|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | Feyzullayev rasul, florea-ioana cristina, sadic gunal, dimitrii grec, codrescu adrian-cătălin. | | Description: Editable Messages : The app will allow users to edit messages even after they have been sent. Scheduled Messages and Reminders : Users will be able to schedule messages for future delivery, as well as set reminders for themselves or friends on specific conversations. Live Location with Journey Tracking : Beyond standard live location sharing, we plan to add a journey tracking feature. Users will be able to see the exact path and estimated arrival time of one user to a certain location. Separate Media Display : Users can send documents and photos in chat, displayed separately from text messages in distinct sections. Image-to-Text-to-Speech : This is an extra feature that we aim to implement If time permits. We aim to allow the user to convert text into images within the chat. Additionally, the text can then be converted into speech, offering an inclusive feature for those who prefer audio messages or have visual disabilities. | | |  | | --- | | KEY features:1.editable messages2.separate media display3.Scheduled messages and reminders4. live location with journey tracking5. extra feature:Image-to-text-to-speech | |

\*\*Software Requirements Specification (SRS) - Instant Messaging App\*\*

Version: 2.0

Date: [15.12.2024]

Authors: Feyzullayev Rasul, Florea-Ioana Cristina, Sadic Gunal, Dimitrii Grec, Codrescu Adrian-Cătălin

---

Table of Contents

\*\*Software Requirements Specification (SRS) for Instant Messaging App\*\*

1. Introduction..................................................... Page 3
2. Workflow Diagram ……………………………. Page 4
3. Gantt Chart ……………………………………. Page 5
4. Database Diagram …………………………... Page 6

5. \*\*Project Backlog, User Stories (INVEST)\*\* .......................................... Page 8

5.1 Editable Messages ......................................... Page 8 - 9

5.2 Scheduled Messages and Reminders ............... Page 9 - 10

5.3 Live Location with Journey Tracking ................ Page 10 - 11

5.4 Separate Media Display .................................. Page 11 - 12

* 1. Image-to-Text-to-Speech ................................ Page 12

6.\*\*Diagrams\*\* ………………….. Page 13 – 20

6.1 Design Patterns …………. Page 21 - 22

7. \*\*Development Tools and Progress\*\*

7.1 Project Structure …………………………... Page 23

7.2 FrontEnd ..................................................... Page 23

7.3 UI/UX Design ............................................ Page 24

7.4 BackEnd …………………………………… Page 25

INTRODUCTION

Description: Editable Messages : The app will allow users to edit messages even after they have been sent.

Scheduled Messages and Reminders : Users will be able to schedule messages for future delivery, as well as set reminders for themselves or friends on specific conversations. Live

Location with Journey Tracking : Beyond standard live location sharing, we plan to add a journey tracking feature. Users will be able to see the exact path and estimated arrival time of one user to a certain location.

Separate Media Display : Users can send documents and photos in chat, displayed separately from text messages in distinct sections.

Image-to-Text-to-Speech : This is an extra feature that we aim to implement If time permits. We aim to allow the user to convert text into images within the chat. Additionally, the text can then be converted into speech, offering an inclusive feature for those who prefer audio messages or have visual disabilities.

