Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: HandsGiving

Final Report

Atakan Arslan, Oğuzhan Dere, Ege Hakan Karaağaç, Mehmet Tolga Tomris, Fırat Yönak

Supervisor: Özgür Ulusoy

Jury Members: Selim Aksoy, Ayşegül Dündar

Final Report

April 30, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1 Introduction

With the increase in the number of people in need in the world and the extension of human life, solidarity has become more important. People started to need each other more both materially and spiritually. However, the distance between people and their inability to communicate easily with each other made it more difficult to cooperate. HandsGiving is an application intended to be a solution to all these problems faced by people. It will provide a platform for people in need of help to have their voices heard. Those people who are in need will have the opportunity to express their needs to volunteers through the application. The application will also be designed to reduce the negative effects of the COVID-19 virus on people's social life [1]. As part of COVID-19 measures, elderly people had to be restricted from going out to the streets [2]. Both the elderly who were left alone in their homes due to these restrictions and those who already had fewer people around became more in need for social activities and their needs became more difficult to meet. Thanks to the video chat feature that HandsGiving will offer, people will gain the chance to meet with different volunteers and meet their social needs. HandsGiving is going to be a free application, which has the sole purpose of connecting people of any age with each other for various reasons. The application feeds on the mutual benefits of the communal life, which leads to a win-win situation rather than what most applications were created for. It is also worth mentioning that the scope of this application is even more than social concerns.

2 Requirements Details

2.1 Functional Requirements

In this section, functional requirements will be discussed. Functional software requirements help you to capture the intended behavior of the system. This behavior may be expressed as functions, services, or tasks or which system is required to perform [3]. As part of functional requirements, we will discuss system requirements, storage of application data requirements, and user-specific requirements.

2.1.1 System functionality

The system should:

- exist for any possible fraud action.
- receive user type, name, surname, mail address, password and address of the user as inputs for the registration process.
- send a confirmation mail to the new users for account confirmation.
- ask the user's permission to access the user's current location.
- receive mail address, password as inputs for the authentication.
- receive location, type of the help request, explanation of the help to create a new help request for the needy users.
- show available video chat requests to the users on the social platform of the application.
- keep the past conversations between two users through the chat and display these messages.
- add automatically the two sides of a help process as friends.
- have a list of help requests with filter parameters such as social/financial to help benevolent users.
- include the personal information of a user in order to make it easier for other users to decide whether they want to help or not.
- be available on the application store of the according platform for free.
- be able to show the previous helping activities of a user in that user’s profile.
- be able to maintain and view the reviews acquired by a user.

2.1.2 User functionality

- Users should be able to log in/log out or register through the application.
- Users should be able to create their own profiles and publish their personal information.
- Users should have the option to hide their personal information which is not required to show.
- Users should be able to create video chat requests.
- Users should be able to chat with their friends.
- Users should be able to add friends.

- Users should be able to download the application from the application store.
- After a request is accepted and completed, both sides of the requests should be able to give a review which includes a rate out of 5 and a short comment consisting of a few sentences (i.e. maximum of 300 characters, this value can be increased depending on the users' demand over time.). Users should be able to view the reviews of users in their profiles.
- Needy users should be able to create help requests either socially or financially.
- Needy users should be able to list their own requests and cancel them if they are not active or removable.
- Benevolent users should be able to see the help requests which are near to their location in the application.
- Benevolent users should be able to help any user who has a specific help request.
- Benevolent users should be able to see the detail of a specified request by clicking a detail button.
- Benevolent users should be able to accept or ignore a help request. They do not have to accept every response to a help request.

2.1.3 Storage of the Application Data

- The application data should be stored in a relational database with efficient relations of entities, for easier maintainability of the data.

- The property data of each help request should be kept on the server. The property data includes information such as the content of the request, request creator user mail, and sometimes even the request location coordinate information.

- Each user information with each user-mail should be kept in the server, same as the help requests.

- Other important data that will be present in user profiles such as review data should also be kept separately with mail addresses in the mentioned relational database.

2.2 Non-functional Requirements

2.2.1 Usability & Accessibility

- The application should be available on an application store since the user community of the program should be able to contact each other by downloading the application over an application store.
- The application should have an option that allows an easier interface for elders. This option can be used by users who are not old also.
- The easier interface should have bigger text boxes that are easier to touch and activate or have a zoom-in capability.
- The request creation section should be clear, and it should be on the main page of the application.

2.2.2 Reliability of the System

- The application should not mix users during the financial or social support of a benevolent. Tests for this requirement must be made.
- The application must have an error handling system that determines the reason for any error and notifies the user accordingly.
- The server of the application should be capable of working as expected in the scope of the maximum number of users.
- The maximum number of users should be fixed to 10000 at first, the number can be increased if the number of users gets close to the maximum amount.

2.2.3 Reliability in between the Users

- The email verification system should exist in the system for the purpose of lowering the number of imposter users in the application. The review system also strengthens the reliability between the users.

2.2.4 Privacy and Security of the User Information

- The GPS information should be kept private and protected from any possible abuse. GPS information should not be used without consent.

- The server must be secure enough for any possible threat by a client.

2.2.5 Efficiency

- The application should be able to run smoothly in older generation smartphones since the user community of the application will be including financially suffering people.
- The application's total file size should not take more than 100 MB in order to make it easier to store and access. Redundant files incurred by events such as application updates should be removed immediately.
- The application should be able to detect users who have not been using the application for a long time period and delete the accounts within their knowledge for a better server capacity efficiency, in case of any unexpected user number increase. This can be done by sending a mail to their mail address or by sending a notification message to their phone number if it is registered in the system. The users should be able to stop the deletion progress.

2.2.6 Extensibility

- The object documentation of the application should be systematic and open to any upgrades such as new functional/non-functional requirements or user interface changes.

- The resource codes of the application should be well-commented and easy to understand for newer employees and for the sake of the application.

2.2.7 Maintainability

- The application should have its modules correctly separated, such that any update or change in the application should not interfere with any other non-related module.

- The server should be built such that it should be possible to increase the maximum user number without changing the whole system.

2.3.8 Recovery of the Data

- The application should have a recovery system such that any client data should not be lost when the servers are down. Any unsavable data should be notified to the corresponding user and the server should be recovered back to its latest save state.

2.3.9 Legality

- The application should notify the user in the process of registration that the application does not take any responsibility for any illegal event performed by the users and on any illegal events' consequences on any user.

- The application should not share any private user information with third parties

2.3.10 Performance

- The creation of a request should not take more than 5 seconds when there are no existing errors. If it takes more than 5 seconds, the user must be notified with a reason.
- The starting process of the application should not take more than 10 seconds. Since impatience of users may lead to losing them.

3 Final Architecture and Design Details

3.1 Client-Server Architecture

The target of HandsGiving is to bring needy and benevolent users to a common point; that results in many information flows between users. Information like personal information and help processes needs to be saved and displayed whenever a user wants to see it. One of the main features of the app is a video chat that requires constant data flows. All of these data needs to be collected in a place and shared from there. As a result of these reasons, client-server architecture is chosen. The server handles the back end works like saving the data, managing the data, and responding to the requests. On the client-side, the application displays the data and enables the users to interact with other people and take input from the user to handle requests of the users in a proper way. To briefly show the decomposition for these two subsystems, the following diagram is provided:

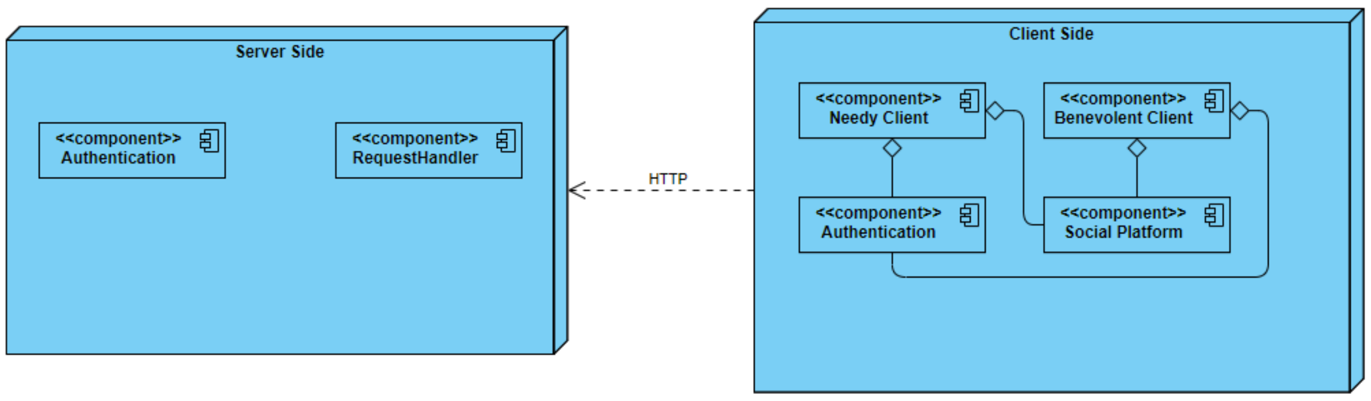


Figure 1: Subsystem Decomposition of HandsGiving

3.1.1 Client-Side

The client side of the application is composed of four components which are needy client, benevolent client, social platform and authentication. The main reason for dividing the client side into four components is to separate fundamental parts from each other. The needy and benevolent components are the fundamental components of the client side. They are separated from each other since there are two user types and most of their functionalities are different from each other. There are two basic categories of the application which are common for both of them; that's why we made them as a component which are authentication and social platform. Each one of the components contains the User Interface classes which are related with the function classes.

