

T.C.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

**ACCOMMODATION FINDER MOBILE
APPLICATION FOR GTU STUDENTS**

AKİF KARTAL

**SUPERVISOR
PROF. DR. FATİH ERDOĞAN SEVİLGEN**

**GEBZE
2022**

**T.C.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT**

**ACCOMMODATION FINDER MOBILE
APPLICATION FOR GTU STUDENTS**

AKİF KARTAL

**SUPERVISOR
PROF. DR. FATİH ERDOĞAN SEVİLGEN**

**2022
GEBZE**



GRADUATION PROJECT
JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 20/01/2022 by the following jury.

JURY

Member

(Supervisor) : PROF. DR. FATİH ERDOĞAN SEVİLGEN

Member : PROF. DR. ERCHAN APTOULA

Member : RES. ASST. FERDA ABBASOĞLU

Member : RES. ASST. MEHMET BURAK KOCA

ABSTRACT

With this project, Gebze Technical University students will have the chance to find accommodation around the university much more easily. By using this application, Gebze Technical University students can see all private and public dormitory information around the university, such as quota status, pictures and comments of other students about these dormitories. In this way, advertising dormitories with the poster will be over. Besides this, students can see the information of other students such as age, gender, department who are looking for a roommate at the university and can communicate with them through the messaging and chat feature of the application. In this way, the period of searching for a roommate from the billboard will be over. Finally, by using sentiment analysis module of this project, students can easily see the positive and negative comments made to the dormitories. Also, by using this module, dormitory owners can easily analyze positive and negative comments and they will take the necessary actions.

Keywords: accommodation, comments, roommate, chat, sentiment analysis

ÖZET

Bu proje sayesinde, Gebze Teknik üniversitesi öğrencileri üniversite çevresinde çok daha kolay bir şekilde kalacak yer bulma şansına sahip olacaklar. Bu uygulamayı kullanarak Gebze Teknik üniversitesi öğrencileri, üniversitesinin çevresindeki tüm özel ve devlet yurt bilgilerini, kontenjan durumlarını, resimleri ve diğer öğrencilerin bu yurtlar hakkındaki yorumlarını görebilirler. Bu sayede afiş ile yurt tanıtım devri kapanmış olacak. Bunun yanı sıra öğrenciler üniversitede ev arkadaşı arayan diğer öğrencilerin bilgilerini(yaş, cinsiyet, okuduğu bölüm ve sınıf) görebilir ve onlarla uygulama içinden mesajlaşma özelliği sayesinde iletişime geçebilir. Bu sayede panodan ev arkadaşı arama devri kapanmış olacak. Son olarak projemizin duyu analizi modülü sayesinde öğrenciler yurtlara yapılan yorumlardan olumlu ve olumsuz olanları çok rahat bir şekilde görebilirler. Ayrıca bu modül sayesinde yurt sahipleride olumlu ve olumsuz yorumların analizlerini kolayca yapabilecekler ve gerekli aksiyonları alabiliceklerdir.

Anahtar Kelimeler: kalacak yer, yurt yorumları, ev arkadaşı, duyu analizi

ACKNOWLEDGEMENT

My dear supervisor, who does not spare his interest and support in the planning, research, execution, and formation of this project, whose vast knowledge and experience I have benefited from, I would like to express my eternal and sincere thanks to Fatih Erdoğan Sevilgen.

Also, I would like to express my endless and sincere thanks to my teachers, Erchan Aptoula, Mehmet Burak Koca, and Ferda Abbasoğlu, who are leading the way for the improvement of this project.

Lastly, I would like to express my respect and love to my family, who supported me in every way during my education, and to all my teachers who set an example for me with their lives.

Akif Kartal

LIST OF SYMBOLS AND ABBREVIATIONS

Abbreviation : Explanation

GTU	: Gebze Technical University
API	: Application Program Interface
HTTP	: Hyper Text Transfer Protocol

CONTENTS

Abstract	iv
Özet	v
Acknowledgement	vi
List of Symbols and Abbreviations	vii
Contents	ix
List of Figures	x
List of Tables	xi
1 Introduction	1
1.1 Project Definition	1
1.2 The Goal of the Project	1
2 Project Details	2
2.1 Users of the Application	2
2.2 Project Requirements	2
2.2.1 Functional Requirements	2
2.2.2 Non-Functional Requirements	3
2.3 Project Design Plan and Architecture	3
2.3.1 Tools and Technologies	4
2.4 Context Model	4
2.5 Use Case Diagram	5
3 Sentiment Analysis	6
3.1 Dataset	6
3.2 Creating The Model and Testing	6
3.3 Exporting and Deploying The Model to Public	7
3.4 Resources That Benefited From	8
4 Database	9
4.1 Collecting Dormitory Information Data	9

4.2	Cloud Firestore Database	9
4.2.1	Database Tables	9
5	Mobile Application	11
5.1	Screens and Features	11
5.1.1	Sign up and Login	11
5.1.2	Side Menu	13
5.1.3	Dormitory List	14
5.1.4	Dormitory Details	15
5.1.5	Dormitory Comments	16
5.1.6	Reporting an Error and Report List	17
5.1.7	Finding Housemate	18
5.1.8	Chat with Other Students	19
5.1.9	User Profile	20
6	Success Criteria	22
6.1	Criterion 1	22
6.2	Criterion 2	22
6.3	Criterion 3	24
6.3.1	Testing with Confusion Matrix	24
6.3.1.1	Splitting Data	24
6.3.1.2	Creating Confusion Matrix and Testing	25
6.3.2	Testing with Turkish Sentences	25
6.3.2.1	Turkish Sentences and Success Results	25
6.3.3	Sentiment Analysis Results in Literature	27
6.4	Criterion 4	27
6.4.1	Login Response Time	28
6.4.2	Updating Student Profile Response Time	28
6.4.3	Sentiment Analysis API Response Time	29
6.5	Summary of Success Criteria Results	30
7	Conclusions	31
	Bibliography	32

LIST OF FIGURES

2.1	Project Architecture	3
2.2	Tools and Technologies	4
2.3	Context Model of the Project	5
2.4	Use Case Diagram	5
4.1	Table Relations in Database	10
4.2	Real Data and Tables in Firestore Database	10
5.1	Mobile Application Logo	11
5.2	Sign Up and Login Part 1	12
5.3	Sign Up and Login Part 2	13
5.4	Side Menu and Options	14
5.5	List of All Dormitories	15
5.6	Dormitory Information Screen	16
5.7	Dormitory Comments Screen	17
5.8	Error Reporting	18
5.9	Finding Housemate Screens	19
5.10	Chat With Other Students	20
5.11	Updating User Profile Information	21
6.1	Details about Application Size in Google Play Store	23
6.2	Mobile Application Size Graph on Google Play Console	23
6.3	Confusion Matrix Representation ¹	24
6.4	Counting How Many Successful Sentiment Results We Have	26
6.5	Sentiment Analysis API Response Time in Postman	29

LIST OF TABLES

2.1	Functional Requirements	2
2.2	Non-Functional Requirements	3
6.1	Confusion Matrix Accuracy Result	25
6.2	Some of My Test Turkish Dormitory Comments and Sentiment Results	26
6.3	Overall Accuracy for Sentiment Analysis	26
6.4	Sentiment Analysis Results in Literature [3]	27
6.5	My Sentiment Analysis Result	27
6.6	Database Login Response Time Result	28
6.7	Database Response Time Results	29
6.8	Overall Backend Response Time	30
6.9	Success Summary of All Results	30

1. INTRODUCTION

Nowadays, especially after the COVID19 pandemic finding a place to stay is much difficult for students in the university. This project is aimed to make easier this process with a powerful user experience and safe way for GTU¹ students.

