

CS102**Spring 2023/2024**

Instructor:

Aynur DayanıkProject
Group

Assistant:

Musa Ege Ünalan**1A****~VOYN~****Efsane Milan Kadrosu****Furkan Karademir & Muhammed Emir Eren & Selman Şahin &
İzzet Yağızefe Sungurlu & Alp Eren Köken**

Criteria	TA/Grader	Instructor
Presentation		
Overall		

Detailed Design Report

(version 2.0)**18 May 2024**

1. Introduction

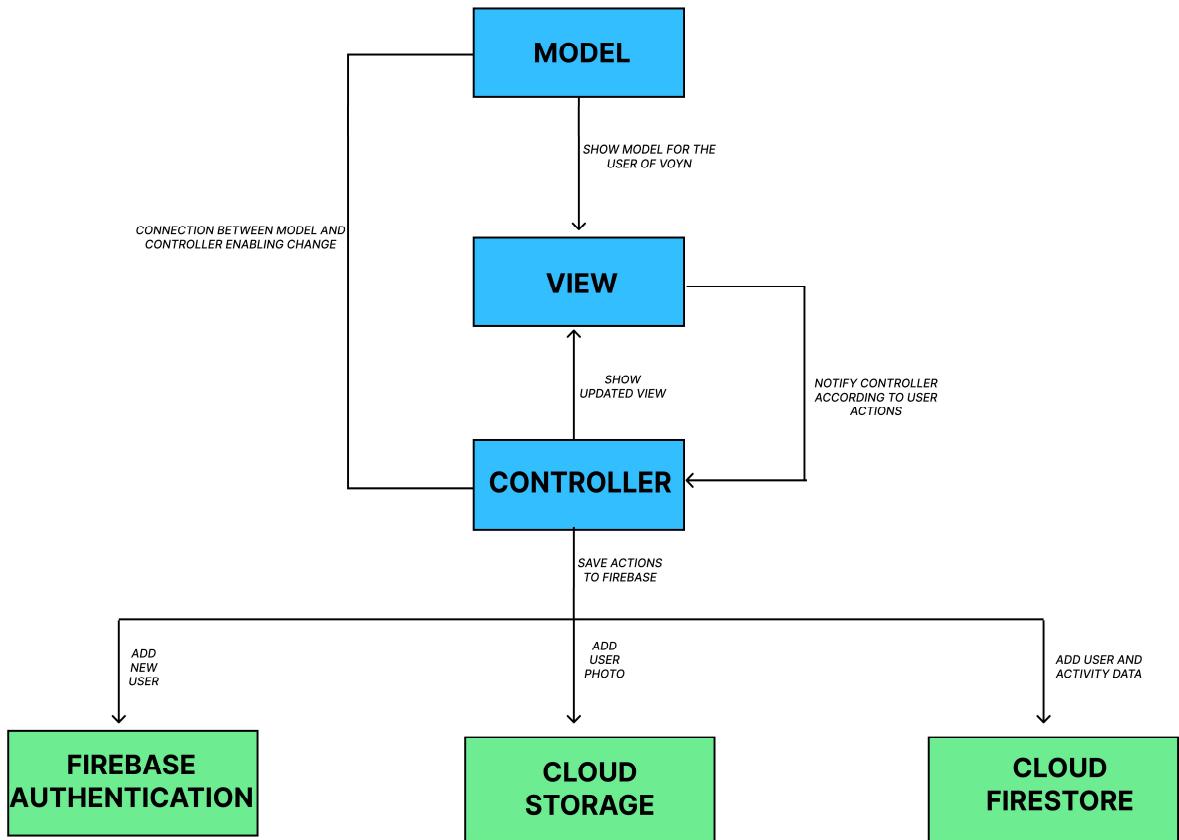
VOYN is an Android application that acts as a bridge between people, especially university students, who have similar interests and purposes. It offers solutions to today's three notable fields transportation, accommodation, and sports by providing easiness to connect people. It enables people to find or create an activity that meets their needs. It also enables them to communicate on chat to deal with. The most outstanding feature of the VOYN is that it brings such a complex and important area to users in one application with an interface that is as simple and easy to use as possible. With all these features, VOYN is a candidate to be an indispensable part of university students' daily lives with its potential benefits.

2. System Overview

2.1 Basic Organization/ Architecture of VOYN

Voyn will be a mobile application. It will be only available for mobile devices that have an Android operating system. There will be an online database to provide interactions between users and we will use Android Studio which is a software of Google to help Android developers.