Two Tier Architecture

Each one of the client side components uses two tier architecture. The first layer is the service layer. This layer gets inputs from the other layer and creates requests to deliver the server side. When the response is received from the server, the service makes alterations on the other layer based on the responses. This layer provides the communication between the server side and the other layer. The other layer is the user interface. It is responsible to display the data to the user and receive inputs.

Needy client component contains the classes which are related with the needy users like profile, homepage and help request which are the fundamental classes of the component.

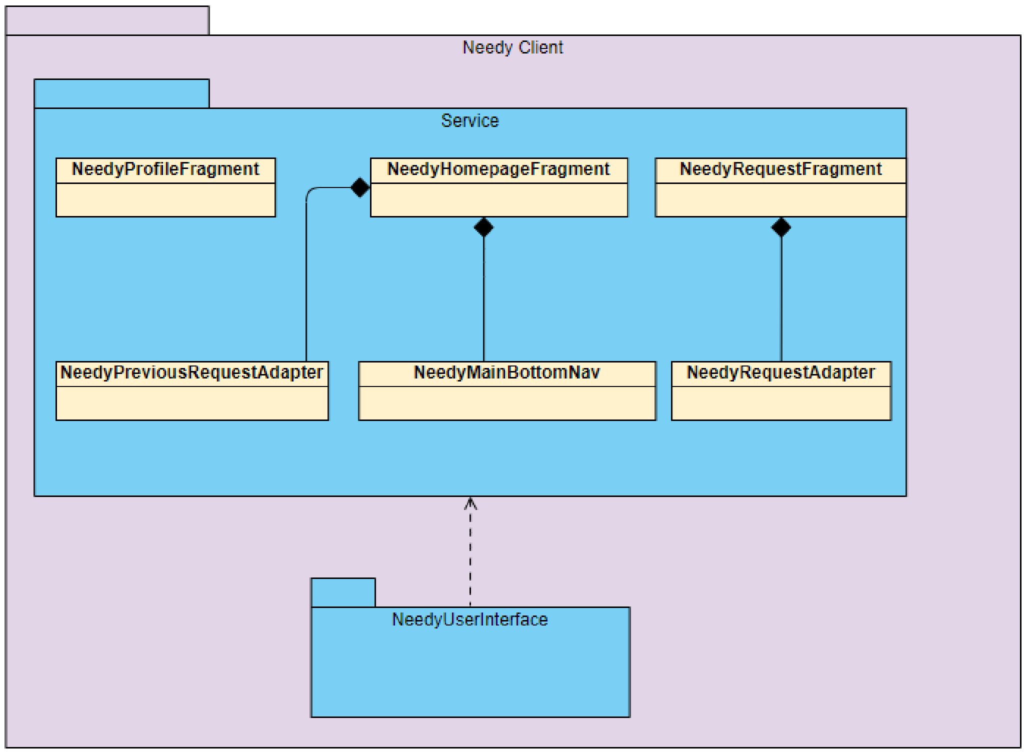


Figure 2: Needy Client Component decomposition

Benevolent client component contains the classes which are related with the benevolent users like profile, homepage, and assigned requests which are the fundamental classes of the component.

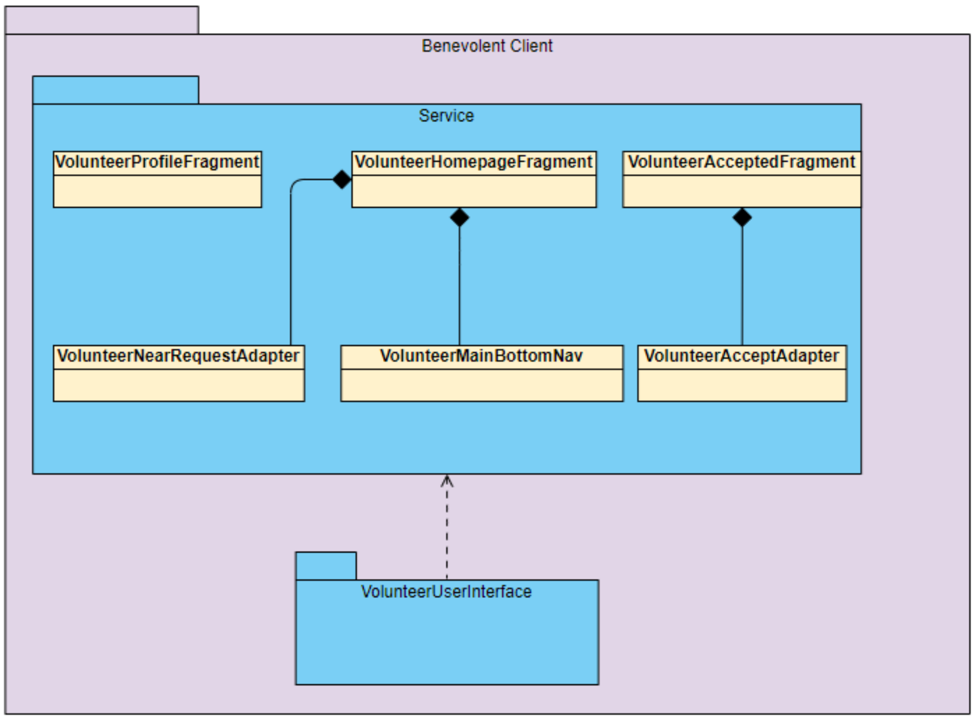


Figure 3: Benevolent Client Component decomposition

The authentication component is composed of the classes which are responsible for registration, sign in and forgot password.

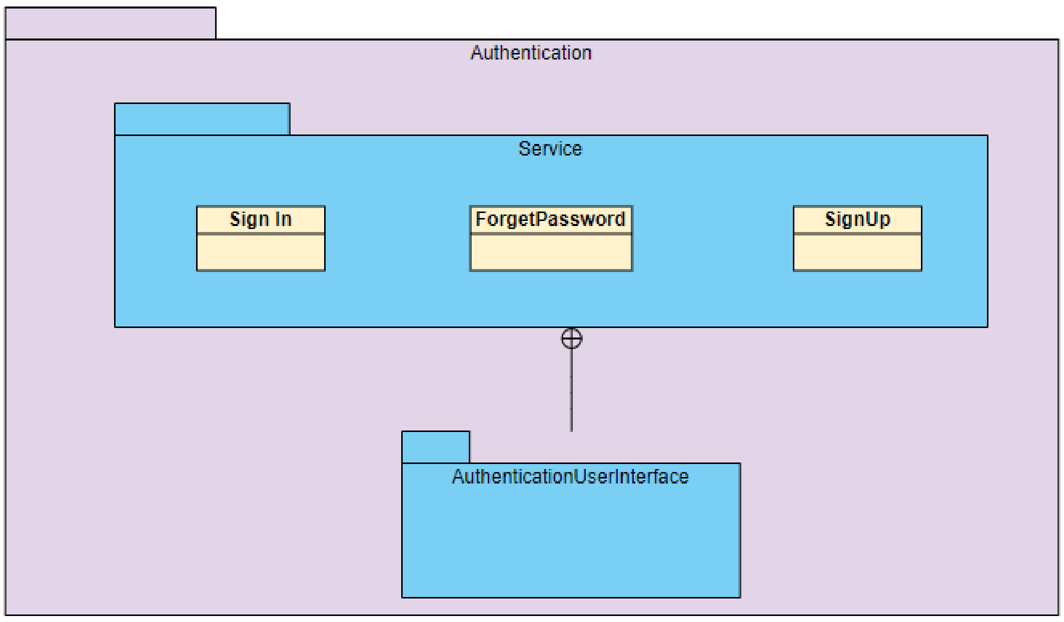


Figure 4: Authentication Component decomposition

The social platform is the part of the client-side which contains the classes related to chat and video chat.