WorkFlow Diagram

A diagram of workflow

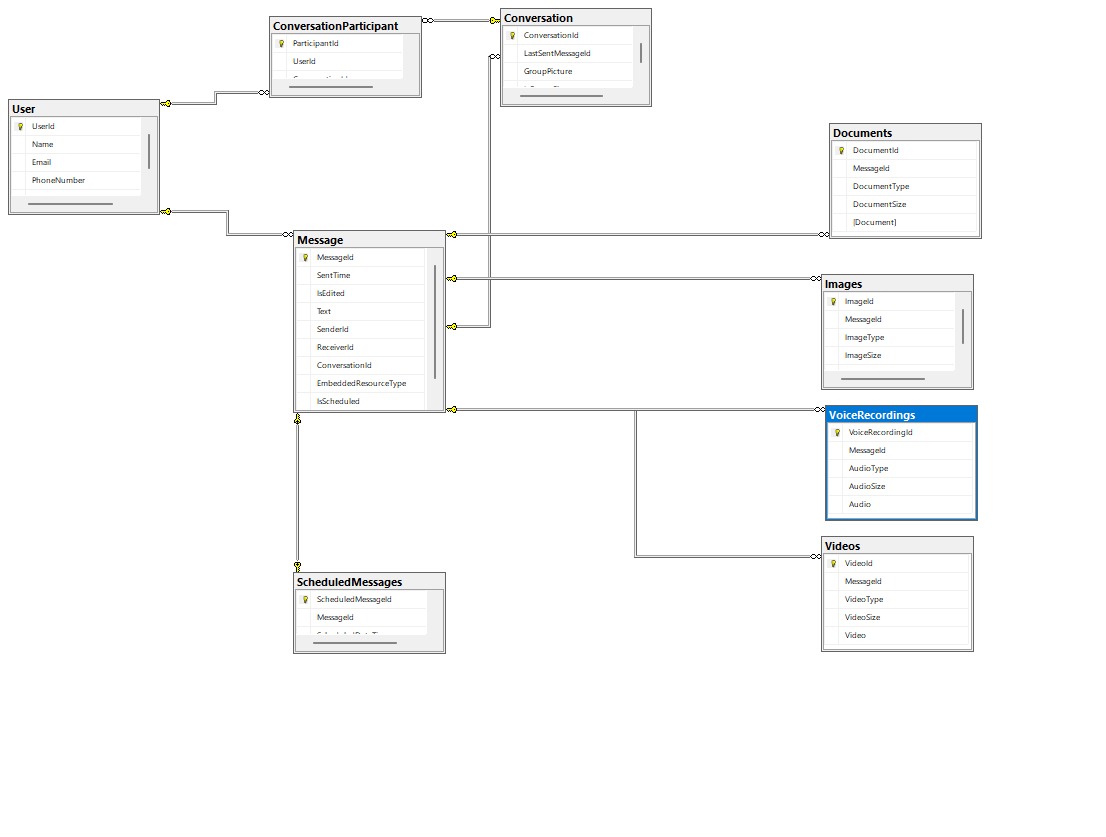
Description automatically generated

Gantt Chart

A graph with colorful squares

Description automatically generated

Database Diagram



User Stories Diagram

A diagram of a diagram

Description automatically generated

This stories identifies:

The user role (user),

The action or feature (edit a sent message),

The benefit or reason (to correct mistakes).

PROJECT BACKLOG

|  |  |  |  |
| --- | --- | --- | --- |
| Priority | User Story Number | Planning Poker Game: Story Points for every User Story | User Story Description |
| High | 1 | 8 | Send and receive messages |
| High | 11 | 21 | Separate section for documents |
| High | 12 | 21 | Separate section for photos |
| Medium | 2 | 13 | Edit sent messages |
| Medium | 3 | 8 | Indication of edited messages |
| Medium | 4 | 21 | Schedule messages for future delivery |
| Medium | 5 | 13 | View all scheduled messages |
| Medium | 6 | 8 | Notification on scheduled message delivery |
| Medium | 7 | 3 | Set reminders within conversations |
| Medium | 8 | 34 | Share live location in real time |
| Medium | 9 | 34 | Allow friends to track journey |
| Medium | 10 | 34 | View route of shared journey |
| Low | 13 | 55 | Convert text images into text |
| Low | 14 | 34 | Convert text to speech |
| Low | 15 | 21 | Take a picture and send immediately |
| Low | 16 | 34 | Send instant voice recording |

“INVEST criteria” (Independent, Negotiable, Valuable, Estimable, Small, Testable) :

Editable Messages

User Stories:

1. As a user, I want to send and receive messages.

INVEST:

Independent: This story is independent.

Negotiable: The team can discuss how messages are sent/received.

Valuable: It provides fundamental value by enabling basic communication.

Estimable: The effort required to implement is easy to estimate.

Small: The scope is small enough to complete in a sprint.

Testable: We can test by verifying message exchange.

Priority : High

2. As a user, I want to edit a sent message so that I can correct mistakes in my conversations.

INVEST :

Independent: Editing can be implemented independently of other stories.

Negotiable: Editable time frames can be discussed (e.g., edit within 10 minutes).

Valuable: Helps users avoid miscommunication.

Estimable: Developers can estimate the effort required.

Small: This feature can be scoped to allow basic editing functionality.

Testable: Testable by confirming users can edit messages within a set time.

Priority : Medium

3. As a user, I want to see an indication when a message has been edited so that I know it has been modified from the original.

INVEST :

Independent: Can be implemented separately from the editing functionality.

Negotiable: The design of the “edited” label is flexible.

Valuable: Adds transparency to communication.

Estimable: Easy to estimate the effort to add an indication label.

Small: Small addition to the feature.

Testable: Testable by confirming “edited” labels display correctly.