1.1. Project Definition

In this project, I need to make a mobile application so that GTU students can find a place such as a dormitory around the university to stay. With this mobile application, my aim is to make this process easier and fast way. In order to do that I need to have a database to store our data such as user data and dormitory information. Besides this, I need to code a mobile application in a user-friendly way. After coding a mobile application and creating a database I need to connect them. Besides all of these, I need to have a sentiment analysis module to evaluate comments made by the students for dormitories. Lastly, the students who are looking for a housemate should chat with each other so that they can easily find a housemate.

1.2. The Goal of the Project

Making reason for this project is to make it easier to find a place to stay and communicate with each other for GTU students with an easy-to-use mobile application.

By using this mobile application, the students can have the following options.

- Students can find a place such as public or private dormitory around the university to stay.
- Students can find housemates and chat with them via this application.
- Students can see positive and negative comments about dormitories.
- Students and Owners can report any new/missing information about dormitories and follow status of the report.

¹Gebze Technical University

2. PROJECT DETAILS

In order to make this project I need to consider and determine details about this project.

2.1. Users of the Application

Any registered user can use this application and all features.

2.2. Project Requirements

I have the following functional and non-functional requirements for this project.

2.2.1. Functional Requirements

Requirement ID	Description of the Requirement
FR1	Collecting data about dormitories around GTU
FR2	The user shall be able to sign up and log in
FR3	The user shall be able to see the list of dormitories and details about them
FR4	The user shall be able to comment and see other comments about dormitories
FR5	The user shall be able to like or unlike comments for dormitories
FR6	The comments for dormitories shall be sorted by their number of like
FR7	The user shall be able to see positive and negative comments about dormitories and filter them
FR8	The user shall be able to see a list of students who are looking for a housemate and see their information(age, gender etc.)
FR9	The user shall be able to chat with other students
FR10	The user shall be able to report missing/wrong information about dormitories
FR11	The user shall be able to follow status of the his/her reports
FR12	The user shall be able to update his/her profile information

Table 2.1: Functional Requirements

2.2.2. Non-Functional Requirements

Requirement ID	Description of the Requirement
NFR1	User must have an android device
NFR2	User's device must have a stable internet connection
NFR3	Application needs to have a database to keep data
NFR4	Application should be fast enough to use
NFR5	Application should be easy to use and user experience should be simple enough so that even the elderly users is able to catch up fast.
NFR6	Information about the dormitories should be updated
NFR7	Application don't copyright infringement about dormitory images

Table 2.2: Non-Functional Requirements

2.3. Project Design Plan and Architecture

This project has more than one module, such as mobile application module, database module, and sentiment analysis module. I have created the following architecture to get all of these modules together so that they can run.

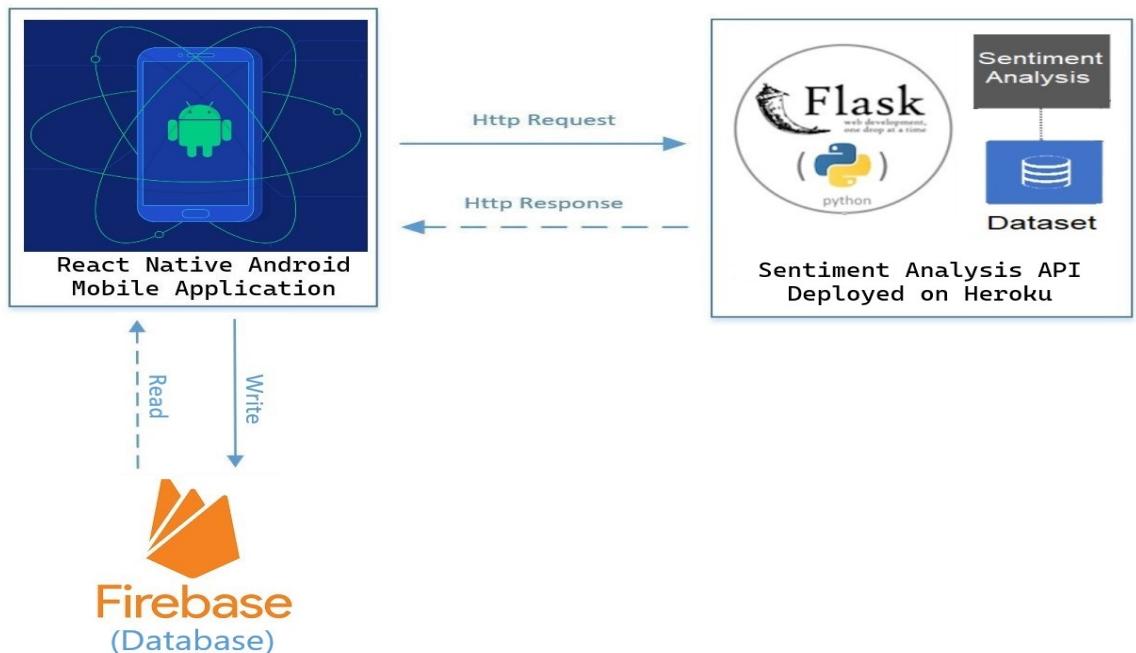


Figure 2.1: Project Architecture

2.3.1. Tools and Technologies

In order to make this project following tools and technologies have been used.



Figure 2.2: Tools and Technologies

1. **React Native and Javascript:** These are used to make the mobile application.
2. **Firebase:** Database of the project to keep data.
3. **Python, Colab, Flask, and Heroku:** These are used to make the sentiment analysis module of the project.
4. **Visual Studio Code:** This is used as an editor to code the mobile application.
5. **Git and Github:** These are used to keep source code and to deploy api on heroku.¹

2.4. Context Model

The context model is the physical scope of the system being designed². In the following picture, we can see the context model of this project.

¹Heroku is a free cloud platform to deploy and host your application.