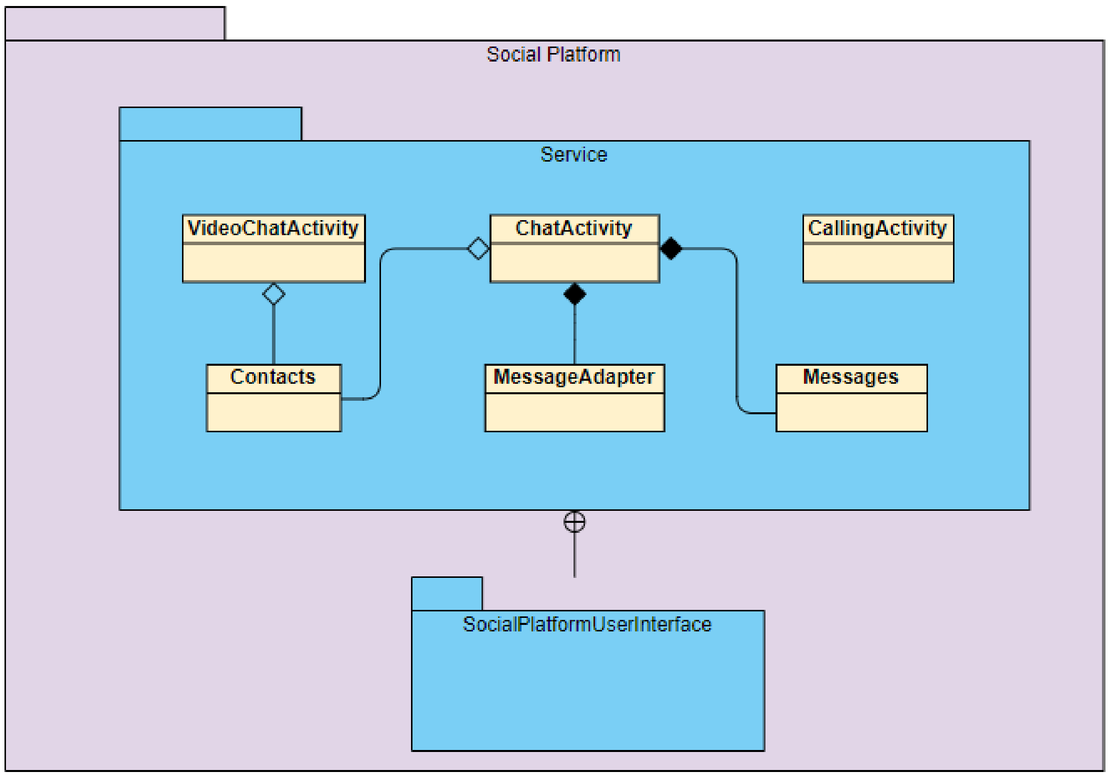


Figure 5: Social Platform Component decomposition

3.1.2 Server-Side

The server side of the application is composed of request based functions by using Node.js which is an open source, cross-platform runtime environment for developing server-side and networking applications [4]. We prefered to use Node.js because Node.js based server never waits for an API to return data and its library is very fast in code execution. These functions provide the data flow between database and client side with HTTP protocols.

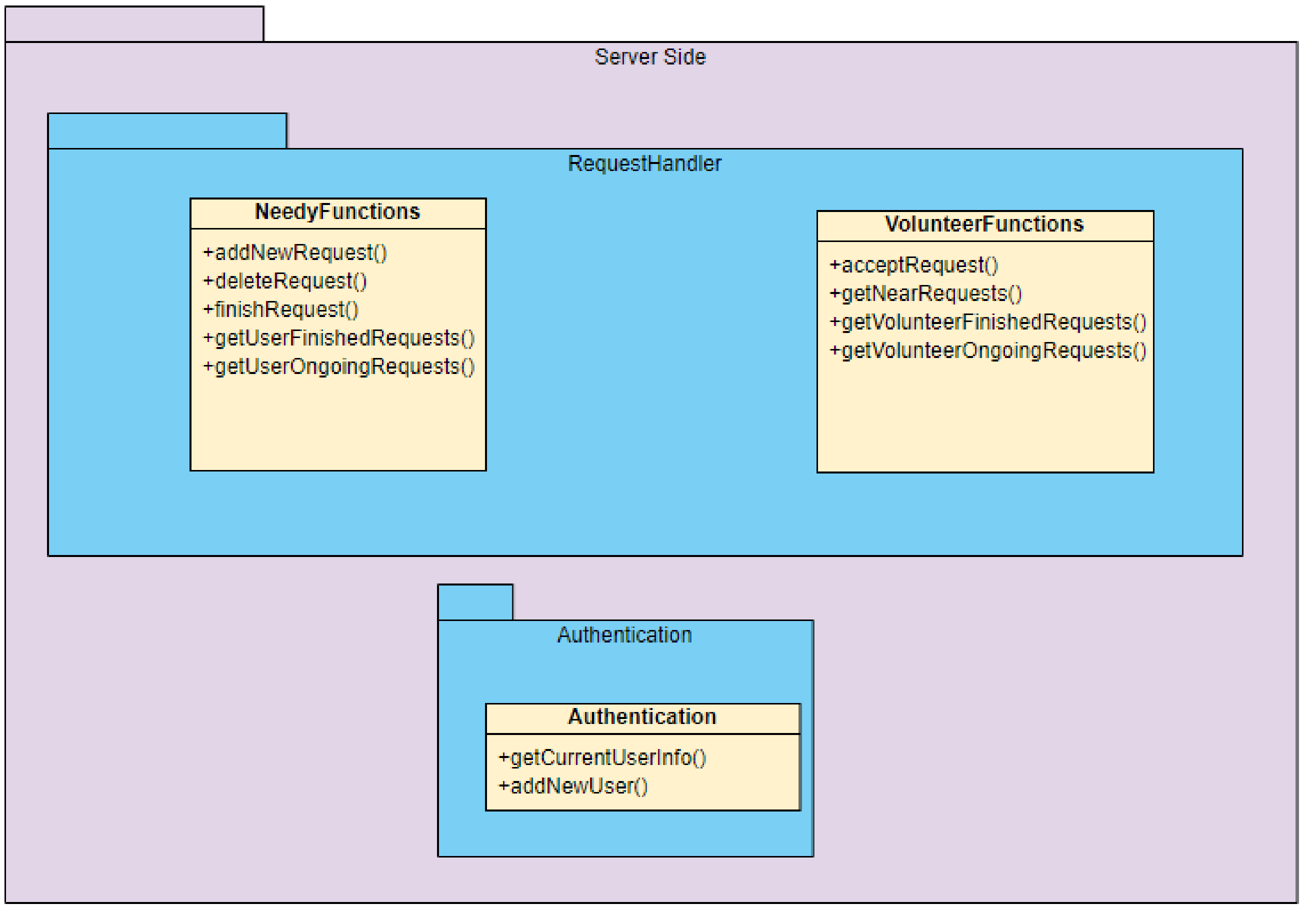


Figure 6:Server side decomposition

3.2 Firebase

To store the information of users and the application, we prefered to use Firebase in the HandsGiving project because it manages all data real-time in the database. So, the exchange of data to and from the database is easy and quick. As mentioned before, HandsGiving provides a video call feature. In order to provide a good experience to users, data flow should be quick and constant therefore we prefered Firebase which allows syncing the real-time data across all the devices- Android, iOS, and the web without refreshing the screen [5]. Firebase is compatible with all operating systems and this feature will help us to create the application for other platforms and reach more people.

3.2.1 Firestore Database

For the database of the project we stored the users' and help requests's data in the NoSQL Firestore Database. This database has two tables which are "Users" and "Request". The "Users" table has "Email, "Location", "Name", "Surname", and "UserType" fields and the "Request" table has "Accepted", "Description", "Finished", "Location", "LocationDesc", "RequestID", "RequestType", "userEmail", and "volunteerEmail" fields.

3.2.2 Google Cloud Storage

To store files in the server, Google Cloud Storage is used. Volunteer's and Needy's profile photos are uploaded in the Images folder with their id as the file name. For downloading and showing the images in the app we used Firebase's Storage Reference tool.

4. Development/Implementation Details

Authentication

While implementing authentication, Firebase is used for storing information of the user and retrieving data of the user if it exists on the database to compare given input. For the registration process users provide user type, name, surname, mail address, location, and password information. Location is taken by the LocationProvider class. After a new user providing information, his or her data is compared by the current users of the system. If there is a user with the same data as a mail address. The account will not be saved into the database and the user gets a notification. If there is no user with given data, a new user is created in the database and this data is saved in this user block. After the collection creation is completed, a verification mail is sent to the given mail address. If a new user does not confirm his or her mail address in a time, the account of the user will be deleted. With the mail verification, the registration period is completed.

For the sign-in process, users provide email and password to the system. After the system gets inputs, it searches the table called “User” to find a registered user in the database. If a user collection is found in this search period, the user will enter the application. Otherwise, a notification message is displayed to tell why the user cannot sign in.

Forgot Password

For the forgot password process, users provide a mail address to the application. After the input is received, the server searches the database to find a user with a given email. If there is a user with the given email, the system automatically sends an email to the user by using an API provided by Firebase. When a user clicks the URL in the sent email, a page will be displayed with an empty box. The user needs to enter the new password into the provided box. The new password is taken from the box by the system and delivered to the database. In the database, the old password is deleted and this input is placed in the password of the user with the given email.

Graphical User Interface

Graphical user interface includes activities, fragments and adapters. Fragments were used for a better performance and more convenient visuals for the android application [6]. Views such as google maps also implemented via fragments. Activities are required for android user interfaces. The lifecycle of the activity class was taken into consideration for any possible bugs on the usage of the application. Fragments, activities and adapters all use xml files as references to their user interfaces, leading to a more organized implementation. Adapters were used to adapt custom views to list views which were used in fragments for the purpose of viewing requests.

Both login and signup screens are activities. Upon signing in, users are directed to a main activity, which includes three main fragments (social, homepage and profile fragments). Each fragment holds various views such as buttons and textviews with custom backgrounds which are created uniquely to the application. The main theme color is green, since green often refers to positiveness and health. Fragments also use listviews which are customized with the help of adapter classes.

Google Cloud Functions (Server)

For the server side of the project we decided to use Google’s serverless Cloud Functions technology. The functions in Cloud Functions get HTTPS onCall requests from the client side and perform queries to communicate with the Firestore Database. These queries send back True or False responses to the client side if the operation is one of the Update, Delete, or Add operation and if the operation is a Get operation it sends a JSON object which includes the requested data from the Firestore Database. We send the needed data for example, user email, to perform the queries inside the Data JSON object from the client side. We used the user’s latitude, longitude data and compared them with the requests latitude, longitude to perform the “Near Requests” operation. We used boolean variables “Accepted” and “Finished” in queries to get the according requests for each fragment of the project.