Priority : Medium

Scheduled Messages and Reminders

User Stories:

4. As a user, I want to schedule a message for future delivery so that it sends automatically at the right time.

INVEST:

Independent: Scheduling can be developed separately.

Negotiable: Timing options (e.g., specific hours/dates) are flexible.

Valuable: Adds convenience and enhances user experience.

Estimable: Estimable by determining the complexity of scheduling.

Small: Can be scoped to handle simple scheduling.

Testable: Testable by verifying messages send at scheduled times.

Priority : Medium

5. As a user, I want to view all my scheduled messages so that I can manage and adjust them as needed.

INVEST:

Independent: Doesn’t rely on other features.

Negotiable: How messages are displayed can be discussed.

Valuable: Allows users to review and manage schedules.

Estimable: Can be estimated based on display requirements.

Small: Scope is manageable within a sprint.

Testable: Testable by confirming scheduled messages can be viewed.

Priority : Medium

6. As a user, I want to receive a notification when a scheduled message is sent so that I am aware it was delivered.

INVEST :

Independent: Can be built independently of message scheduling.

Negotiable: Notification timing and content can be discussed.

Valuable: Keeps users informed about scheduled messages.

Estimable: Simple to estimate.

Small: Can be implemented within a sprint.

Testable: Testable by confirming notifications work as expected.

Priority : Medium

7. As a user, I want to set reminders within specific conversations so that I don’t forget important events or messages.

INVEST :

Independent: Can function independently of other stories.

Negotiable: Frequency and timing of reminders can vary.

Valuable: Helps users remember important events.

Estimable: Reasonable to estimate based on reminder functionality.

Small: Can be developed in a limited scope.

Testable: Testable by verifying reminders appear correctly.

Priority : Medium

Live Location with Journey Tracking

User Stories:

8. As a user, I want to share my live location with others so that they can see where I am in real time.

INVEST :

Independent: Does not rely on other features.

Negotiable: Options for duration and sharing method can be discussed.

Valuable: Provides useful real-time location info.

Estimable: Can be estimated based on GPS integration.

Small: Can be scoped to handle basic location sharing.

Testable: Testable by confirming location updates work.

Priority : Medium

9. As a user, I want to allow friends to track my journey so that they can see my estimated arrival time.

INVEST :

Independent: Can be developed separately.

Negotiable: Details of journey tracking can be discussed.

Valuable: Improves coordination with friends.

Estimable: Effort can be estimated based on GPS requirements.

Small: Can be developed with minimal complexity.

Testable: Testable by confirming journey tracking accuracy.

Priority : Medium

10. As a user, I want to be able to view the route of a shared journey so that I know the exact path my friend is taking.

INVEST:

Independent: View-only; doesn’t affect journey sharing.

Negotiable: Route details can be customized.

Valuable: Adds visual aid to journey tracking.

Estimable: Effort estimable based on map functionality.

Small: Can be scoped to a basic map view.

Testable: Testable by verifying route display works.

Priority : Medium

Separate Media Display

User Stories:

11. As a user, I want to send documents in a conversation and have them displayed in a separate section so that I can easily find them later.

INVEST :

Independent: Doesn’t depend on other features.

Negotiable: Display style can be discussed.

Valuable: Organizes conversations for easy access.

Estimable: Effort can be estimated based on display requirements.

Small: Scope limited to document display.

Testable: Testable by confirming documents appear in a section.

Priority : High

12. As a user, I want to send photos in a conversation and have them displayed in a separate section so that they are organized and accessible.

INVEST :

Independent: Does not rely on other features.

Negotiable: Display style can be adjusted.

Valuable: Provides easy access to photos.

Estimable: Similar estimate as document display.

Small: Limited to photo display.

Testable: Testable by verifying photo display works.

Priority : High

Image-to-Text-to-Speech

User Stories:

13. As a user, I want to convert images into text within the chat so that I can make my messages visually engaging.

INVEST

Independent : This feature can work independently without depending on other stories.

Negotiable : The team can discuss options for how the text conversion displays and whether there are limitations on image types.

Valuable : Adds visual engagement, enhancing user experience and accessibility.

Estimable : Can be estimated based on the complexity of the image processing and text extraction.

Small : Small enough to develop within a sprint, focusing only on text extraction from images.

Testable : Can be tested by checking if the images are accurately converted to text in the chat.

Priority : Low

14. As a user, I want to convert text to speech so that I can listen to messages instead of reading them.

Independent : This story functions separately from other features.