2.1.1 Architecture Diagram



2.2 Database

Firebase¹: Firebase is a NoSQL database backed by Google. It helps app and game developers build a large-scale and stable product, so they can get to market faster and securely. It is hard to set up a database in a short time considering the security and stability aspect of it as a group of freshmen. Therefore, we are planning to utilize Firebase since our application especially depends on the interaction between different users.

2.2.1 Firebase Authentication

Firebase Authentication² allows developers to implement a secure, stable, and fast authentication system that the users can easily sign in from different platforms and devices. It provides a secure solution supporting email and password accounts, phone numbers, signing in with different social media accounts, and signing in with a Google account. We are planning to use email and password accounts for our app. Also, firebase authentication provides a forgot password helper. If a user forgets their password, we will send a mail using the Firebase authentication system and the user will directly change their password using Firebase's changing password page.

2.2.2 Cloud Firestore

Firebase's cloud Firestore³ is a NoSQL database that helps developers easily store, sync, and query data for their applications. It lets the developers structure data the way they like using collections and documents. Developers can sync their data across devices either online or

¹ “Firebase” <https://firebase.google.com/> (accessed April 26, 2024).

² “Firebase Authentication” <https://firebase.google.com/products/auth> (accessed April 26, 2024).

³ “Cloud Firestore” <https://firebase.google.com/products/firestore> (accessed April 26, 2024).

offline. We will frequently use Firebase's Cloud FireStore in our application. Our application has complex objects to be stored in the FireStore. For example, transportation objects, accommodation objects, and sports objects. These will all have more than one ArrayList in them and all ArrayLists will have to sync every time a new user attends to an activity. It would be hard to design a database that can easily and securely sync our data. For our chat system, we will also utilize FireStore. There will be a lot of different collections in the FireStore and every two users who have chatted in the past can see their past messages, to do this we have to store so much data in the database and call them frequently. Also, for synchronizing the chat every time a user sends a new message, we will use FireStore's listener which is provided by FireBase.

2.2.3 Cloud Storage

FireBase's Cloud Storage⁴ is for saving user's photos and videos easily and it uses strong user-based security. We will need this service to store the profile pictures of our users.

2.2.4 Firebase Crashlytics

FireBase Crashlytics⁵ is a real-time crash report tool that will help us resolve the bugs of our app when our app is used by a lot of people at the same time. This service helps the developers to get crash notifications from real users with the activity report of the user to see which actions led to that crash. We are planning to upload our app to stores, so this service can help us debug our app.

2.3 Android Studio

The user interface and front-end for VOYN, an Android social media application will be created using Android Studio⁶, a powerful integrated development environment (IDE) specifically designed for Android application development. Android Studio provides an extensive array of tools and resources, empowering us to create an intuitive and visually appealing user interface that aligns with VOYN's branding and user experience goals. Our main target will be on employing Android's built-in UI elements, such as layouts and views, to create screens that display a variety of features such as sign in and sign up pages, chat, listed activities, activity details, profile page, and menu page. We plan to adjust the components included by VOYN in order to improve the overall user interaction with our app. Also, we will utilize the layout editor within Android Studio to create visual designs for XML-based layout files, which help to specify the layout and arrangement of the UI components. Furthermore, making use of Android Studio's capabilities for user interface and front-end development ensures that we can benefit from the powerful tools and resources, resulting in the launching of a very detailed and useful Android app for the partner needs of users for sports, accommodation, and transportation.