LocationProvider

To get the current location of the user, LocationProvider implements the LocationListener interface which contains four functions which are onLocationChanged, onStatusChanged, onProviderEnabled, onProviderDisabled. In this class, only the onLocationChanged function is implemented.

When a user clicks on the current location button, the system tries to access location services. In order to access services on the device, the system asks for user permission for access. If the user gives permission, the system is connected with the location services and gets the current location of the user. The location is delivered to onLocationChanged function. In this function, the location of the user is added into the database as latitude and longitude and the location can be converted to the address to display the user.

Video Chat

Realtime Database Relations

While implementing the video chat feature, Firebase Realtime Database is used for storing the information about all the users and video chat sessions. All users are stored in a table called “Users” and all information about friendships is stored in a table called “Contacts”. When people call each other, information about the caller and receiver is added to the relevant places. A sub child named "calling" is added to the child that corresponds to the caller in the “Users” table and the user ID of the person that is being called is written in that sub child. Similarly, a sub child named “ringing” is added to the child that corresponds to the receiver in the “Users” table and the user ID of the person that is calling is written in that sub child. Based on the information, a video chat session is set up between two users. After the video chat between these users is over, the entire information saved in the database is deleted so that these people can make calls again. If the call is rejected or the caller user decides to stop calling, all stored information is deleted. In this way, users become eligible for making calls continuously.

Video Chat API

To provide a video chat feature to the users, Vonage Tokbox Video API is used. The Vonage Tokbox Video API platform makes it easy to embed real-time, high-quality interactive video, messaging, screen-sharing, and more into web and mobile apps. The platform includes client libraries for web, iOS, Android, Windows, and Linux, as well as server-side SDKs and a REST API. Vonage Video API uses WebRTC for audio-video communications. With the help of this API, the application offers an uninterrupted and high-quality video chat feature. The API consists of three main components and basically works in the following way.

The first component, a client (users), gets the session ID and token from the application server and connects to the session using the token. The caller user publishes audio-video streams and the receiver user subscribes to audio-video streams.

App server creates sessions in the Vonage Video API cloud. It generates tokens for clients. It also sends session IDs and tokens to clients.

The final component of the API is the session. A session is the "Chat room" in the Vonage Video API cloud. It connects clients to one another. It also sends events to clients [7].

Text Messaging

While implementing the text messaging feature, Firebase Realtime Database is used for storing the information about all the users and text messages. Information about sender, receiver, and the messages between them are stored in a table called “Messages”. The ID of the current user in the application is obtained from the Firebase Realtime Database and the messages sent by this user are displayed on the right side, while the messages sent to him/her are displayed on the left side.

5. Testing Details

A significant amount of time spent for testing during the implementation process since testing is crucial to ensure that an application runs as intended. For this purpose, we integrated certain testing patterns into the development of HandsGiving. These patterns are the followings:

5.1 Continuous Integration

In software engineering, the term Continuous Integration refers to the practice of merging all developers' working copies to shared mainline several times a day [8]. This testing pattern is used during the implementation process of the application. The most crucial key for successful Continuous Integration is to make sure that the project copy in the shared mainline is error-free. Github is used to keep the shared mainline. We also benefit from SourceTree to commit, push, pull, merge the code and create new branches in an easy way. It also helps us to see the implementation history of the code. Each group member can access the new code written by another group member from Github. To keep the shared mainline error-free each group member creates their own branch and pushes their new code to this branch after being sure that the integration is successful and the main branch is merged. We wrote test-code regularly and aimed at a certain line coverage, which means that a certain amount of the code lines are executed by our test code. Consequently, we ensured a significant amount of the code lines are running without any failed assertions.

5.2 Code Review

Coding conventions; such as variable and function naming, scoping, and the use of certain design patterns are used in implementation since all of the group members worked in different places and we can see the new code written by another group member when the member pushes the code to Github. To understand the code and significantly speed up the development cycles, the conventions were useful. The team members can review each other's code easily. With each push to Github, each team member firstly reviews the code. If there is a mistake or a way to develop this new code, suggestions are made to the person who wrote the code part. Code review pattern enables us to develop our application and find mistakes and bugs in the code easily.

5.3 Testing the UI

Testing the User Interface of the application was one of our primary concerns since our application has to have a simple and good user interface because of our target group. The purpose of the application is to help needy people who may not be familiar with complex applications and interfaces; that's why as mentioned before the user interface is crucial for HandsGiving. At each stage of the implementation, the user interface was tested by several different emulators on the Android Studio. We did not rely on one emulator since the view of the application pages may be different on different emulators. To get more user feedback, we get help from our families and friends. They used the application and after they tested the application we asked about their experience and their suggestions. According to their bug reports and feedback, we tried to fix the bugs and develop our application. This significantly helped us to detect and solve UI bugs fast, and design a better interface.

5.4 Server Performance and Response

One of the main features of HandsGiving is the video chat feature; it requires constant data flow from one user to another user and this data flow needs to be in a time that cannot be realized by a human. To see the performance and the possible bugs of the feature we prepared some test cases. According to the results of these test cases, bugs are figured out and response time is improved.

6. Maintenance Plan and Details

Maintenance of HandsGiving is compulsory since new ideas and enhancements will reach our feedback channels, and also bugs will be reported by our users. Also, the code base needs to be adjusted to keep up with the upcoming versions of the plug-ins. The patterns which are listed below will be our maintenance plan.

6.1 Server Maintenance and Optimization

On the social platform of HandsGiving, a video chat feature is available for users to meet with new people and call their friends. This feature requires constant data flow from one user to another user that's why maintenance of a well API is important to provide a good experience to the users. With optimization, we can even make it better. Our primary goal to add a social platform into the application is to decrease the feeling of loneliness of the users. Without maintaining and optimizing the platform they may not be satisfied. We need to check for any updates that can be done on the feature and user feedback to maintain and optimize the feature.

6.2 Problem Handling

The feedback coming from the users is significant for HandsGiving since with this feedback, problems can be detected easily and the support team can fix the problems early. The support team will take the comments of users from the Google Play Store page of the application and the source code of the application will be open. When we open our GitHub repository to the public,our team will be constantly reviewing the issues posted by the GitHub community and work with them to come up with fixes.

6.3 Google Play Store and other App Stores

The HandsGiving is implemented on Android Studio; that's why, HandsGiving will be available on Google Play Store after we are done. One of our goals is to release the application for other App stores like Apple's to reach other possible users and help more people. To be able to upload the app, we will need to write user agreements and send the app to the authorities for review.

6.4 Releasing New Versions

The HandsGiving application will require new versions after the app becomes available for the first time in the App Stores since the feedback and the suggestions will be made by the users; and based one these comments new versions of the app will be released. These new versions will involve changes to the UI and video chat improvements to provide better experiences to the users .

7. Other Project Elements

7.1 Consideration of Various Factors in Engineering Design

During the research process, there are several factors that can have an impact on HandsGiving's results and development process. These factors will be explained in the sections below.

7.1.1 Public Health

The application idea was created in the coronavirus epidemic. Because of this epidemic, many people have to stay home and some of them stay lonely. Because of that people are affected psychologically. The social platform is added into the application in order to help these people to socialize. Video chat features will be improved in HandsGiving to make it easier for the elderly to socialize. In this way, it will be aimed to keep the people psychologically healthy. Besides, people who have difficulties in meeting their food and other basic needs will be helped to be healthier.

7.1.2 Data Privacy

In the HandsGiving application, some private information is required. For instance, in the registration process, people need to enter their mail address, a password, and their current location. Mail addresses will be available on their profile; however, their password will not be shared with any other people since each person's account has to be private for them. Location information will also be kept confidential. It is just used when a needy user creates a help request. The address of benevolent users will not be shared. To get the address of a user, the application will ask the user's permission to access the location of the user. Private information will not be shared by any third-party applications.

7.1.3 Public Welfare

Our application will be free to download and contain no in-app purchases, it has no effect on public welfare. Anyone who has a device with an Android operating system can download the application for free and use each function of the application for free. It will not affect the budget.

7.1.4 Global Factors

Since HandsGiving will be an application created specifically for our country, it will not have an impact on global factors.

7.1.5 Cultural Factors

In the first version, this app will be available for Turkey which contains people from different cultures. During the help processes and video calls, people may meet with someone from another culture. With these interactions, they might have a chance to learn about other cultures.

7.1.6 Social Factors

The main purpose of HandsGiving is to create a platform that brings the needy and benevolent people to a point. With this goal, it will create social awareness by making it easier to reach people who need help. A wider audience will be reached to meet people's needs faster. As a result of help processes in the application, people will be more aware of other people.

7.1.7 Economic Factors

HandsGiving application benefits from the Google Firebase platform to store and retrieve data. After the total number of users exceeds 1000, we need to pay a price until this time the only factor that will affect us economically will be the release price of the application on the Google Play Store.

	Effect Level	Effect
Public Health	9	Mental and physical health
Data Privacy	9	Keeping private information confidential
Public Welfare	0	None
Global Factors	0	None
Cultural Factors	5	Bringing different cultures together while helping each other
Social Factors	8	Raising social awareness about people in need of assistance
Economic Factors	4	Requiring budget for release and after total users exceeding 1000

Table 1: Factors that can affect analysis and design.

7.2. Ethics and Professional Responsibilities

From the perspective of ethics, the developed application has different aspects that are needed to be covered. The application will have some confidential information about users, but this information will not be shared with anyone without the user's permission. In addition, the locations of the users will not be shared with anyone too. The application will ask users to obtain their permission before sharing any information with another user. The application will provide a terms and conditions agreement that the user needs to read and accept while registering to HandsGiving. Apart from these, to prevent any copyright issues, our team will prefer to use open-source products in order to develop the application.

