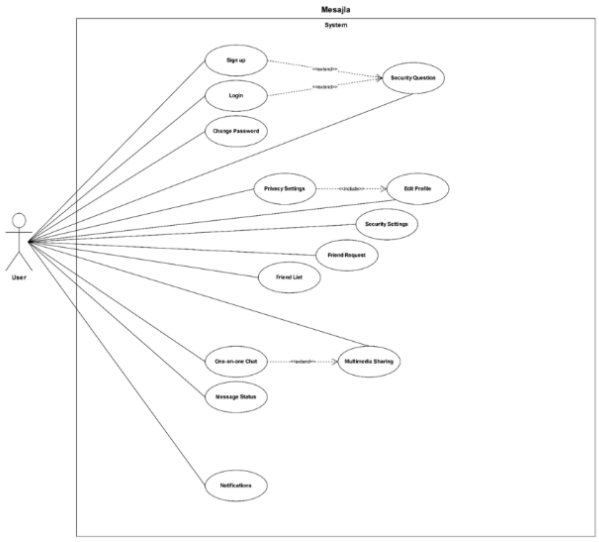

Figure 3: Use Case Model of the application

Design Decisions and Rationale

Several trade-offs and engineering decisions shaped the final design:

- **Modularity over Performance**: We prioritized modular design (like microservices) to make the system scalable and maintainable, even if it adds some performance overhead.
- **Flexibility over Complexity**: The system is designed to be flexible, allowing for easier updates, particularly for the AI module, without adding unnecessary complexity.
- **Accuracy over Efficiency**: Our AI model is trained on large datasets for better response accuracy, which increases training time but delivers higher-quality interactions.
- **Security & Usability**: Authentication is designed to be both secure (e.g., GDPR-compliant) and user-friendly, striking a balance between safety and ease of use.
- **Privacy over Accuracy**: User data privacy is a core principle. We opted to limit data collection from users to protect their privacy, even if it slightly reduces AI response precision.

Final Project Status

At the origin of this project, there are some requirements and fulfillments set for building the application. We aimed to find innovative ways to create a chatting platform that follows trending AI technology. For this reason, it is supported with a generative AI model that analyses the communication history between users and gives suggestions based on the user history. With this, the platform would be distinct from its other familiar communication applications. Also, the lack of such sufficient environments for chatting in our country motivated us to implement this project.

For the UI of the application, there's fairly simple and easy to use design to make chatting between parties smoother. This design offers their users clear and understandable interface efficiently. With the help of utilization and of AI in this interface, the users can get recommendations to enhance their experience. This application also provides protection on user authentication and information, and supports blocking non-familiar or unwanted users, making it secure, thus following the requirements of GDPR.

The platform is still improvised day by day to increase its users' needs. As in this day, it still gets updates to meet the user expectations. One of the primary focuses is to improvise the image and file transformation in the chatting environment. Even though it contains some of the functionalities (file transformation, personal photo of the user etc.), they need to be advanced for better utilization. Also, the system currently isn't built in for Android or iOS. In the future, this is also a goal to achieve.

Impact of Engineering Solutions

The main idea of this project is to build a messaging application and improve it by combining a generative AI model that is becoming more and more popular each day. With the assistance of the model, the user experience and the efficiency of chatting is increased. With this approach, the application takes a new perspective upon the utilization of generative AI. With the important elements such as handling data synchronization and security of the data, Mesajla offers a safe environment for their users to communicate.

Mesajla implements cost-reducing functionalities in storage, model deployment, development and training and many more. This way, it is fairly efficient in terms of resource usage and time utilization.

The application creates an environment that allows their users to socialize. People can express their feelings, emotions and opinions on this platform. During this, the two parties during their communication are supplied with Mesajla about the information of the current dialogue. The personalized model that credits each user a unique approach, so that the suggestions given by the generative AI are according to the conversational flow. This supports faster and more natural chatting experiences.