²https://en.wikipedia.org/wiki/Context_model

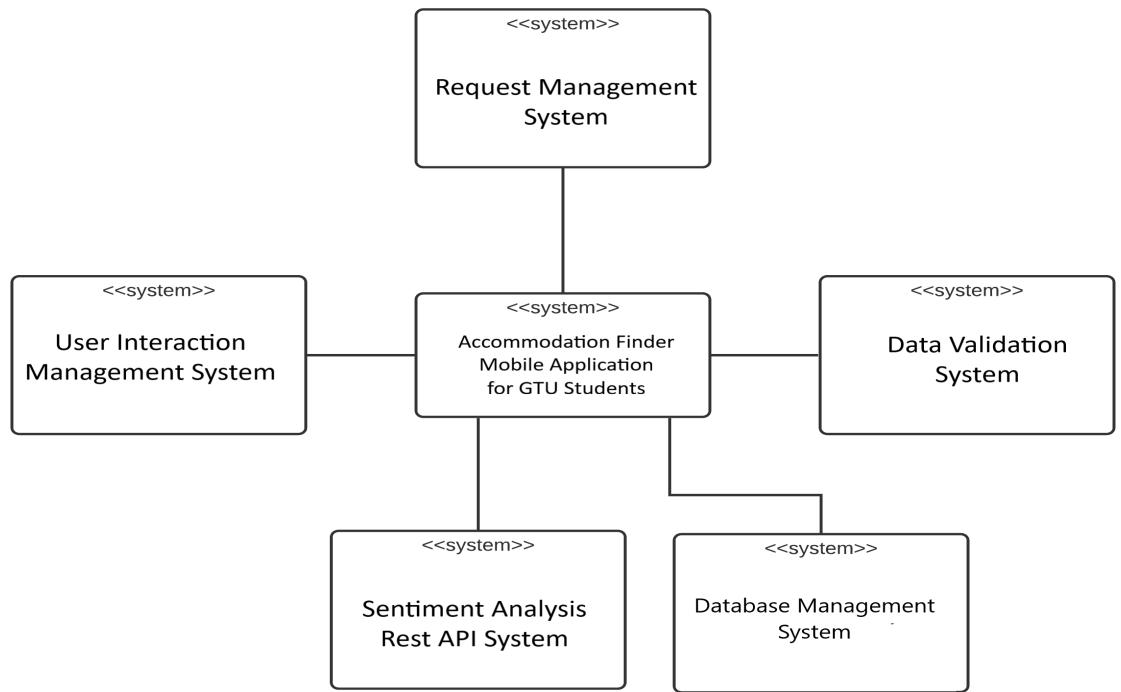


Figure 2.3: Context Model of the Project

2.5. Use Case Diagram

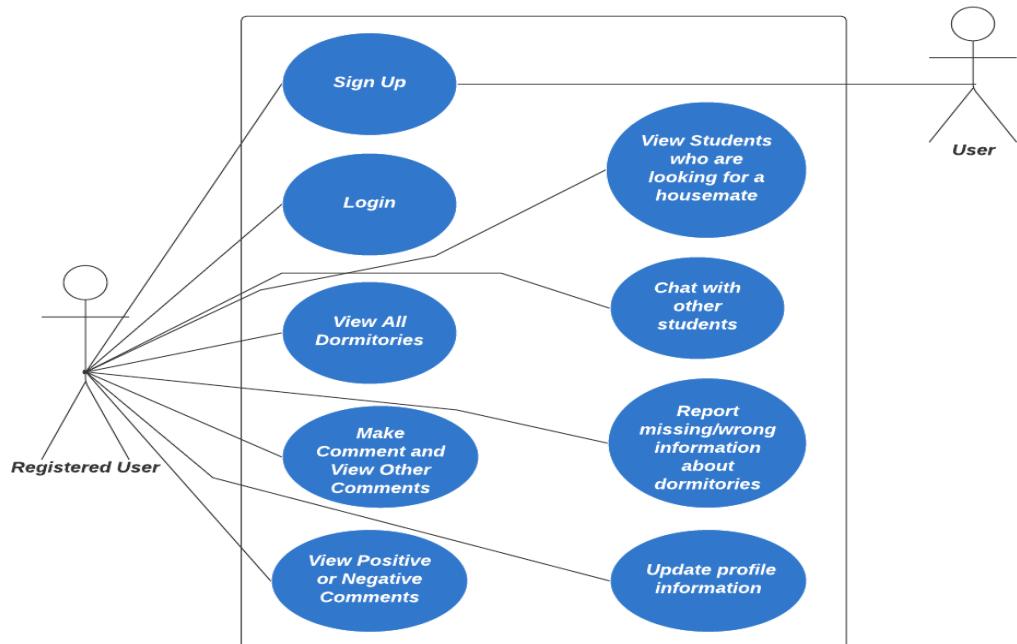


Figure 2.4: Use Case Diagram

3. SENTIMENT ANALYSIS

In this module, I tried to determine whether a dormitory comment is positive or negative. In order to do this, I have to use some Natural Language Processing methods. In my solution, in order to create word vectors, I have used TF-IDF(term frequency-inverse document frequency) vectorizer and Logistic Regression as a Classifier from python scikit-learn library. In order to complete this module following steps have been applied.

1. Find a good dataset
2. Clear data and remove noise in the dataset
3. Split train and test data
4. Create and train a model by using TF-IDF vectorizer and Logistic Regression
5. Test that model accuracy
6. Export and deploy the model

Next we will see these steps in detailed way.

3.1. Dataset

As we know our comments will be in Turkish. But there was no good dataset in Turkish for dormitory comments all we have are movie reviews or product reviews in Turkish. Therefore while I was searching for a good dataset I have found the this dataset.¹ In this English dataset, we have 35038 hotel comments dataset with the response(positive or negative). For this module, I will use this dataset by translating Turkish comments into English.

3.2. Creating The Model and Testing

After finding the dataset, I can create and train our model. Before that, I need to clear and remove noise in the data. In the following code pieces, we can see these steps.

¹<https://www.kaggle.com/anu0012/hotel-review?select=train.csv>

```

1 from sklearn.feature_extraction.text import TfidfVectorizer
2 from sklearn.linear_model import LogisticRegression
3 from sklearn.pipeline import Pipeline
4
5 tvec = TfidfVectorizer(ngram_range=(1, 1))
6 clf2 = LogisticRegression()
7
8 model = Pipeline([('vectorizer',tvec),
9                   ('classifier',clf2)])
10
11 model.fit(attribute_train, target_train)

```

Listing 3.1: Creating Model

In following code piece, I will test my model by using Confusion Matrix. In this test my test data will be %10 of train data. For more information check chapter 6.3.

```

1 from sklearn.metrics import confusion_matrix
2 from sklearn.metrics import accuracy_score
3
4 verdict = model.predict(attribute_test)
5 confusion_matrix(verdict, target_test)
6
7 print("Accuracy : ", accuracy_score(verdict, target_test))

```

Listing 3.2: Testing Model with Confusion Matrix

3.3. Exporting and Deploying The Model to Public

In order to use this model in my mobile application I need deploy it to the public so that my application can use it. In order to do this first I need to export my model into a file by using python **pickle**¹ library.

```

1 import pickle
2 path='/content/drive/MyDrive/Colab Notebooks/model2/'
3 pickle.dump(model, open(path+'model.pkl','wb'))

```

Listing 3.3: Exporting Model Using Pickle into a File

¹<https://docs.python.org/3/library/pickle.html>

After exporting our model I can use that model in our API. In the following code piece we can see how I am using the exported model file to integrate with my mobile application.

```
1 import pickle
2 from flask import Flask
3 from flask import jsonify
4 from flask import request
5 from googletrans import
6
7 app = Flask(__name__)
8 model = pickle.load(open('model.pkl', 'rb'))
9 translator = Translator()
10
11 @app.route("/api/v1/getResult", methods=['POST'])
12 def getresult():
13     request_data = request.get_json()
14     comment = request_data['comment']
15     com_en = translator.translate(comment, dest="en").text
16     com_en = com_en.replace("country", "dormitory")
17     com_en = [com_en]
18     result = model.predict(com_en)
19     return jsonify({'result': result[0]})
```

Listing 3.4: Use of Exported Model to Evaluate Translated Comment in API

3.4. Resources That Benefited From

In order to implement this sentiment analysis module, I have benefited from these resources.[1]–[6]

4. DATABASE

In order to store data, we need a database in this project. But the important thing is this database should be **reachable** from anywhere which means it shouldn't be a local database on any computer. Therefore cloud databases is good for this project. In-state of the art **Google firebase platform** is mostly used for this purpose because it has powerful options and is free for the starter package. Firebase works with two different databases. The Real-Time Database is the original Firebase database product, and Cloud Firestore is a new and improved version of the Real-Time Database.¹ In this project, we have used the Cloud Firestore database of Firebase.

4.1. Collecting Dormitory Information Data

One of the requirements of this project collecting data about dormitories that are around Gebze Technical University. In order to do this, I have collected data by calling each dormitory and talking with the responsible person about that dormitory. In this process, **some of the people avoided giving information about that dormitory**. Therefore there can be some missing and wrong information about dormitories.

4.2. Cloud Firestore Database

Cloud Firestore is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps at a global scale.²

4.2.1. Database Tables

Below we have the tables and relations between them in the figure 4.1. As a relation, we have one-to-one and one-to-many relations between tables. Also in tables, most data types are the **string to manage** data easily in the application.

We can see real data in the figure 4.2. On the left side, we have 4 database tables on the center, we have each separate data in table with its unique id on the right side, we have the context of the data.