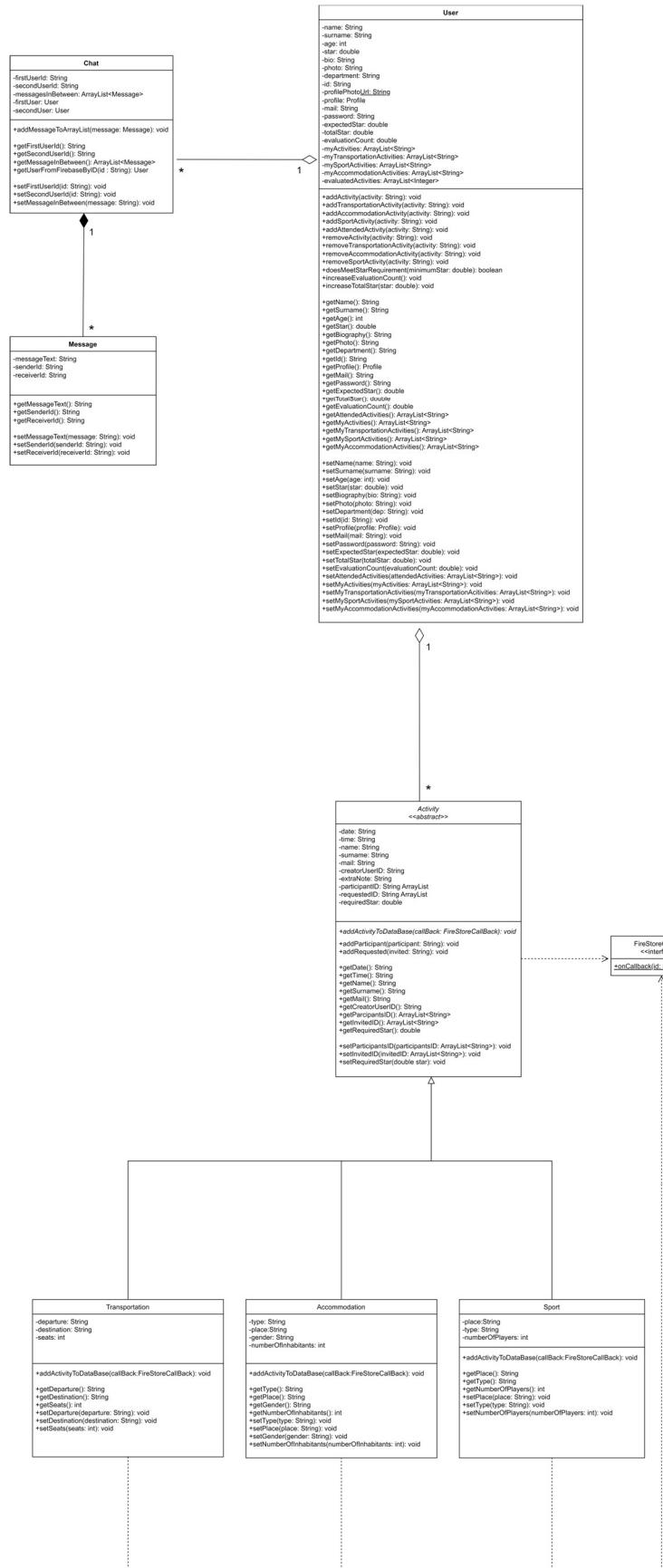
⁴ "Cloud Storage" <https://firebase.google.com/products/storage> (accessed April 26, 2024).

⁵ "Firebase Crashlytics" <https://firebase.google.com/products/crashlytics> (accessed April 26, 2024).

⁶ "Android Studio" <https://developer.android.com/studio> (accessed April 26, 2024).

3. Core Design Details

3.1 UML Class Diagram



⁷ Interface will not be implemented by other classes, we will get it as a parameter. Explained in section 3.2.7

3.2 Model Classes and Interfaces

3.2.1 User

The User class will be responsible for representing users who use the app. It will store various information about the user, such as their name, surname, email, password, age, biography, profile photo, department, ID, star, and profile. Additionally, the User class also will store the user's currently created activities as separate lists of accommodation, transportation, and sports activities. In this class, we have eight important methods named addActivity, addTransportationActivity, addAccommodationActivity, addSportActivity, removeActivity, removeTransportationActivity, removeAccommodationActivity, and removeSportActivity.

- **addActivity:** This method will add any type of activity to the array named myActivities which includes the activities that the user created.
- **addTransportationActivity – addAccommodationActivity – addSportActivity:** These methods will make similar work. In other words, they will add created transportation activity to myTransportationActivities, created accommodation activities to myAccommodationActivities, and created sport activities to mySportActivities, respectively.
- Other methods which include "remove" in its name will do just the opposite. They will remove an activity from the related array.
- **doesMeetStarRequirement:** This method will determine whether the user has a star above the required minimum star for activity or not.

All these methods will enable us to collect the activities that the user opened and delete them when the user wants to delete this activity.

3.2.2 Chat

The Chat class represents a conversation between two users, identified by their stored unique IDs, and includes Message objects that contain the actual content of the conversation. Whenever a user initiates a new conversation with another user, an object of this class is created to manage the conversation and facilitate message exchange. In this class, we will have one important method named addMessageToArrayList.

- **addMessageToArrayList:** This method will enable us to add a message that is sent to the ArrayList named messagesInBetween consisting of messages between two people.

3.2.3 Message

The Message class is designed to serve as a data structure that represents a single message between two users. It will store the unique IDs of the sender and receiver users, and the message content that was written by the sender.

3.2.4 Activity

Activity class will be an abstract class. It will have different instance variables of an activity such as date, time, requested star, ID of the creator and some information about the creator user. In this class, we will have one abstract method named addActivityToDatabase.

- **addActivityToDatabase:** This method will enable us to save new activity that is opened by any user to FireBase. It will be used by Transportation, Accommodation, and Sport classes since they will extend Activity class. This method will work according to the specified class. For example, in the Transportation class addActivityToDatabase method will save transportation activity to the database. There is a similar situation in Accommodation and Sport classes.