In terms of professionalism, the critical factor is that the application uses work from literature. Research about the idea which is the base of our application is conducted by different researchers. Plagiarism is an issue that we want to keep away from our project. Not to face this problem and in order to show that there is no intention of making use of their work and making profit without their consent, this report cites the work that is benefited and gives necessary credit. Also, the HandsGiving application is not designed to make a financial profit.

Concluding, by giving the notice that the predictions given as insights may have certain kinds of errors considering certain trade-offs and the current technology and citing the work benefitted while developing the application, our project fulfills its responsibilities both from the ethics and professionalism perspective.

7.3 Judgements and Impacts to Various Contexts

We decided to use Turkish as the language of recognition since we want to test our program in a country before making it global and chose Turkey. This should not have a big impact on a global scale since the application is not appropriate for people who cannot speak Turkish; however, it still has an impact since Turkish is the fifth most used language in the world [5]. This is described in the table below.

Judgment Description		The decision to use Turkish
	Impact Level	Impact Description
Impact in Global Context	6	Turkish is the 5th most used language in the world.
Impact in Economic Context	0	None
Impact in Environmental Context	0	None
Impact in Societal Context	0	None

Table 2: Judgment table for language of the application

The HandsGiving will be a free application and there will be no in-app purchase. Each one of the users can benefit from each feature of the app. This situation has an economical effect.

Judgment Description		The decision to make the app free
	Impact Level	Impact Description
Impact in Global Context	0	None
Impact in Economic Context	8	Free application and no app-in purchase
Impact in Environmental Context	0	None
Impact in Societal Context	0	None

Table 3: Judgment table for making free application

The main purpose of the HandsGiving is to bring needy and benevolent people together. Volunteers can help others financially by buying some food or necessary equipment for them. When these processes are thought on an economic scale, they have a positive impact on needy people. Besides, in the societal context, with the help processes, people can meet new people. Help is actually something positively correlated with society since increasing helps bring society together.

Judgment Description		Creating a help application
	Impact Level	Impact Description
Impact in Global Context	0	None
Impact in Economic Context	9	Helps the budget of needy people
Impact in Environmental Context	0	None
Impact in Societal Context	10	Bring the society together

Table 4: Judgment table for creating help application

At the beginning, HandsGiving was planned to be a help application and there would be no social platform; however, because of pandemic, some people have to stay at home alone and we want to add a social platform to provide an area which they can socialize on. In the societal context. it gives an opportunity to people to meet with new people and brings the society together.

Judgement Description		Add social platform into th application
	Impact Level	Impact Description
Impact in Global Context	0	None
Impact in Economic Context	0	None
Impact in Environmental Context	0	None
Impact in Societal Context	9	Bring the society together

Table 5: Judgment table for adding social platform into the application

7.4 Teamwork Details

7.4.1 Contributing and functioning effectively on the team

Because a project had a specific end, it was important that every person contributing to the effort knows the desired result. By understanding the desired outcome, a project member can make better individual decisions and reduce confusion and rework. Moreover, each person was an essential piece in the overall project structure. Everyone understood their role and the roles of others. Each group member communicated with each other by asking questions and giving updates about the parts that they are responsible for during the project process frequently. Besides, every group member adapted themselves to the changes that may possibly occur in different phases of the project. Therefore, they could maximize their contribution to the project. Apart from these, tasks in the project were divided in a way that everyone can function effectively. This was achieved by dividing the tasks according to the areas in which project members' knowledge is strong.

7.4.2 Helping creating a collaborative and inclusive environment

In order to have a collaborative and inclusive environment in the project, all team members were informed of the strengths and weaknesses of their members and acted according to them. They tried to compensate for the weaknesses of their team members so that the flow of the project was not disturbed. Each group member had a chance to be open-minded about their ideas and their behaviors to take a unique approach to each situation with a different perspective. That affected the overall performance of the team positively.

7.4.3 Taking lead role and sharing leadership on the team

To encourage everyone to participate in leadership, we broke the project down into work packages and make everyone the leader of one package. Each member of the team was assigned to all work packages so that they have the same workload. Thus, every project member was aware of all the processes of the project. All team members were in contact with each other at every moment of the project to complete tasks simultaneously. There is a dedicated GitHub repository to maintain the project's code base and progress. All team members have access to the archive. This archive will be shared with the project advisor and jury members if they wish.

7.4.4 Meeting Objectives

At the analysis stage of our project, we have constructed a project plan for the design and the development of the project HandsGiving. This section is dedicated to evaluating our initial project plan and will consider to what extent this plan is followed.

As mentioned in the implementation details, even though attempted some of the functionalities that we have mentioned in our analysis were not included in the final version of the project. To be specific, we mentioned in our analysis reports that there would be a feedback mechanism for users to evaluate the other side of the help process after finishing the process and feedback evaluation system which would be able to block the accounts of users based on the evaluation of the feedback for finished help processes and an average rating of him or her. These tasks were the ones that we planned to do after other functionalities of the application is completed. As we observed during our development phase, the time slots that we reserved for particular units were optimistic assumptions and as a result that we could not add these mechanisms to the application.

Considering the challenges that we faced as a team during the development of our application showed that the time estimates that we made were not realistic. As a team, we faced difficulties since none of us had enough experience for such a big project for a long period; that’s why our estimations for each part of the project were less than it requires. As a result of that, we had to speed up some of the project parts and could not add some functionalities we mentioned before.

Our team is also affected by the coronavirus pandemic. Some of our group members suffered from this disease and could not contribute to the project for a while because of it. This situation resulted in a decrease in the total productivity of the project team. Also, the other effect of the pandemic was that we had not had a chance to apply pair programming principles. If we could do it, we may review each other's implementation and find the bugs in the application easily. In order to minimize the effect of the pandemic, we conducted meetings each week through zoom and discord platforms.

Precisely, the objectives that we met can be listed as follows:

- Implementing a social platform that included video call and chat functionalities
- Implementing a system to get user’s current location automatically
- Implementing a help request system to deliver the needs of needy people to benevolent people.
- Implementing a simple and well User Interface.

In conclusion, as a team, we consider our development life-cycle as a success in the sense that the fundamental functionalities of the application have been implemented successfully. In terms of the project plan, the team had certain defects because of the poor estimations for some of the project parts and coronavirus pandemic. With this project, each of us gains experience for the progress of such a project. As a result of that we believe that if we will work on such a project, we can make more precise estimations and follow the project plan more precisely.

Teamwork

During the analysis and design stages, we met almost every week to discuss the details of the design and functionalities of the application, and which technologies we could use in the implementation stage to make our design a reality. Also, we told what we have done so far about the project personally to each other. For the reports, we divided the parts of the reports. Each group member made at least a couple of parts for each report. We used Google Documents to review each other's part and correct if there are mistakes.

In the implementation stage, we divided the parts of the project among the group members based on what they wanted to do on the project. Because of the coronavirus pandemic, we could not have a chance to work together. Each group member used sourcetree and Github to share their works to others. After each commit we merged the new code and the old version. With these tools, we could easily follow the progress of the project. Also, we made meetings on discord and zoom to discuss what we have done and what we should do. In these meetings, each one of us shared ideas with others and got comments about the parts which were implemented by them. As a summary, Throughout the semester we tried our best to hold weekly or bi-weekly meetings with all members who could attend at that time and create a beneficial application.

Peer Contribution

Fırat Yönak

- implemented authentication.
- implemented the database relation to create new users and put new users data into the database.
- implemented LocationProvider class to get users' current location and description of address.
- implemented the ForgetPassword class.
- implemented User Interface of registration.

Mehmet Tolga Tomris

Tolga implemented the social platform of the application with Atakan.

- helped to implement the video chat feature of the application.
- helped to implement the User Interface of the social platform of the application.

- helped to implement the chat feature of the application.
- helped to implement add friends.
- helped to create a real-time database to maintain information about friendships, video chat, and chat.

Atakan Arslan

Atakan implemented the social platform of the application with Tolga.

- helped to implement the video chat feature of the application.
- helped to implement the User Interface of the social platform of the application.
- helped to implement the chat feature of the application.
- helped to implement add friends.
- helped to create a real-time database to maintain information about friendships, video chat, and chat.

Ege Hakan Karaağaç

- implemented server side queries.
- implemented codes to put data to the database.
- implemented server request at client side.
- implemented User Interface of the volunteer homepage.
- implemented the show Google Maps on the application to the users with Oğuzhan.

Oğuzhan Dere

Oğuzhan mostly worked on the User Interface of the application.

- implemented the User Interface of the profile page.
- implemented the User Interface of the needy homepage.
- implemented the User Interface of the pages related to a needy request.
- implemented the User Interface of the sign in page.
- implemented the show Google Maps on the application to the users with Ege Hakan.