¹<https://blog.back4app.com/firebase-vs-firebase/>

²<https://firebase.google.com/products/firestore>

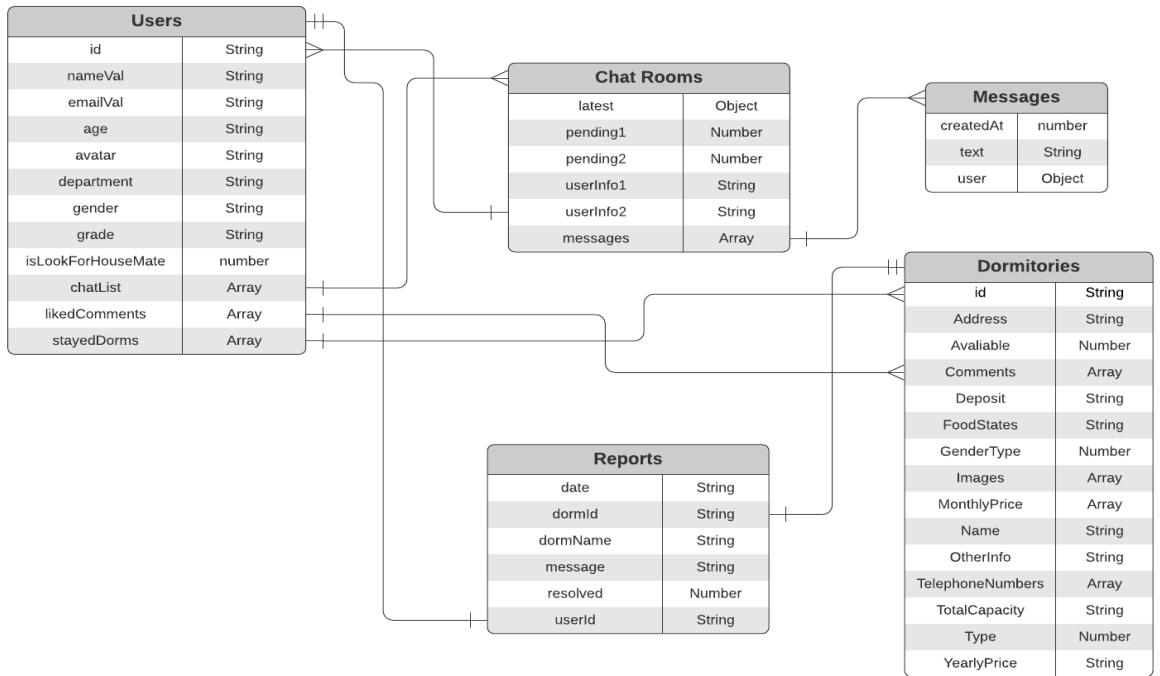


Figure 4.1: Table Relations in Database

Collection	Document ID	Fields
cse495-5ea49	users	<ul style="list-style-type: none"> + Start collection chatRooms dormitories reports users <pre> + Add document 3nyBDrt...V96pwm9V0bnP9sk... 6ZLPFhVW0GTPD1Z8SG7DwJ1Pt8... B3zUXhU5PzQUQHLJohiLFdKK4T... FyvdzKpGndX77aVzYjwC1Q5YxT... JIgLhjQQFUT1cXsQqQSMHMjJ7j... UgusxqeVWjcd53FfGim15t7mWg... VgZpgu00qGZus9J5K7ReaLmqbFl... W9S4vMbC2YZRa092eIltk8KPG0... WE7Z8W8wjNew0tiNpLbxqrCqaz... lce4yBzNdj0dMyujEQb3KgXkvD... mtWCD83i72NWJJQzVSQ0dNRHFV... nErNSb70QIU3YimusPKw97xEkB... oCGmyXs01LTJpizzUuplsX2YjE... putu1t2hHDdFitskSjoX8bg8gg... </pre> <ul style="list-style-type: none"> + Start collection + Add field age: "24" avatar: "https://bootdey.com/img/Content/avatar/avatar7.png" chatList department: "Bilgisayar Mühendisliği" emailVal: "akokartal@gmail.com" gender: "Erkek"

Figure 4.2: Real Data and Tables in Firestore Database

5. MOBILE APPLICATION

Making the mobile application is the most involved part of this project and to make mobile application I have used react native technology which is one of the heavily used technology in business.

One of the critical part of react native is **you need to make everything with code** which means **there is no drag and drop** feature for the components such as buttons etc.



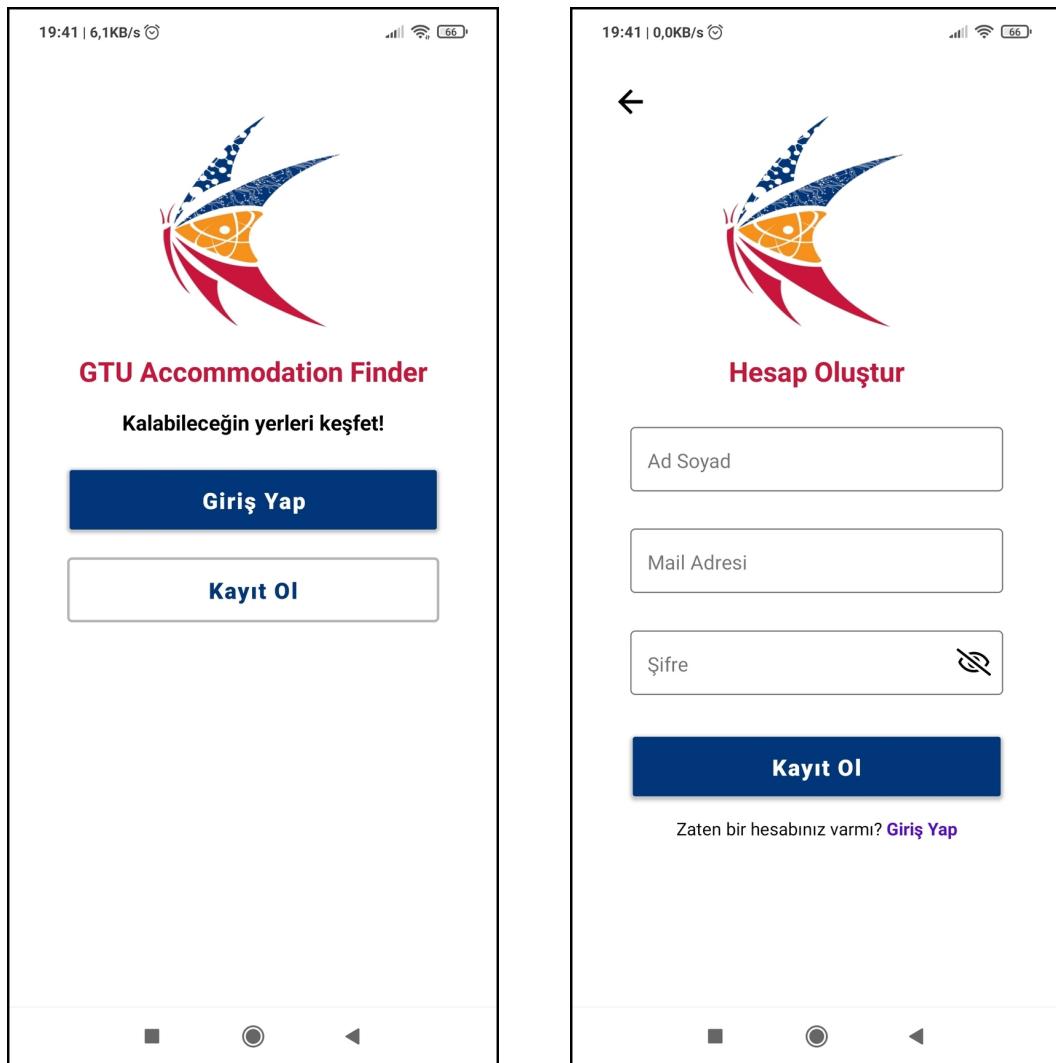
Figure 5.1: Mobile Application Logo

5.1. Screens and Features

Next, we will see each screen and features of the application one by one.