Additionally, we will have addParticipant method in that class.

- **addParticipant:** This method will enable us to save participants who are accepted for an created activity by the creator to the Firebase.

Similarly, we will have addInvited method as well.

- **addInvited:** This method will enable us to save participants who send requests to the owner of the activity to the database.

Thanks to those methods, we will handle the participation and request events.

In addition to the instance variables of the activity class, it will have the departure point and the destination point, and the available seats number of the activity. There will be a particular addActivityToDataBase implementation for transportation in order to store them in the correct place with the correct field names.

3.2.5 Accommodation

Accommodation activities class will inherit from activity class. In addition to the instance variables of the activity class, it will have the place, the gender, the type, and the number of inhabitants of the activity. There will be a particular addActivityToDataBase implementation for accommodation activities in order to store them in the correct place with the correct field names.

3.2.6 Sport

Sport activities class will inherit from activity class. In addition to the instance variables of the activity class, it will have the place, the type, and the number of players of the activity. There will be a particular addActivityToDataBase implementation for sport activities in order to store them in the correct place with the correct field names.

3.2.7 FirestoreCallback

The Firestore Callback interface will be implemented by us. It has only one method which is onCallBack. Our transportation, accommodation, and sport classes will not extend this interface. We will just get an instance as a parameter. We will utilize polymorphism when we are storing the activities in the database and get the creator user ID easily. Using this type of interface is common among Android developers. We are trying to create a modern and efficient Android application that an expert in Android development would create.

3.3 How did We Use MVC

In the design of VOYN, we did our best to follow the model-view-controller pattern. In the model part, we designed model classes named Chat, User, Profile, Activity, Transportation, Accommodation, and Sport all of which represent an object in real life. For example, the User class represents people. In the view part, we will have XML layout files displaying data in Android Studio. What we do will be shown thanks to those XML files. Additionally, we have Adapter classes in Android Studio representing the controller part. Those adapter classes enable us to update the view using model classes. In other words, adapter classes as controller plays a connection role between model and view.

4. Task Assignment

UI DESIGN	Alp Eren Köken, Furkan Karademir, İzzet Yağızefe Sungurlu, Muhammed Emir Eren, Selman Şahin
FRONT-END	Alp Eren Köken, Furkan Karademir, İzzet Yağızefe Sungurlu, Muhammed Emir Eren, Selman Şahin
DATABASE	Furkan Karademir, Muhammed Emir Eren
TRANSPORTATION	Furkan Karademir, Muhammed Emir Eren
ACCOMMODATION	Selman Şahin
SPORTS	İzzet Yağızefe Sungurlu
CHAT	Furkan Karademir
PROFILE	Alp Eren Köken
MY ACTIVITIES	Muhammed Emir Eren
STARS	Alp Eren Köken, Furkan Karademir, Selman Şahin
FILTERING	İzzet Yağızefe Sungurlu, Muhammed Emir Eren

5. Summary & Conclusions

5.1 Conclusion

In conclusion, we use some technologies like Android Studio as an application development platform and Firebase as a database providing authentication, cloud storage, and analytics while implementing VOYN. With the help of Android Studio, we create a spectacular and appealing interface in front-end development. Additionally, thanks to Firebase, we will ensure real-time updates in activities and chat. While we are implementing VOYN, we will create Chat, Message, User, Profile, Activity, Transportation, Accommodation, and Sports as model classes. Briefly, VOYN will be an Android application aiming to connect people in the areas related to transportation, accommodation, and sports by creating an environment that minimizes the risk of cluttering thanks to easily understandable designs, all of which are provided thanks to well-designed structure and wisely used technologies.

5.2 Concern's Regarding VOYN's Core Design and Technologies

For beginners, the complexity of Android Studio can be challenging, requiring more time and effort to learn it. Also, for machines with limited RAM, Android Studio may give slower performance and longer built times. Moreover, the built-in Android emulator in Android Studio can also be slow and resource-intensive, making Android Studio challenging to test apps efficiently. Finally, creating complex layouts in XML files can sometimes be challenging and prone to errors and mistakes, for developers who are not proficient in XML or UI design principles.

5.3 What parts We Couldn't Achieve?

Looking at our reports and final project, we can say that there is no part that we could not achieve thanks to our discipline, and planning.

5.4 What did you dislike about project work?

In the context of such an Android project with Firebase integration, one common dislike might be dealing with unexpected bugs or compatibility issues. Debugging can be time-consuming and challenging, subsequently when it affects the critical features of the project. Also, designing comprehensive XMLs might demand hard effort and time.