Contemporary Issues

Current Trends and Challenges

In order to keep up with current technology and keep the user experience at the highest level, the message uses the AI integrated answering option. However, although this makes the application more advanced, it also comes with some challenges. Such as the production of correct and appropriate answers and the time it takes to create these answers. Of course, these challenges will be corrected over time and more relevant and faster AI answers suitable for users will be produced in real time.

Legal, Ethical, and Privacy Concerns

AI-generated answers, showing the user manipulative or inappropriate things, we can add sensitive content to these, do not show ethical behavior. Moreover, generating answers according to the user's history causes privacy issues. This makes it difficult to comply with laws such as GDPR.

Societal Implications

Thanks to "messaging", users can express themselves better in their chats. They become more sensitive and knowledgeable about complex conversations. However, another important issue is that excessive dependence on the AI can cause real communication to weaken. Here, understanding what responsible use looks like and how it is implemented comes into play.

Tools and Technologies Used

Programming Languages

- **JavaScript**: Used with React to build the front-end interface.
- **Python**: Used to develop and train the AI model.

Frameworks and Libraries

- **React.js**: Building responsive and dynamic front-end components.
- **React Testing Library**: Used for writing and executing unit and UI tests.
- **Flask/Flask_CORS**: A framework for deploying the generative AI model in web server
- **PyTorch**: A framework to train and build the neural network model.

APIs and Databases

- **Firebase**: Handles authentication, real-time database (Firestore), and file storage.
- **Custom AI API**: Connects the back-end system with the AI model to provide smart message suggestions in real time.

New Tools and Platforms Learned

- **Firebase Authentication & Security Rules**: Used to manage secure access and permissions.
- **Lighthouse**: Chrome tool used to test app performance and accessibility.
- **GitHub Actions**: Leveraged for test automation and continuous integration.
- **Chaos Testing Techniques**: Simulates real-world failures (like network issues or data corruption) to test system resilience and error handling.

Background and Resource Usage

Use of Library Resources

To guide the development of Mesajla, we referred to various academic resources and technical standards. This included research papers and books on software architecture, software testing, and secure web application development. We aligned our process with key IEEE engineering standards to ensure quality and maintainability.

Internet Research

We also conducted extensive online research throughout the project:

- **Similar Designs**: We analyzed popular messaging platforms like WhatsApp Web, Messenger, and Slack to understand effective UI/UX patterns and real-time chat architectures.
- **Component Information**: Firebase documentation, React community resources, and open-source chat UIs were used to implement critical components.
- **Engineering Principles Applied**:
    - Separation of concerns and modular design across the codebase.
    - Compliance with standards for requirement specification and software architecture.
    - GDPR compliance practices in handling user data and designing secure communication workflows.

Test Results

The application was tested in a total of 27 different scenarios. These are user registration, login processes, adding friends, creating groups, AI response responsiveness and system design. In this way, we aimed to maintain the functionality in real-time chat.

User Registration and Sign-in

TS01: Valid User Registration

- Purpose: Verify that any user has successfully completed the registration process with a valid email, password and security question.
- Steps: Go to the Sign-up page. Enter a valid email and password. Select a security question and enter the answer. Verify that the triangle button is active. Click it to successfully register.
- Status: PASS
- Notes:

TS02: Triangle Button Inactive in Incomplete Form

- Purpose: Verify that the triangle "next" button is inactive until you fill in the information required for registration.
- Steps: Navigate to the registration page, make sure at least one field is empty, check the status of the triangle button.
- Status: PASS
- Notes:

TS03: Redirect to Profile Settings after Registration

- Purpose: After successful registration, verify that the user is redirected to the profile setup page.