5.1.1. Sign up and Login

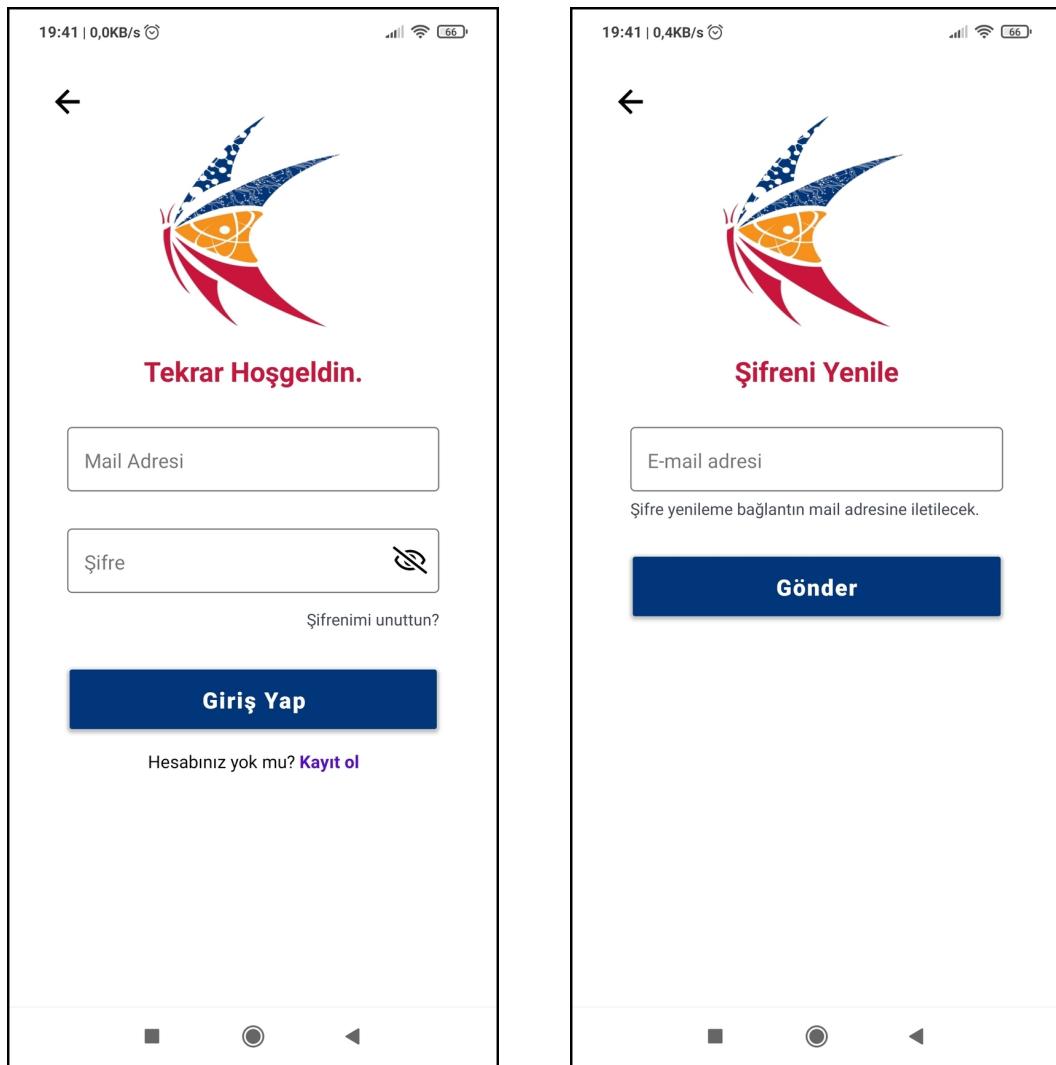
In the following images, we can see mobile application entrance screens.



(a) Entrance Screen

(b) Sign Up Screen

Figure 5.2: Sign Up and Login Part 1



(a) Login Screen

(b) Password Change Screen

Figure 5.3: Sign Up and Login Part 2

In order to implement sign up and login screens **user interface**, I have **benefited from** this resource [7]. Note that, I didn't use this resource directly. I just benefited and improved it for my requirements.

5.1.2. Side Menu

In the following image, we can see mobile application side menu and options for any user.

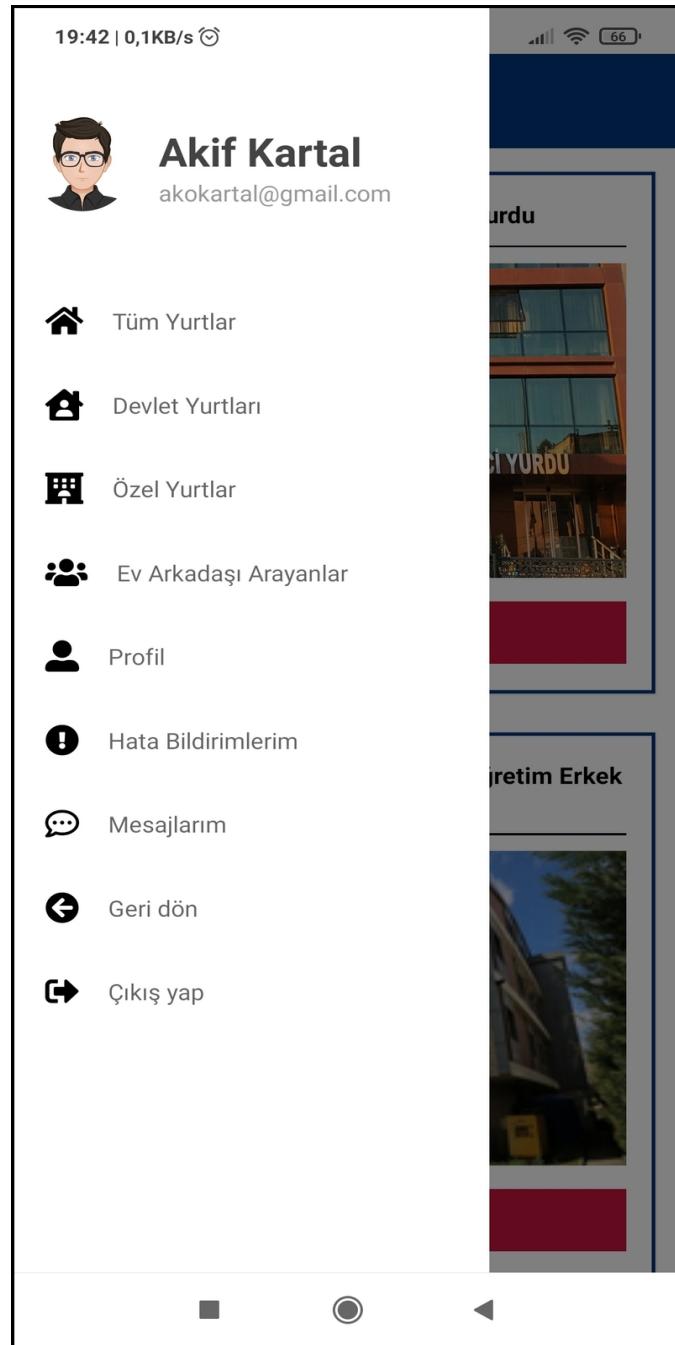


Figure 5.4: Side Menu and Options

5.1.3. Dormitory List

In the following image, we can see dormitory list screen.