Negotiable : The team can discuss voice options, language support, and possible limitations.

Valuable : Useful for users with visual impairments or those who prefer listening.

Estimable : Feasible to estimate based on existing text-to-speech tools and integrations.

Small : Can be scoped to basic text-to-speech functionality in a single sprint.

Testable : Testable by verifying that selected messages can be successfully converted to audio.

Priority : Low

15. As a user, I want to be able to take a picture with my camera and send it right away so that I don’t waste time selecting it.

Independent : This functionality can be implemented without dependencies on other stories.

Negotiable : The capture and send process can be discussed to ensure it’s user-friendly and efficient.

Valuable : Adds convenience by allowing users to capture and send instantly.

Estimable : Straightforward to estimate based on existing camera and upload functionality.

Small : Small in scope, involving only camera capture and direct send capability.

Testable : Testable by confirming that a captured photo can be sent immediately without extra steps.

Priority : Low

16. As a user, I want to send an instant voice recording so that I don't have to waste time recording and resending.

INVEST :

Independent: This feature can function separately from other messaging functions.

Negotiable: Options for recording duration or playback control are flexible.

Valuable: Adds convenience for users who prefer quick audio communication.

Estimable: Effort can be estimated based on recording functionality.

Small: Small enough to complete within a sprint.

Testable: Testable by verifying the voice recording functionality works smoothly.

Priority : Low

UML DIAGRAMS

A diagram of workflow

Description automatically generated

This workflow diagram illustrates a messaging app's key processes: user authentication, message scheduling and sending, live location sharing, media sharing, and accessibility features like text-to-image/speech conversion. Each action includes success or failure paths, with notifications sent based on outcomes.