- Steps: Complete the registration process, then check if you are correctly directed to the profile page.
- Status: PASS
- Notes:

TS04: Valid Login Process

- Purpose: Verify that a registered user has successfully logged in with the correct credentials.
- Steps: The login page was visited, the correct user information was entered, the triangle button was clicked and the navigation was checked to see if it was successful.
- Status: PASS
- Notes:

TS05: Navigating from Login Page to Registration Page

- Purpose: Verify that clicking the “Sign Up” button redirects to the registration page.
- Steps: Click on the sign up button, check that you are directed to the registration page.
- Status: PASS
- Notes:

Profile Setup

TS06: Profile Settings

- Purpose: Verify that the user cannot continue unless they enter their first and last name
- Steps: After completing the registration process, check the automatic redirection.
- Status: PASS
- Notes:

TS7: Cannot Continue with Empty Profile Fields

- Purpose: Verify that the user cannot proceed unless name and surname are filled.
- Steps: Do not enter input in the name and surname fields, check the triangle button.
- Status: PASS
- Notes:

TS8: Verification of being redirected to the chat page after completing the profile.

- Purpose: Verify that after the name and surname are filled, to navigate to the chat page, the triangle button activates.
- Steps: Enter your name and surname, click the triangle button and check if it goes to the chat page successfully.
- Status: PASS
- Notes:

Real-time Messaging Functionality

TS9: Real-time Messaging

- Purpose: Verification of whether the user can send real-time messages during chat.
- Steps: Two separate accounts were logged in using two different devices. Messages were sent and instant access was checked.
- Status: PASS
- Notes:

TS10: Correct Message information

- Purpose: Verify that users' profile pictures, name, and online status can be viewed.
- Steps: First, a message was sent, then the sender's information was checked.
- Status: FAIL
- Notes: User status did not work.

TS11: Add friends and start a conversation.

- Purpose: Verify that the user can successfully add friends and start a new chat.
- Steps: The add friend button was used and a conversation was started with the added friend.
- Status: PASS
- Notes:

TS12: Correct ordering of messages

- Purpose: Verify the order and timing of messages in the chat interface.
- Steps: Check the order of messages.
- Status: PASS
- Notes:

TS13: Sending Emoji and Files

- Purpose: Verify that emoji and files can be sent without any problems.
- Steps: Send emoji and files in chat. Then check that they are displayed correctly.
- Status: FAIL
- Notes: Cannot send emojis

TS14: Persistence of message conversation

- Purpose: Verify that messaging history across sessions is retained.
- Steps: A message was sent to a user. The session was logged out and back in. Checked if the chat history was still there.
- Status: PASS
- Notes:

AI-generated Message Suggestions

TS15: AI Message Suggestions Appear

- Purpose: Verification that AI-generated message suggestions are displayed during chat.
- Steps: Chatting between users has started. Ai suggestions were checked under the chat writing field.
- Status: PASS
- Notes:

TS16: The selected AI message is sent correctly.

- Purpose: Clicking on a suggested message and verifying that it has progressed successfully in the chat.
- Steps: One of the suggested messages was selected. It was checked whether the message was delivered correctly.
- Status: PASS
- Notes:

TS17: Refreshing AI Message Recommendations

- Purpose: Verify that clicking the cross icon changes the message suggestions.
- Steps: For any suggestion, the cross symbol was clicked. It was checked whether there were new suggestions.
- Status: PASS
- Notes:

TS18: Multiple AI Recommendation Display

- Purpose: Verify that at least 2-3 recommendations are displayed.
- Steps: Enter the chat, check the number of suggested AI messages.
- Status: PASS
- Notes:

TS19: Not selecting suggested AI messages

- Purpose: Testing the system's response to new messages when AI suggestions are not used.
- Steps: AI message suggestions were not used for a few messages. Checked if new AI message suggestions were coming for other messages.
- Status: PASS
- Notes:

Group Chat Functionality

TS20: Creating groups and adding users

- Purpose: Verified that the user can create a group and add other users to the group.
- Steps: Clicked on the "Create group option", selected the members to be added, then checked the group list to confirm the addition of group members.
- Status: PASS
- Notes:

TS21: Real Time Messaging in Group Chat

- Purpose: Verify that users can type and view real-time messages in group chat.
- Steps: A message was sent in the group chat. Checked whether other members can see the messages instantly.
- Status: PASS
- Notes:

TS22: Viewing Groups

- Purpose: Verification that the user can view the list of groups they are a member of.
- Steps: Joined several groups. Checked if all of them appear in the group list.
- Status: PASS
- Notes:

TS23: Leaving the group

- Purpose: Verify that users can leave the group without encountering any errors.
- Steps: Clicked on the “Leave Group” option, then checked if the group still exists.
- Status: FAIL
- Notes: User cannot leave the group.