7.5 New Knowledge Acquired and Learning Strategies Used

Before the senior project, all of us were not familiar with Android studio. We just used the CS102 course and it was 3 years ago and the applications we made on Android

Studio was so simple; that's why we learned a lot of new information before and during the implementation process about coding in Android Studio. In order to be familiar with Android Studio we watched many videos on both youtube and udemy. The server functions were implemented in JavaScript language which we only used in CS353 course. We learned how to code in javascript using node.js library from sources on the Internet. We stored the data of the application and users in Firebase which was not familiar for us before the project. While learning how to store data to firebase and retrieve data from firebase, the sources on the internet, youtube videos, and especially Google's own sources were beneficial for us. User Interface is another topic, we totally learned during the project since each of us knows how to create a user interface by drag and drop to the page in Android Studio; however, the results of this creation of user interface may be different for different devices and some parts of the interface may slide. To create a stable, simple, and elegant user interface for the HandsGiving, we had to learn how to do it. From udemy course and youtube videos, we learned. In the social platform of HandsGiving, users can call their friends or meet new people through video chat which we had not implemented before the coding process of HandsGiving. To figure out how to implement a video chat for the application, we got help from youtube.

We improved our knowledge of version control systems and Git with Github workflow through gaining more experience and working together and also using tutorials. The research was a big part of the project to learn new things. Other than that, most of the time we used hands-on experience to learn a new subject.

8. Conclusion and Future Work

In conclusion, we designed an Android-based mobile application to bring needy and benevolent people together. With this application, needy people can meet some of their needs which can be financial or physical like carrying some stuff. Furthermore, a social platform is included in the application to help people to socialize and meet new people. The social platform idea occurred because of the coronavirus pandemic.

For the future of our project, we can create the application for different operating systems. By doing this we can increase the total number of users and find a chance to help many more people. Language support also can be added to the application to find new users globally since now the application can be used only by people who can speak Turkish. We can develop a user type for charities to help people reach these foundations more easily. By doing that improvement, these foundations can reach more people. Also, they can find more volunteers from benevolent users of the application. Adding them into the application can make the help process more official and increase the number of volunteers who will be registered to the application.

9. User Manual

Sign-In

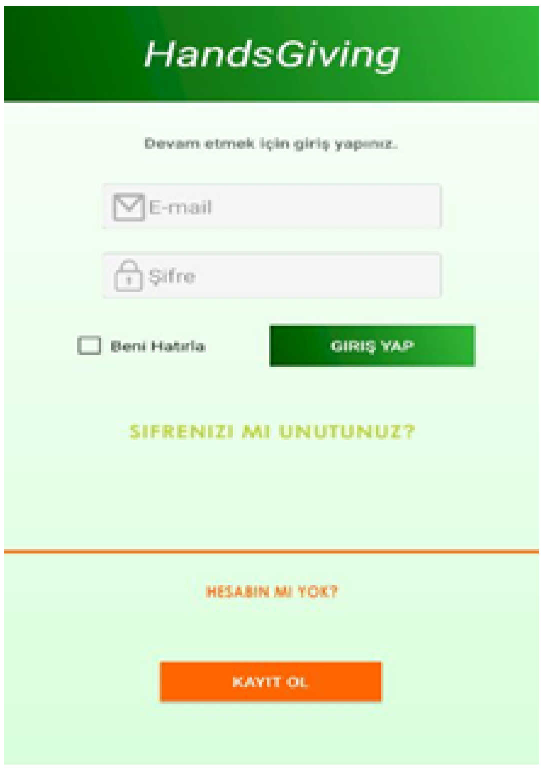


Figure 7: Sign-In Page

- The application starts with a sign-in page.
- Users can log in to their account by using this page.
- They need to write their e-mail and password in the corresponding fields and press the "GİRİŞ YAP" button.
- If they want their account information to be remembered, they need to select the "Beni Hatırla" box.
- If they forget their password, they can click on the "ŞIFRENIZI MI UNUTTUNUZ?" link and reset their password.
- If they do not have an account, they need to click on the "KAYIT OL" button.

Sign-Up

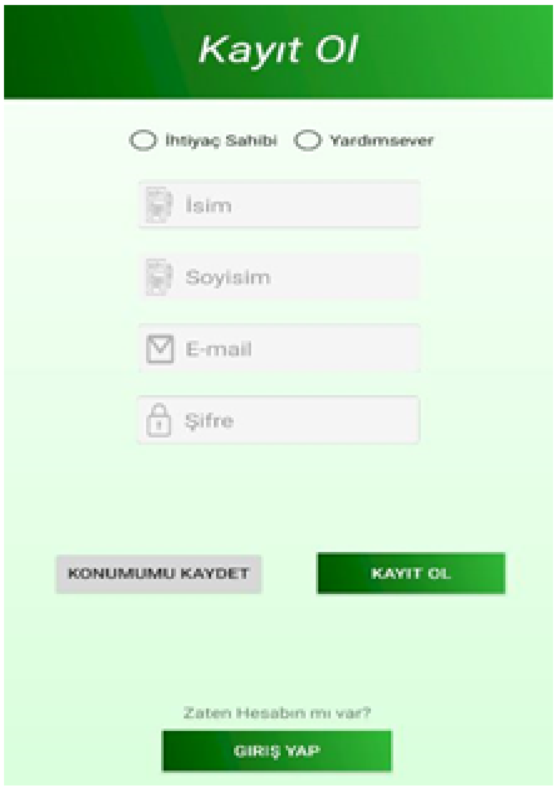


Figure 8: Sign-Up Page

- There are two types of membership in HandsGiving.
- Users can sign-up as a needy by choosing the “İhtiyaç Sahibi” box or they can sign up as a volunteer by choosing the “Yardımsever” box.
- They need to write their name, surname, e-mail, and password in the corresponding fields.
- They need to click on the “KONUMUMU KAYDET” button to share their location with the application.
- Finally, they need to click on the “KAYIT OL” button to create their account.

Needy User Homepage

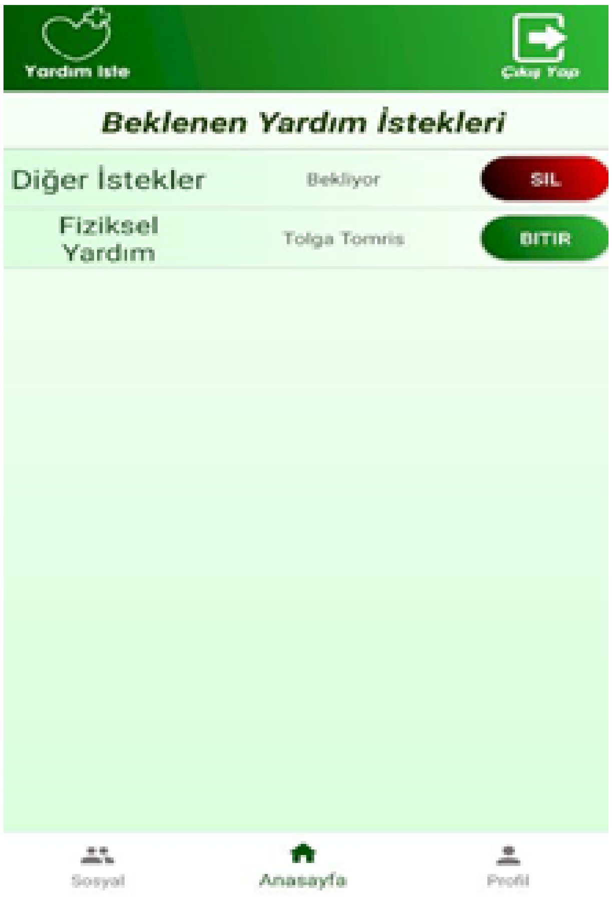


Figure 9: Needy User Homepage

- When needy people log in to their account, they will see the needy homepage.
- They can see previous help requests created by them.
- If the request is not accepted by any volunteer yet, they will see “Bekliyor” as a status of their request.
- If the request is accepted by a volunteer, they will see the name of that volunteer as a status of their request.
- If the status is “Bekliyor”, they can remove their request by clicking the “SİL” button.
- If the request is done by a volunteer, they can click on the “BİTİR” button to finish the request.
- They can create a new request by clicking the “Yardım İste” button.
- They can go to the social platform of the application by clicking the “Sosyal” button.
- They can log out from their accounts by clicking the “Çıkış Yap” button.

Needy User Profile Page

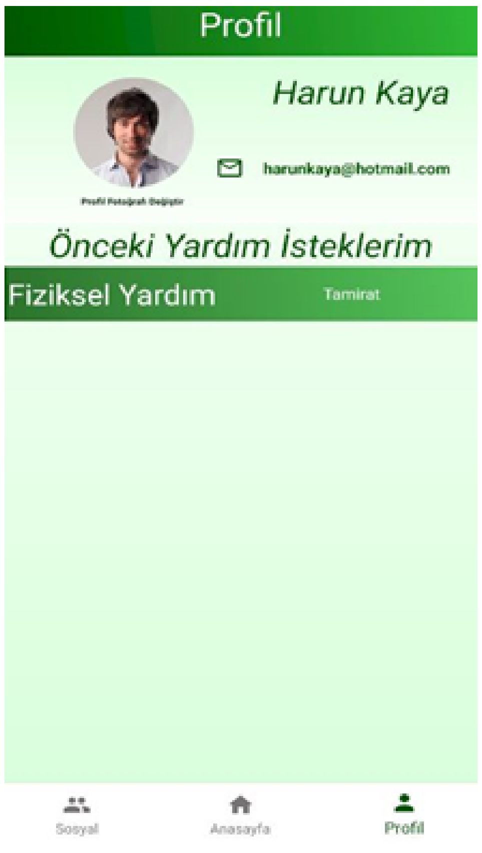


Figure 10: Needy User Profile Page

- On the needy user profile page, needy people are able to see their names and e-mail addresses.
- They can change their profile images by clicking the “Profil Fotoğrafı Değiştir” link.
- They can see the previous help requests created by them.
- They can see the type of their requests and explanation to their request.