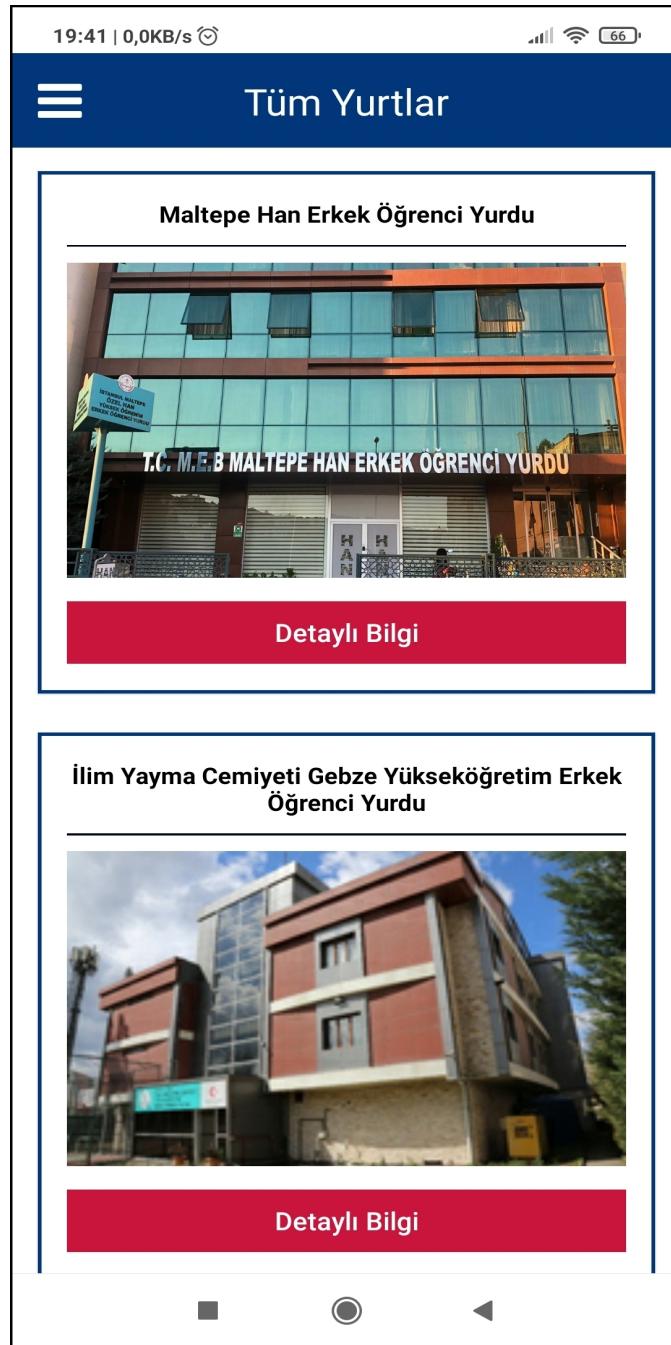


Figure 5.5: List of All Dormitories

5.1.4. Dormitory Details

After clicking the more information button in the dormitory list screen we can see information about that dormitory. In the following images, we can see which information do we have for a dormitory.

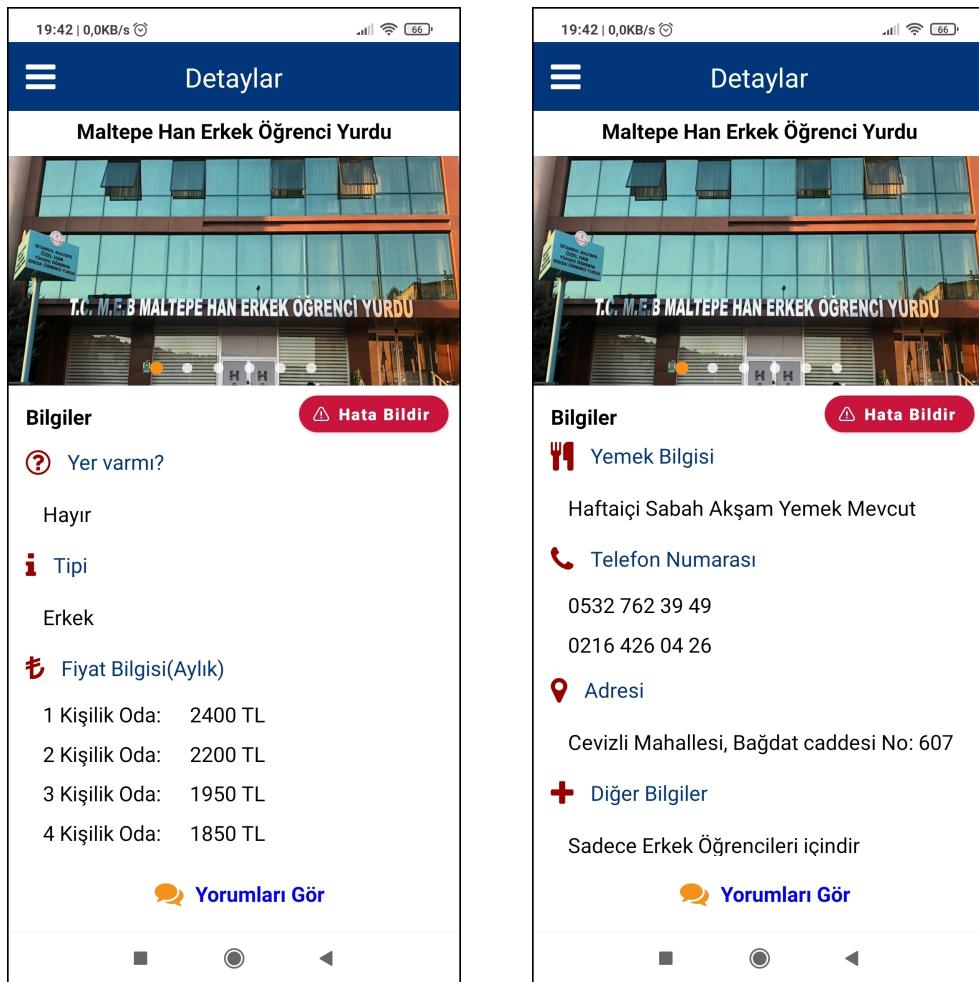
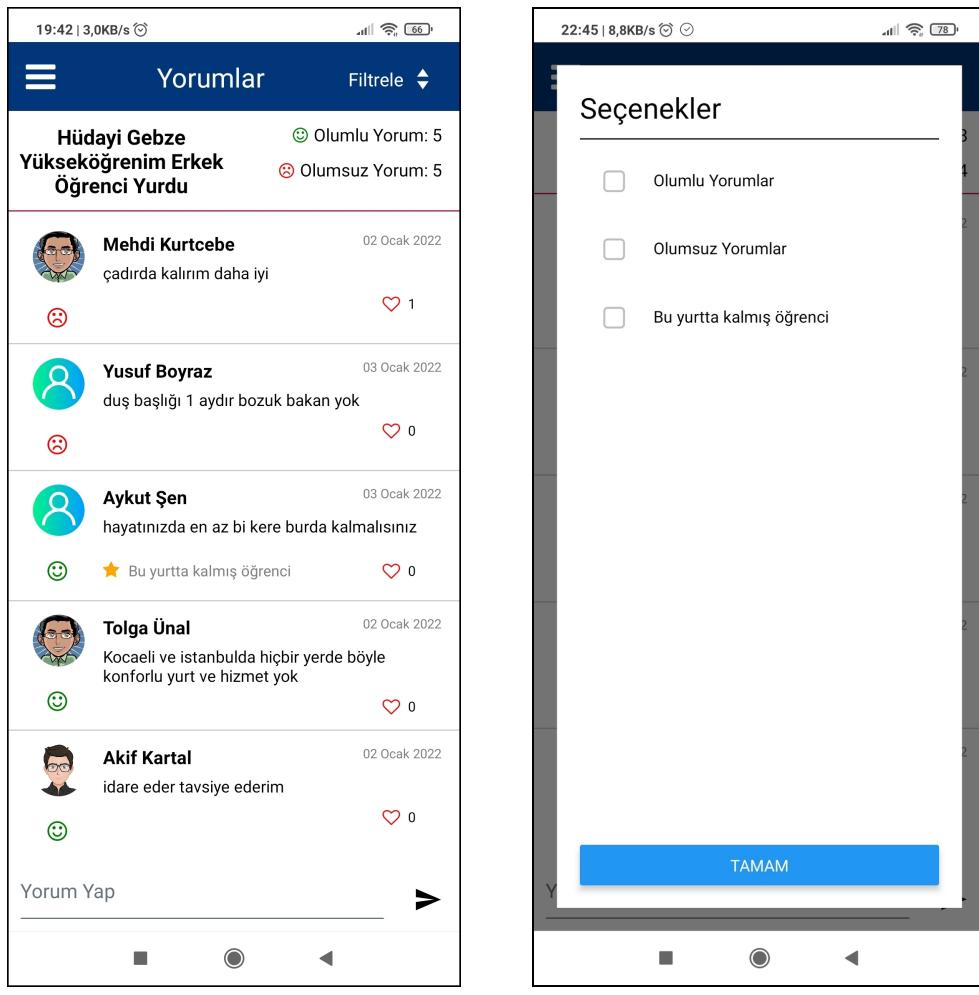


Figure 5.6: Dormitory Information Screen

5.1.5. Dormitory Comments

After clicking the see comment button in the dormitory information screen the user can see comments about that dormitory. In the following image, we can see the comment screen.