UI and Navigation

TS24: Correct Behavior of Triangle Buttons that Provide Navigation Between Pages

- Purpose: Verify that the Triangle buttons that navigate between pages behave as they should. (Active = black, gray = inactive)
- Steps: Tests were performed on forms on different pages. After the form was completed, the status of the button was checked.
- Status: PASS
- Notes:

TS25: Responsiveness

- Purpose: Validation of responsiveness on different screen sizes.
- Steps: The application was opened on different devices. The placement of the components was checked.
- Status: PASS
- Notes:

TS26: UI Consistency

- Purpose: Verify consistent UI across pages.

- Steps: Sign-up, Sign-in, Profile Setup, Chat Page visited then colors, fonts and icons compared.
- Status: PASS
- Notes:

TS27: Accuracy of online status updates

- Purpose: Checking whether real-time status updates, online or offline, are reflected correctly to the user:
- Steps: One user was logged out. The other user checked if the other was offline.
- Status: PASS
- Notes:

Overall,

Total Test Cases: 27

Passed: 24

Failed: 3

Bugs Identified: TS10, TS13, TS23

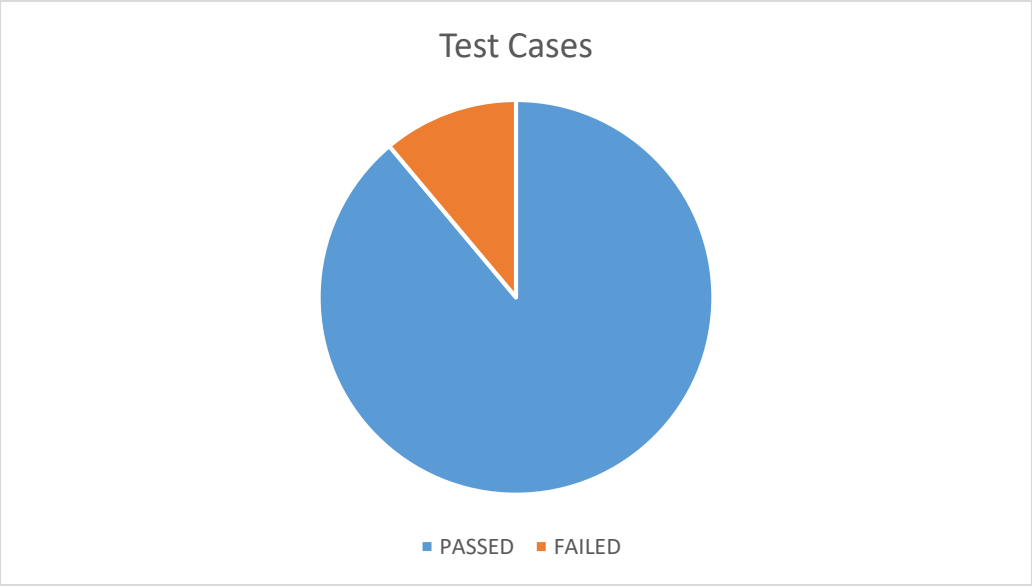

Figure 4: Distribution of success and failure in test cases

Conclusion and Future Work

This project allowed us to improvise on key factors in engineering such as teamwork, planning for architecture and design of the project, research and many more. The structure of databases, front-end, back-end and the generative AI model and connecting them required working collaboratively, and each member in this group tried their best to support this goal. In the end, we managed to create a chatting application that is supported by the generative AI to increase the user experience. By referencing the goals and issues that we mentioned in this report, we aim to enhance the application furthermore in the future.

References

Intersoft Consulting. (n.d.). *General Data Protection Regulation (GDPR) – Legal Text*. Retrieved May 25, 2025, from https://gdpr-info.eu/