Rasul Feyzullayev

A diagram of a state diagram

Description automatically generated

Rasul Feyzullayev

A diagram of a company

Description automatically generated

Activity Diagram: Cristina Florea

A diagram of a message activity

Description automatically generated

Cristina Florea

Package Diagram

A diagram of a diagram of a company

Description automatically generated

Usecase Diagram

Gunal Sadic

A diagram of a diagram

Description automatically generated

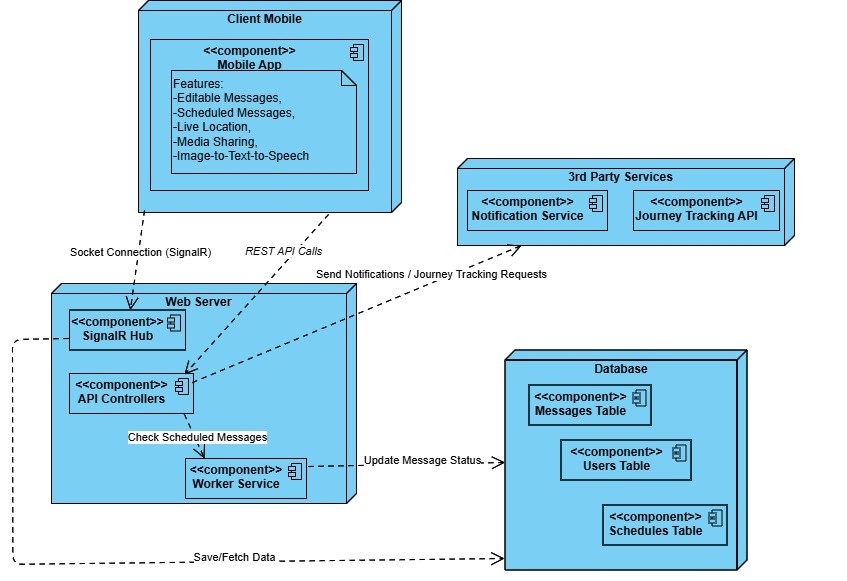
Class Diagram

Gunal Sadic

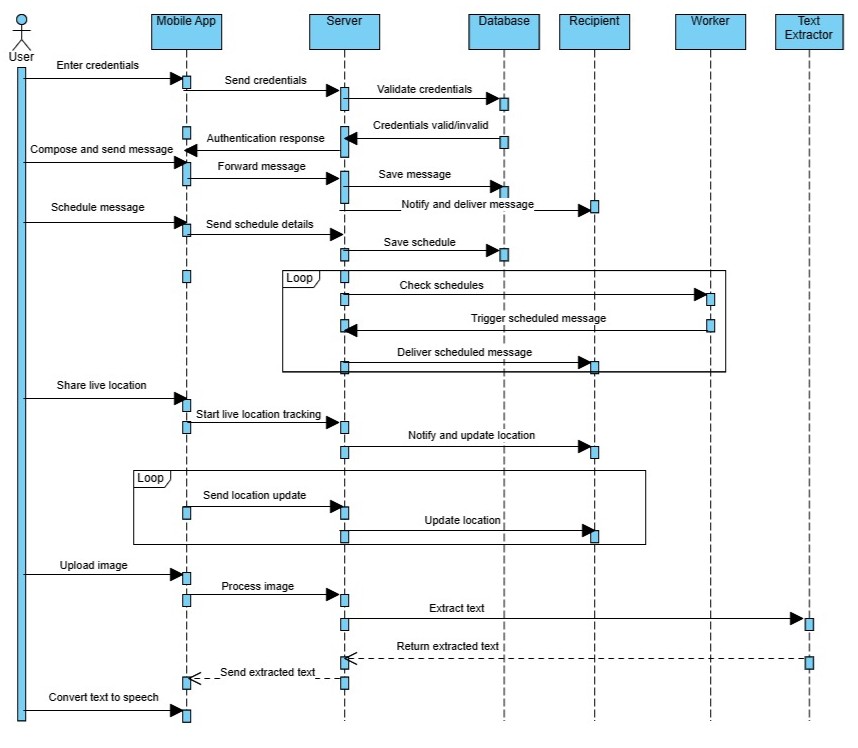
A diagram of a computer

Description automatically generated with medium confidence

Codrescu Catalin Deployment Diagram



Codrescu Catalin Interaction Diagram



DMITRI GREC

A diagram of a user

Description automatically generated

Design Patterns:

Factory Design PatternA screen shot of a computer program

Description automatically generated

A blue square with black text

Description automatically generated

A computer screen with text and images

Description automatically generated with medium confidence

Singleton

A computer screen with text on it

Description automatically generated

A blue screen with black text

Description automatically generated

Development Tools

* Android Studio
* Visual Studio 2022 & Rider
* SQL Server Management Studio (SSMS)
* Flutter, Dart
* SQL Server
* .NET, SignalR
* Figma

\*\*Development Tools and Progress

FrontEnd:

---

We created screens for Forgot password, Validation Code, Create new Password, Profile Picture, Contacts and Conversations. We added validations to the text fields in the forms on these screens.

We made services which execute calls to the back-end. We implemented a Signal-R client which sends and receives messages on sockets.

We've created the layouts for Editable and Scheduling messages and made their respective calls to the back-end. We also stored the jwt token from the login endpoing response in a secure storage.

<https://github.com/FloreaIoanaCristina/Mobile_Messaging_App_FE/blob/main/lib/conversation_page.dart>

---

UI/UX Design

---

To create a seamless and intuitive user experience, we utilized \*\*Figma\*\* to design a comprehensive template for our messaging application. This template prioritizes user-friendliness and modern aesthetics, ensuring the interface is visually appealing and easy to navigate. The design includes essential features such as a \*\*Sign-In page\*\*, \*\*Sign-Up page\*\*, \*\*Password Recovery system\*\*, and \*\*Profile Setup interface\*\*, all crafted with simplicity and efficiency in mind.

We also developed detailed layouts for the \*\*Contacts page\*\*, which enables users to efficiently search and manage their connections, and the \*\*Conversation page\*\*, which supports real-time messaging with clean, dynamic elements.

Each component of the application has been thoughtfully structured to deliver a consistent and responsive experience across devices. The use of modern design principles ensures that the app not only meets functional requirements but also delivers an engaging and visually appealing platform for users. This design represents a significant step toward the app's final implementation and demonstrates a commitment to excellence in both form and function.

---

https://[www.figma.com/design/iLll7IugfNnp3aINactIPF/Proiect-MOPS?node-id=0-1&node-type=canvas&t=wd9k1nyV7mkiWsPL-0](https://www.figma.com/design/iLll7IugfNnp3aINactIPF/Proiect-MOPS?node-id=0-1&node-type=canvas&t=wd9k1nyV7mkiWsPL-0)

BackEnd

---

We created a ChatHub using SignalR which receives and sends messages and saves them in the database. We configured controllers for the api methods and we created a Worker that executes commands regularily to check if a scheduled message is due to be sent or not.

The Hub class was registered in the application as part of this setup, using the following configuration in the `Startup.cs` file:

A black background with white text

Description automatically generated

<https://github.com/GunalSadic/IMBackend>