In the comment screen, the user can make a comment, like or unlike a comment, filter the comments, and also can see which comment is positive or negative.



(a) Dormitory Comments

(b) Filters For Comments

Figure 5.7: Dormitory Comments Screen

In order to implement comment list **user interface**, I have **benefited from** this resource [8]. Note that, I didn't use this resource directly. I just benefited and improved it for my requirements.

5.1.6. Reporting an Error and Report List

In the dormitory information screen, we have report error button. After clicking the report error button in the dormitory information screen we can report an error about dormitory information. After reporting we can follow the report status from my reports screen.

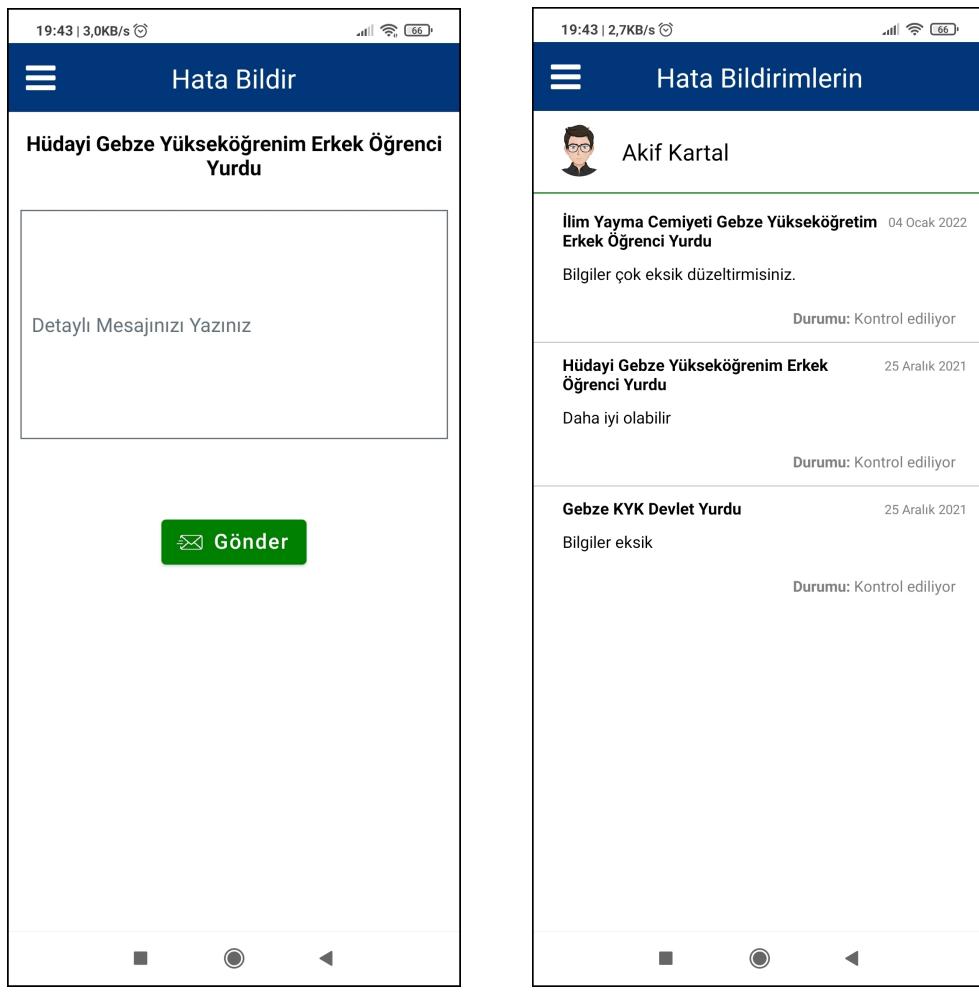


Figure 5.8: Error Reporting

5.1.7. Finding Housemate

In this mobile application, you can see the list of students who are looking for a housemate and also chat with them. In the student list screen when you click the more information button of a student you can see more detail about that student and also you can start a new chat with that student.

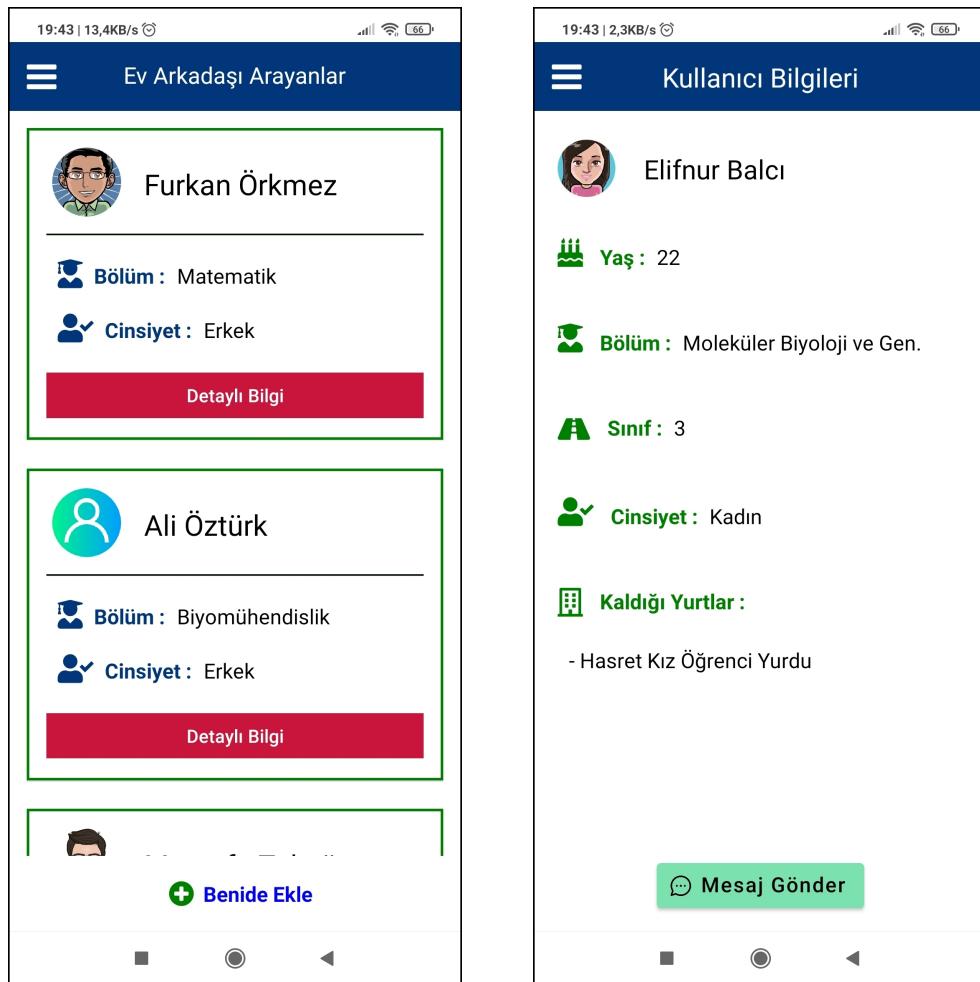


Figure 5.9: Finding Housemate Screens

5.1.8. Chat with Other Students

After clicking the send message button in the student information screen the user can start a chat with that student. After starting a chat, user can see his/her chats on the message list screen. In the following images, you can see this feature more detailed way.