5.5 What was the most difficult aspect of it?

The most difficult aspect of the project could be ensuring seamless real-time chatting. Implementing features like chat functionality involves handling synchronization, message ordering, and addressing edge cases such as network interruptions or concurrent updates. Also, evaluating users with stars is quite challenging to design and fix the issues of its code. Subsequently, using Firebase in the project requires careful planning and understanding of databases.

5.6 What would you do differently if you had to start over again?

Such an application including diverse features and XMLs may demand more testing. If we had to start over, the functionality of the features may be tested comprehensively.

5.7 How much time did you spend on it?

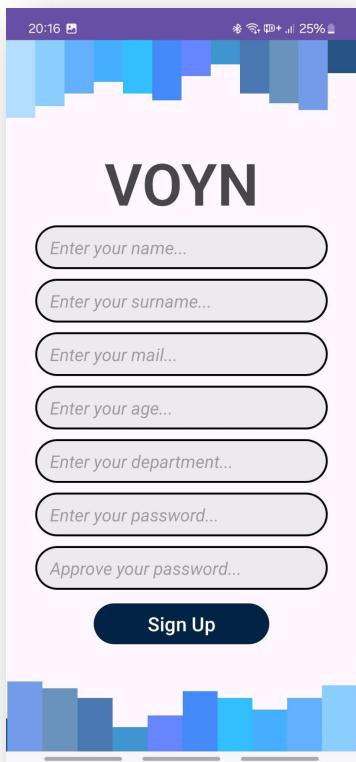
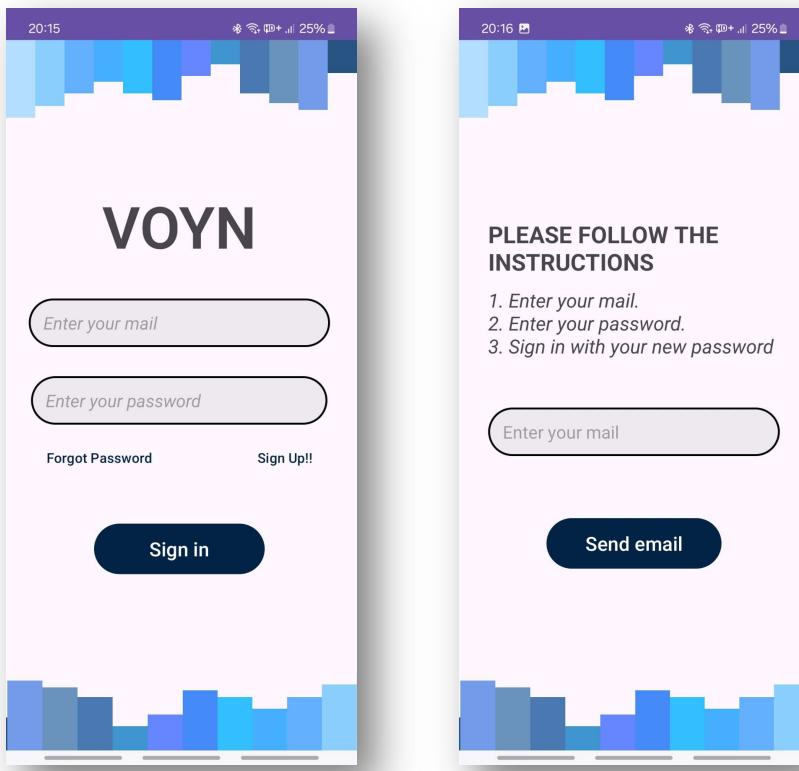
The idea of the project was founded in January 2024. The work on the application has also started in that period. We have worked on this project for five months. As hours, considering the learning process of Android development, the time we spent on reports, and the implementation of the project which we started to try implementing our features earlier in February, we can say that we have spent approximately 2 hours daily on this project. In total, approximately 200 hours are given to this project for each person in the group.

5.8 Are you proud of what you have achieved?

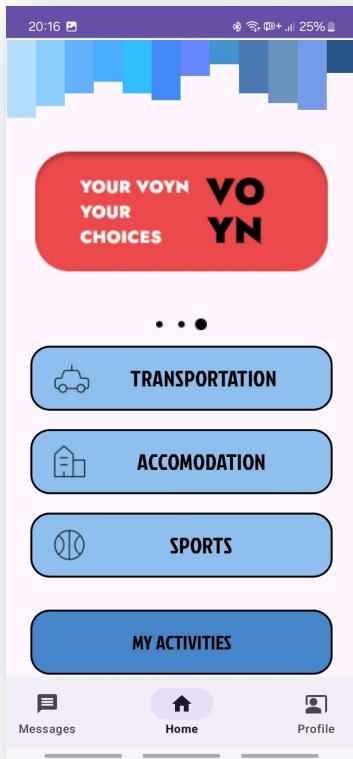
All the previously promised features of the application work smoothly. Moreover, the application is sufficiently detailed and quite useful. We spent great effort and time to complete the project. Overall, this project satisfied our expectations and we are proud of our achievements.