Needy Create Help Request Page

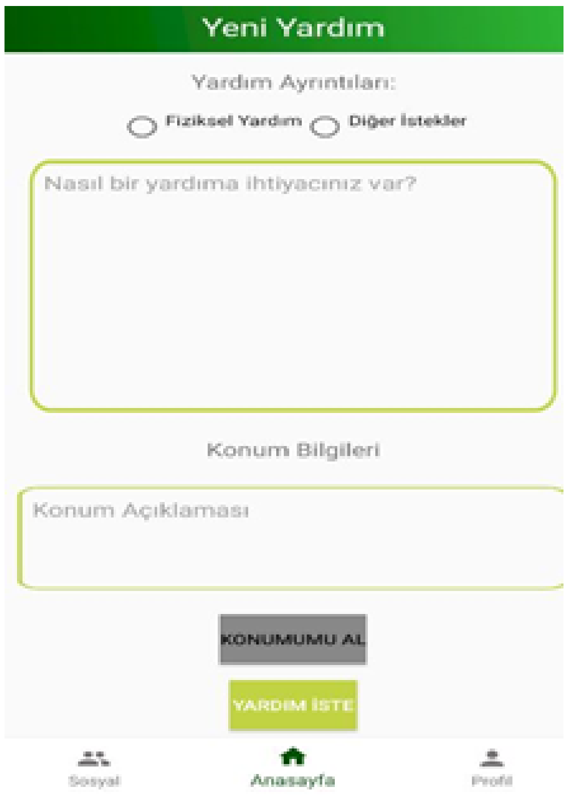


Figure 11: Needy Create Help Request Page

- On the needy create help request page, needy people are able to create new help requests according to their needs.
- They can choose their help details by clicking the "Fiziksel Yardım" or "Diğer İstekler" button.
- They can fill out the "Nasıl bir yardıma ihtiyacınız var?" form to explain their need in a more detailed way.
- They can fill out the "Konum Bilgileri" form to explain their addresses in a more detailed way.
- They need to click on the "KONUMUMU AL" button to share their location with the application.

Volunteer User Homepage

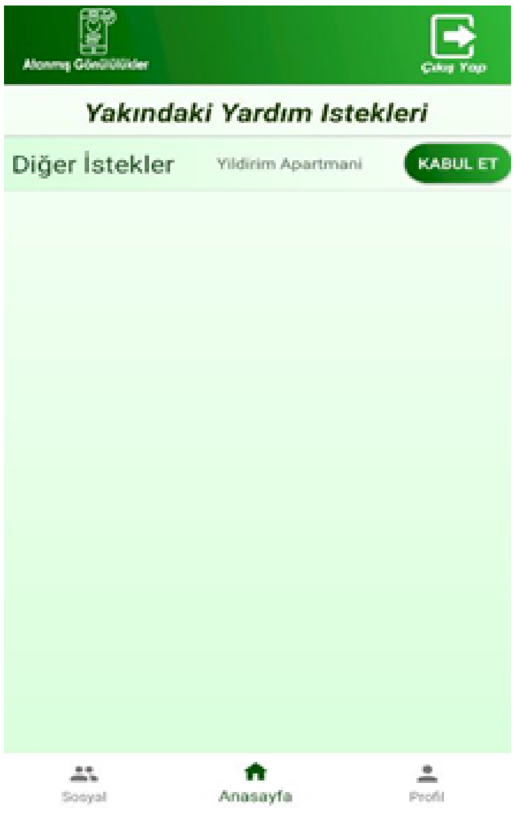


Figure 12: Volunteer User Homepage

- When volunteer people log in to their account, they will see the volunteer homepage.
- They can see help requests close to their location.
- They can accept any request by clicking the “KABUL ET” button.
- They can see the previous request accepted by them by clicking the “Atanmış Gönüllülükler” button.
- They can go to the social platform of the application by clicking the “Sosyal” button.
- They can log out from their accounts by clicking the “Çıkış Yap” button.

Volunteer User Profile Page

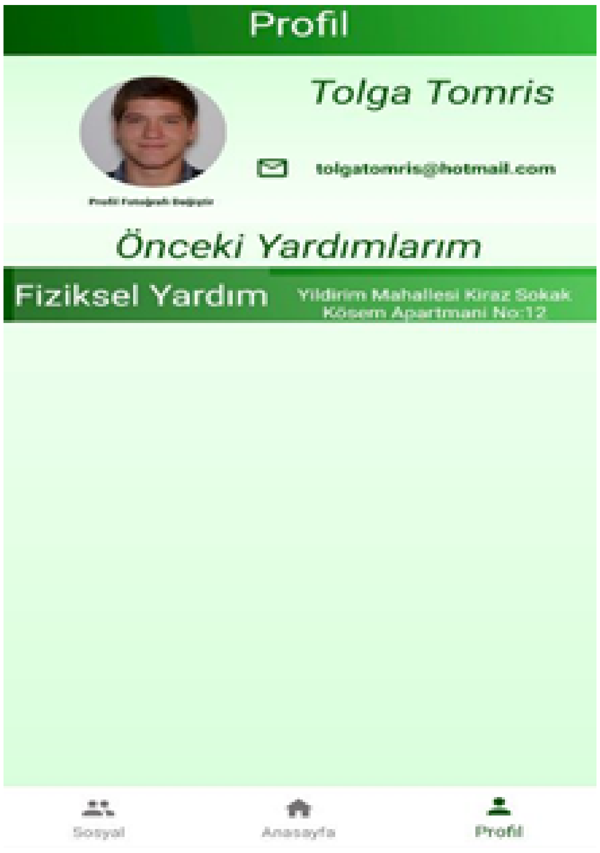
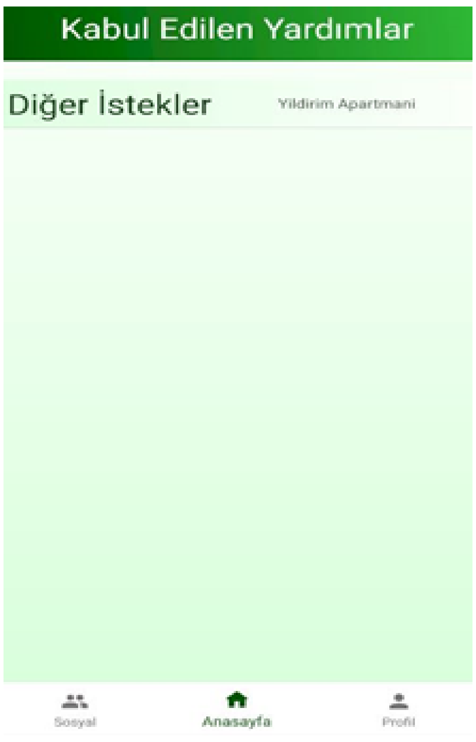


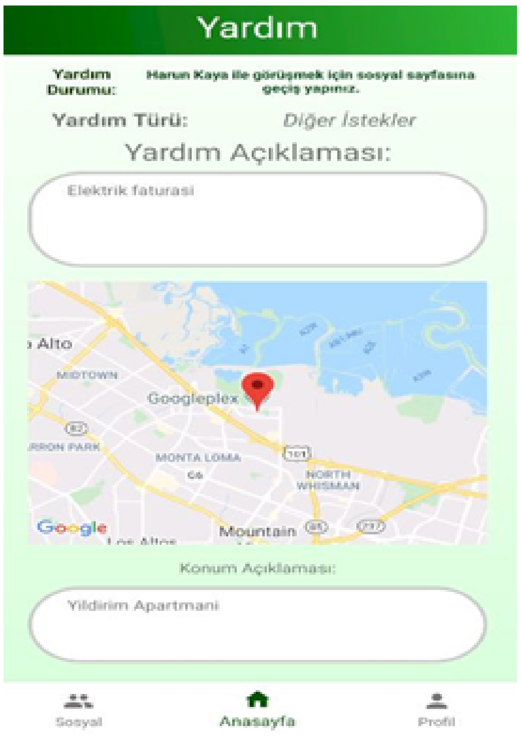
Figure 13: Volunteer User Profile Page

- On the volunteer user profile page, volunteers are able to see their names and email addresses.
- They can change their profile images by clicking the “Profil Fotoğrafı Değiştir” link.
- They can see the previous help requests that are completed by them.

Volunteer Assigned Help Request



Diğer İstekler Yildirim Apartmani



Yardım Durumu: Harun Kaya ile görüşmek için sosyal sayfasına geçiş yapınız.

Yardım Türü: Diğer İstekler

Yardım Açıklaması:

Elektrik faturasi

Konum Açıklaması:

Yildirim Apartmani

Figure 14: Volunteer Assigned Help Requests *Figure 15: Details of a Help Request*

- On this page, volunteers can see the help requests that are accepted by them.
- They can also see the location of the request.
- They can see the details of the request by clicking one of their assigned requests.
- Detailed information of the help request will be shown on a page as in Figure 15.