6. Screenshots

6.1 Sign In and Sign-Up Pages



6.2 Menu Page



6.3 Activity Pages

The image contains two side-by-side screenshots of the "Opened Activities" page. Both screens show a list of activities with columns for Time, Date, Person Limit, Where From, and Where To. A legend at the top indicates that a red square means "LOCKED".

Screenshot 1 (Left):

Time	Date	Person Limit	Where From	Where To
09.05	2024.05.30	0/3	Ankara	Antalya
10.10	2025.05.05	0/4	ankara	istanbul
14.00	2025.07.07	0/3	ankara	izmir
09.27	2024.05.15	0/3	Ankara	Istanbul
13.00	2024.05.30	0/2	Konuva	W

Screenshot 2 (Right):

Date	Place	Person Limit
2024.07.05	Dorm	91
2024.10.10	bahceli	0/3
2025.09.09	78	0/3



6.4 Activity Details Pages





6.5 Create New Activity Pages

The image displays two mobile application screens for creating new activity pages, side-by-side.

Left Screen (Transportation):

- Section:** OPENING A TRANSPORTATION ACTIVITY
- Fields:**
 - DEPARTURE POINT (with placeholder "Enter the departure point...")
 - DESTINATION (with placeholder "Enter the destination...")
 - DATE (with placeholder "Enter the date in the form Year/Month/Day")
 - TIME (with placeholder "Enter the Time")
 - NUMBER OF AVAILABLE SEATS (with placeholder "Enter the number of available seats") and MINIMUM STARS (with placeholder "Min stars")
 - EXTRA NOTES (with placeholder "You can add extra notes...")
- Buttons:** OPEN ACTIVITY (with a checkmark icon) and CANCEL

Right Screen (Accommodation):

- Section:** OPENING A ACCOMMODATION ACTIVITY
- Fields:**
 - TYPE (with placeholder "Enter the type...")
 - PLACE (with placeholder "Enter the place...")
 - DATE (with placeholder "Enter the date in the form Year/Month/Day")
 - GENDER (with placeholder "Enter the gender you want to live together...")
 - NUMBER OF INHABITANTS (with placeholder "Enter the number of inhabitants...") and MINIMUM STARS (with placeholder "Min stars")
 - EXTRA NOTES (with placeholder "You can add extra notes...")
- Buttons:** OPEN ACTIVITY (with a checkmark icon) and CANCEL

20:17 25%

SPORT

OPENING A SPORT ACTIVITY

TYPE

Enter the type...

PLACE

Enter the place...

DATE

Enter the date in the form Year/Month/Day

TIME

Enter the Time

NUMBER OF PLAYERS

Enter the number of available seats

MINIMUM STARS

min stars

EXTRA NOTES

You can add extra notes...

OPEN ACTIVITY

CANCEL

6.6 Filter Pages

20:16 25%

TRANSPORTATION FILTER

DATE:

May 2024

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

TIME: Enter the time...

FROM: Type departure point...

TO: Type destination point...

AVAILABLE **LOCKED**

APPLY FILTER

20:17 25%

ACCOMMODATION FILTER

DATE:

May 2024

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

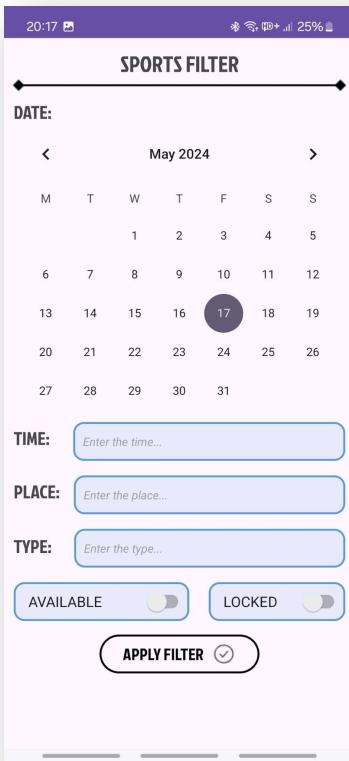
TYPE: Enter the type...

PLACE: Enter the place...

GENDER: Enter the gender...