Contacts Page

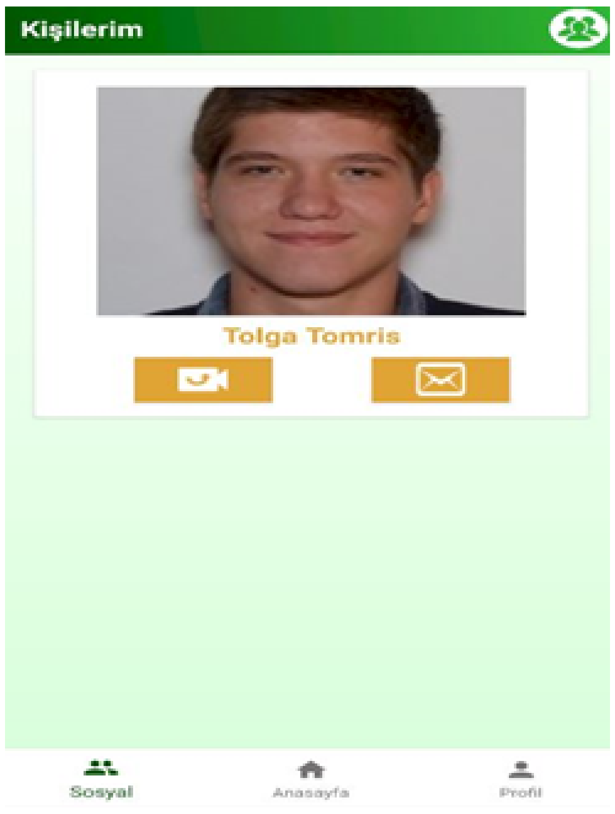


Figure 16: Contacts Page

- On the contacts page, the users are able to see their contacts.
- They can see the names and profile pictures of their contacts.
- They can create a video call with their contacts.
- They can send a text message to their contacts.
- Needy people can search for other users by clicking the button at the top right.
- Volunteers can see their friend requests by clicking the button at the top right.

Search for Other Users Page

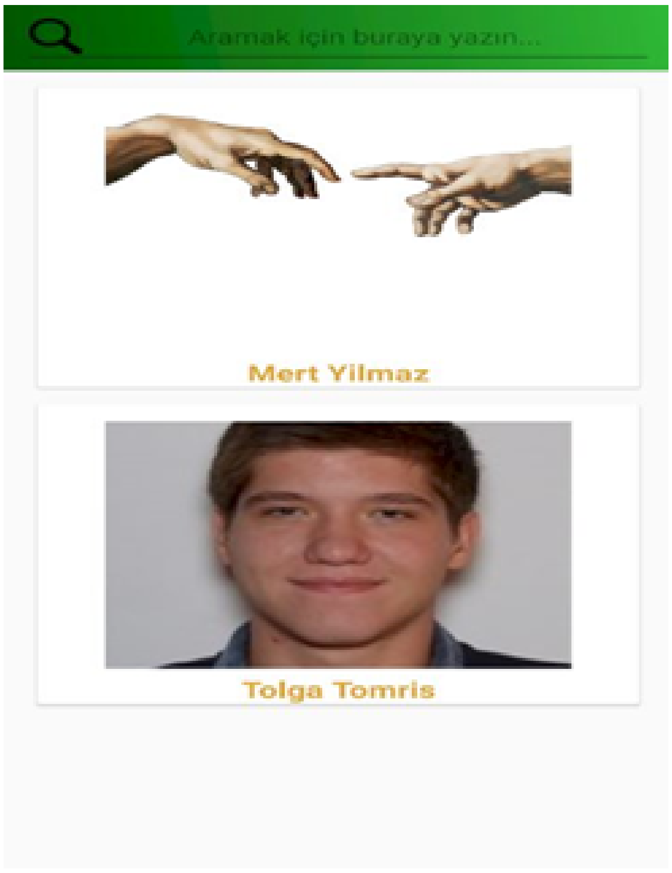


Figure 17: Search for People Page

- Only needy people can search for users and add them as a friend for their socializing needs.
- They can search for volunteers by typing a key word.
- They can click on any of the volunteer’s images and see their profile.
- They can send a friend request to them from that page.

Notifications Page

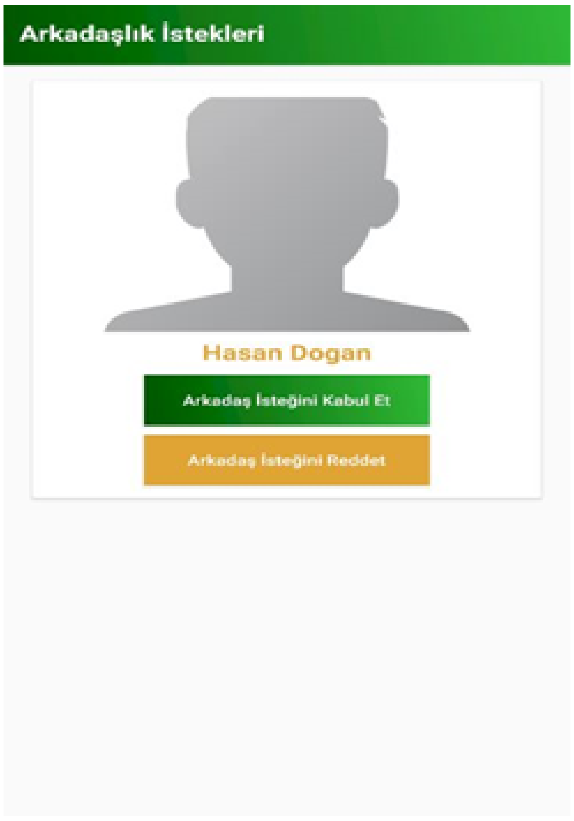
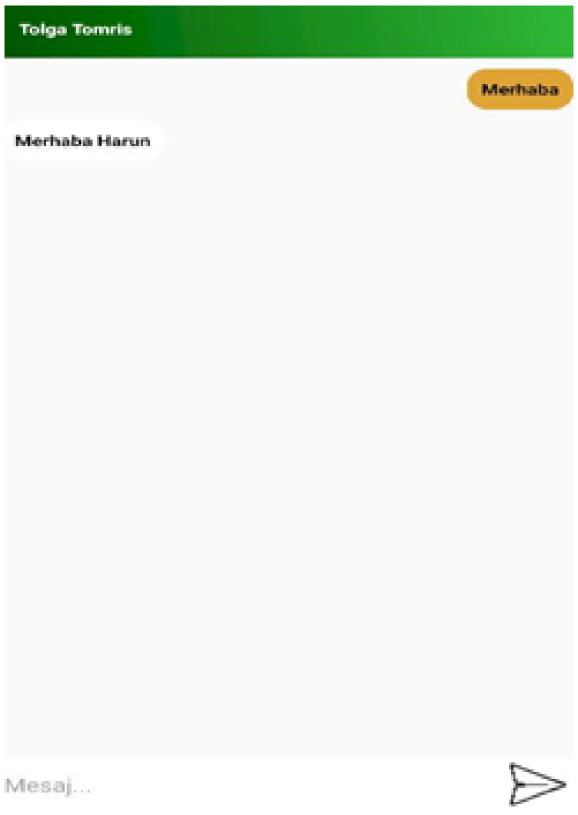


Figure 18: Notifications Page

- Volunteers can see the friend requests that are sent to them on the notifications page.
- They can accept the request by clicking the "Arkadaş İsteğini Kabul Et" button or decline it by clicking the "Arkadaş İsteğini Reddet" button.
- When a request is accepted each person will be shown on their contacts page.

Text Messaging Page


Tolga Tomris

Merhaba

Merhaba Harun

Mesaj...

Figure 19: Text Messaging Page

- Users are able to send text messages by using this page.
- They can socialize from this page or they can talk about the status of help requests.

10. References

[1] Li, Sijia, et al. “*The Impact of COVID-19 Epidemic Declaration on Psychological Consequences: A Study on Active Weibo Users.*” *MDPI*, Multidisciplinary Digital Publishing Institute, 19 Mar. 2020.[Online]. Available: www.mdpi.com/1660-4601/17/6/2032/htm. [Accessed: 24- Apr2021].

[2] Hürriyet.” *65 Yaş Üstü Ne Zaman Sokağa Çıkacak?”.*[Online]. Available: www.hurriyet.com.tr/galeri-65-yas-ustu-ne-zaman-sokaga-cikacak-41503410/2. [Accessed: 24- Apr2021].

[3] Guru99. “*Functional requirements vs non functional requirements: Key differences*”.[Online]. Available: https://www.guru99.com/functional-vs-non-functional-requirements.html. [Accessed: 24- Apr- 2021].

[4] tutorialspoint. “*What is Node.js?*” . [Online].Available: https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm . [Accessed: 26- Apr- 2021].

[5] TRISTATE TECHNOLOGY. “*Why Firebase is the Best Mobile Backend as a Service?*”. [Online].Available: https://www.tristatetechnology.com/blog/firebase-backend-mobile-app/#:~:text=Firebase%20manages%20all%20data%20real,.%2C%20you%20can%20use%20Firebase.[Accessed: 27- Apr- 2021].

[6] javaTpoint. “*Android Fragments*”. [Online].Available: https://www.javatpoint.com/android-fragments#:~:text=Android%20Fragment%20is%20the%20part,fragments%20are%20included%20in%20activity.[Accessed: 27- Apr- 2021].

[7] Company, Vonage. “Video API Basics.” *Video API Basics | Vonage Video API Developer*, tokbox.com/developer/guides/basics/.

[8] Wikipedia. “*Continuous Integration*”.[Online].Available: https://en.wikipedia.org/wiki/Continuous_integration [Accessed: 25- Apr- 2021].

[9] EDUMAG. “Dünyada En Çok Konuşulan 5 Dil”.[Online].Available:
https://edumag.net/galeri/dunyada-en-cok-konusulan-5-dil/ [Accessed: 26- Apr- 2021].