AVAILABLE **LOCKED**

APPLY FILTER



6.7 My Activities Pages

20:17 25%

MY ACTIVITIES

CLICK ON YOUR ACTIVITY TO SEE INCOMING INVITATIONS

Past Selected Created

CLICK ON TO SEE PARTICIPATED ACTIVITIES

CREATED TRANSPORTATION ACTIVITIES

Time 12:00	Date 2024.09.12	Person Limit 0/12
Where From Kayseri	→	Where To Siwas

CREATED SPORT ACTIVITIES

Football	2025.01.01	
Time 16:30	Place East campus	Person Limit 0/5
Tennis	2024.08.01	

CREATED ACCOMODATION ACTIVITIES

Date 2024..08.16	Place Bilkent Main Campus	Person Limit 0/Z
Gender: Male		

20:17 24%

PARTICIPATED TRANSPORTATION ACTIVITIES

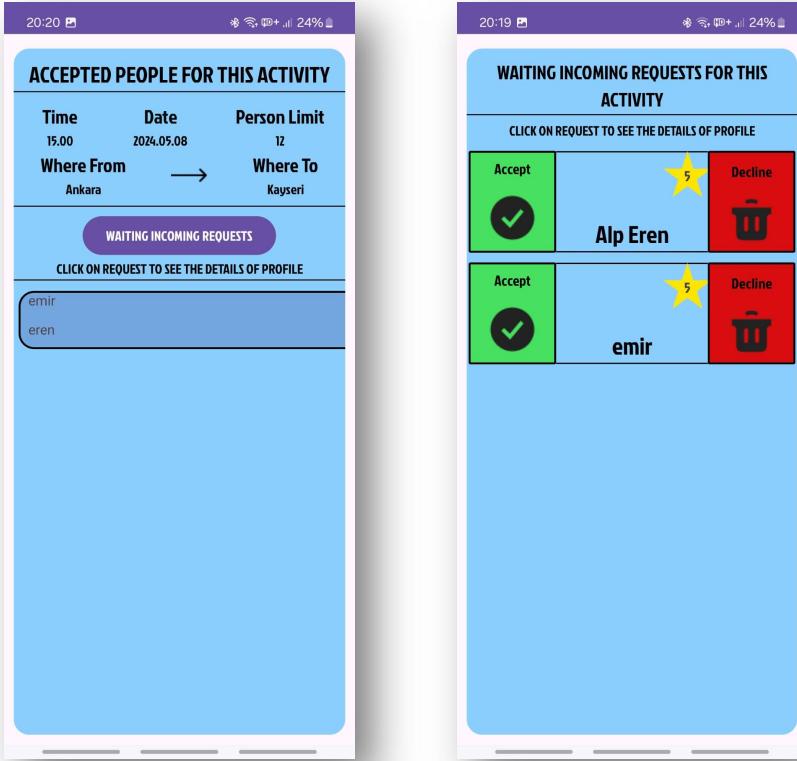
Time 12:00	Date 2020.01.01	Person Limit 1/4
Where From Ankara	→	Where To Istanbul

PARTICIPATED ACCOMMODATION ACTIVITIES

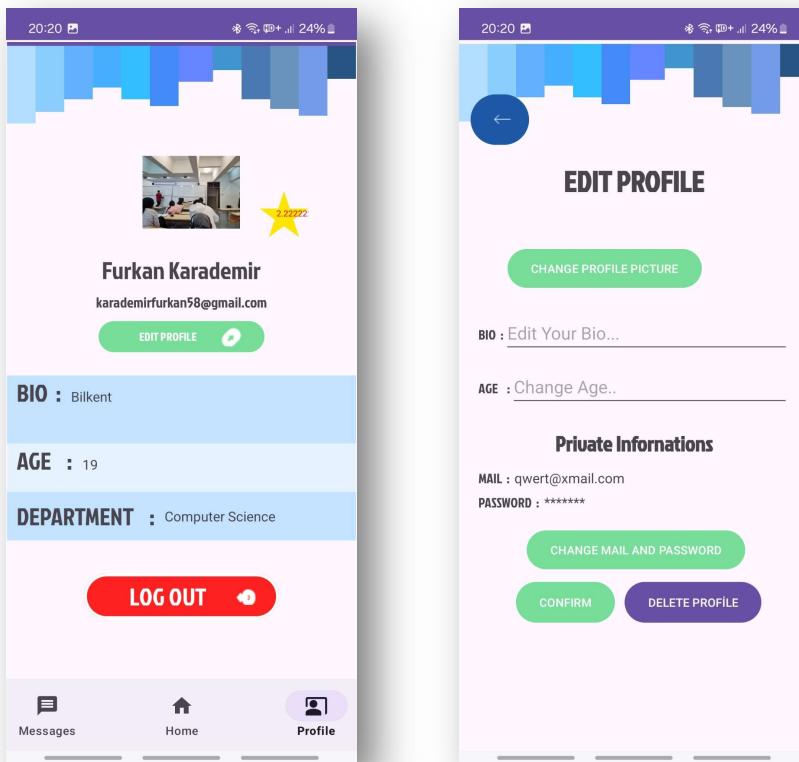
PARTICIPATED SPORT ACTIVITIES

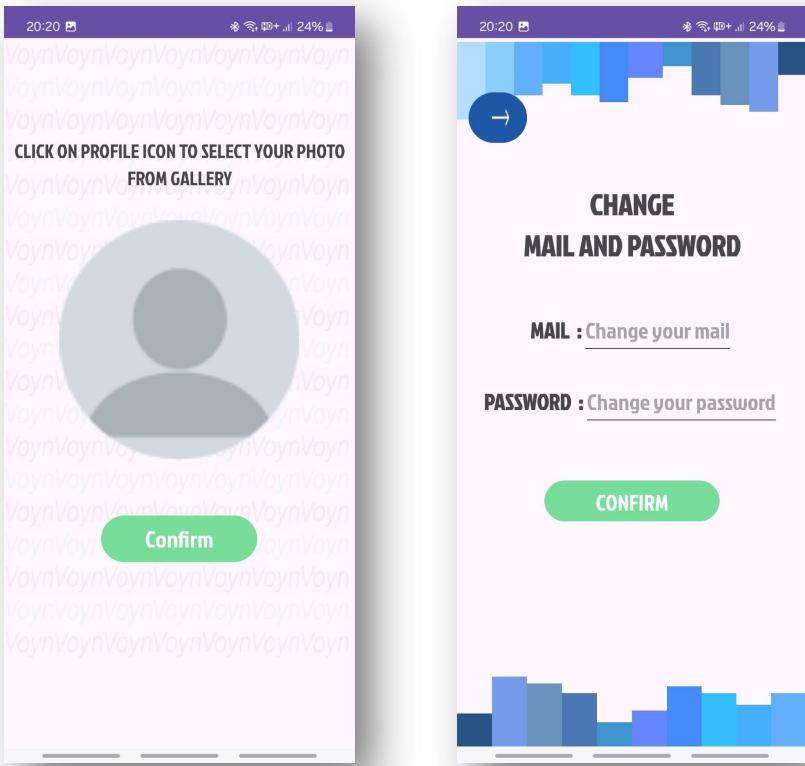
Football	2025.04.03	
Time 13:00	Place Main	Person Limit 1/9

6.8 Incoming Requests Pages

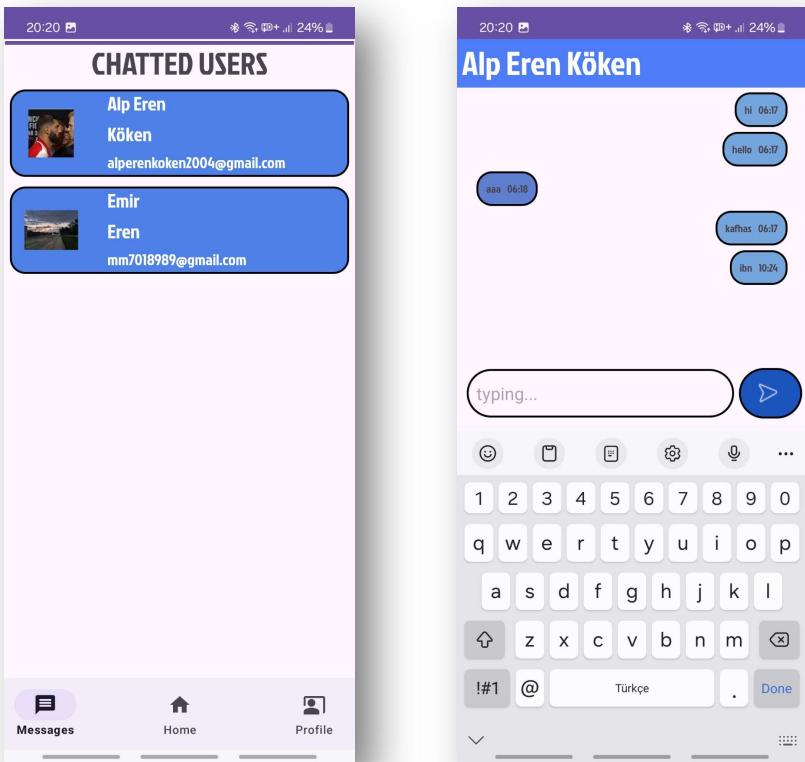


6.9 Profile and Edit Profile Pages





6.10 Chat Pages



6.11 Profile Page For Others