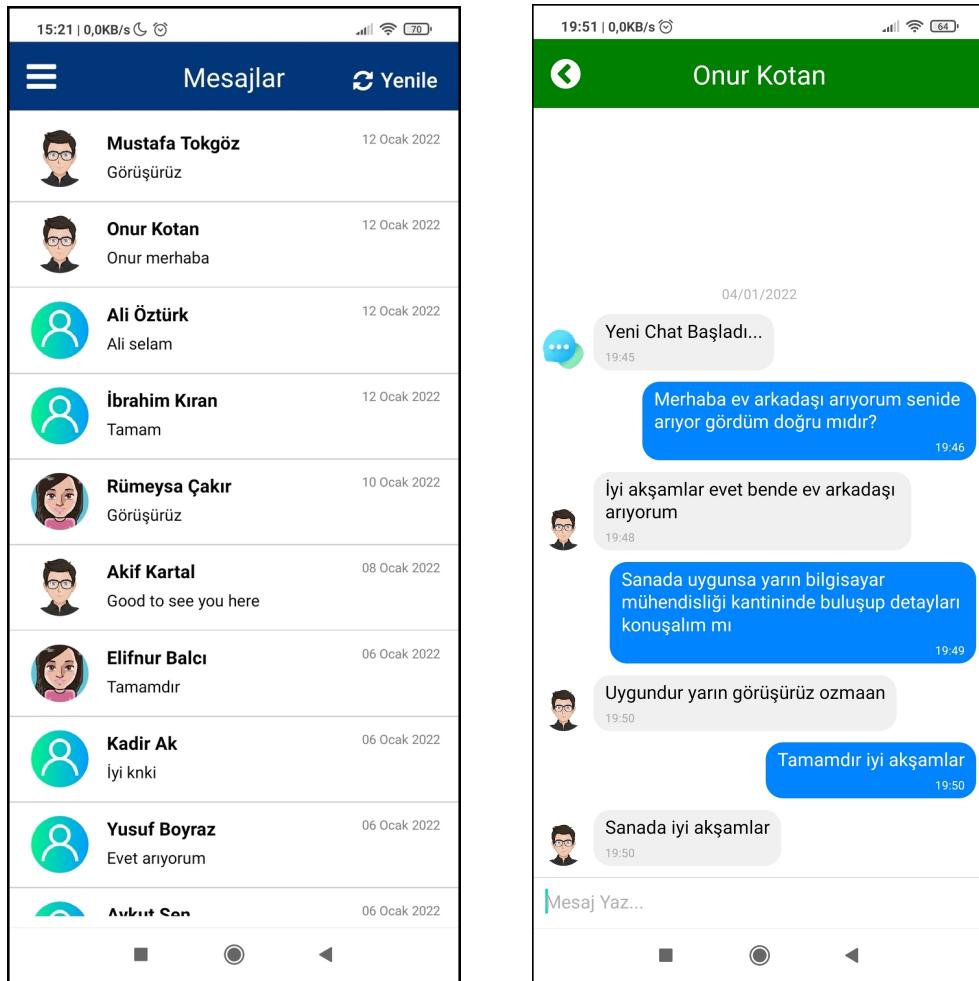
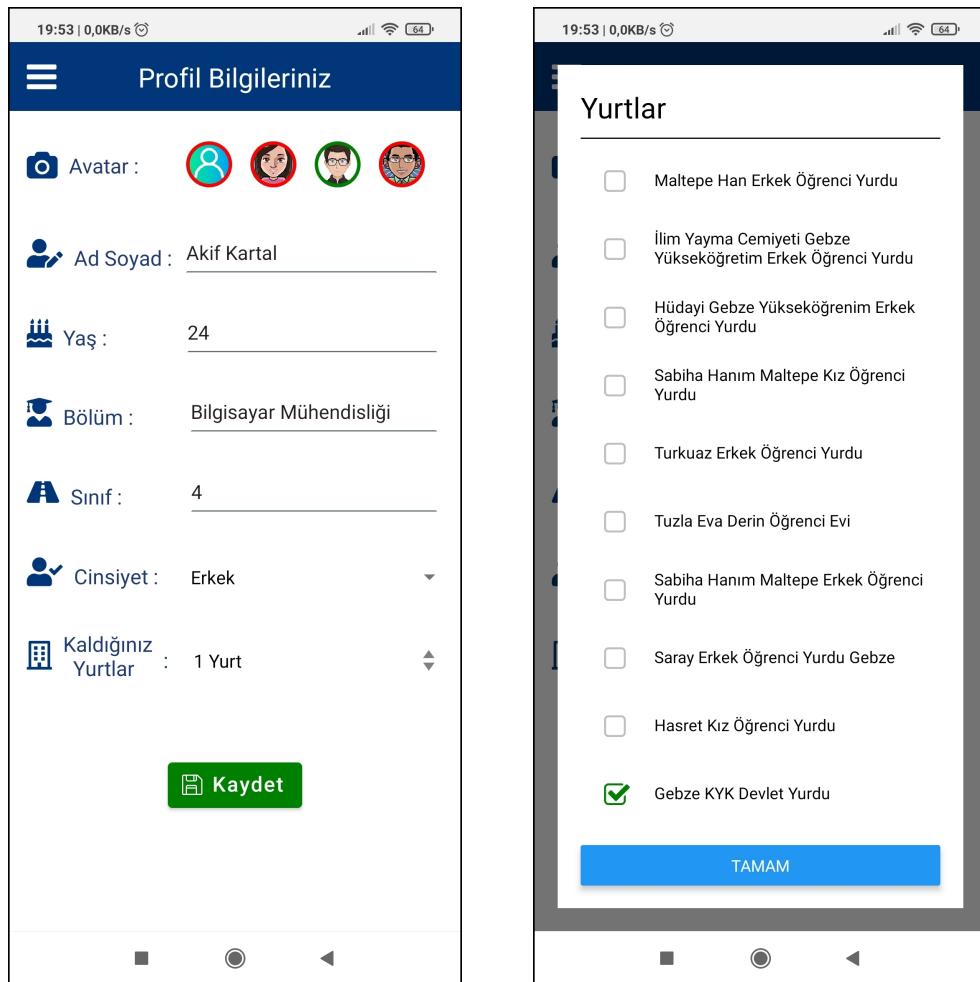


Figure 5.10: Chat With Other Students

In order to make **live chat screen user interface** I have used this library [9]. Note that all messages are stored in my database and message list implementation belongs to me.

5.1.9. User Profile

As a user experience rule, I don't take too much information in the sign-up stage. But the user can update his/her profile information after login or sign-up. Therefore I have created a profile screen. On this screen, the user can update his/her information. In the following images, you can see this screen.



(a) Profile Information Page

(b) Selecting Dormitories Which The User Has Stayed Before

Figure 5.11: Updating User Profile Information

6. SUCCESS CRITERIA

For this project, we have determined 4 success criteria. These are;

1. At least 8 different screens in the mobile application.
2. Mobile Application size will be less than 100 MB.
3. At least %60 sentiment analysis accuracy.
4. At most 3.5 seconds (3500 ms.) backend(database) response time.

Next, we will see that how I have accomplished these criteria one by one.

6.1. Criterion 1

- At least 8 different screens in the mobile application.

I have accomplished this criterion successfully. In the mobile application, we have **14 different** screens. These are start, login, sign up, forgot password, dormitory list, housemate list, user information, chat, chat list, user profile, error reports list, dormitory information, error reporting, and dormitory comments. For more information check chapter 5.1.

6.2. Criterion 2

- Mobile Application size will be less than 100 MB.

I have accomplished this criterion successfully. In order to show this, I have uploaded my mobile application on the google play store.¹ In the google play store, **my application download size is 29,93 MB.**

Check the application size in the following images.

¹<https://play.google.com/store/apps/details?id=com.cse495.GTUAccommodationFinder>

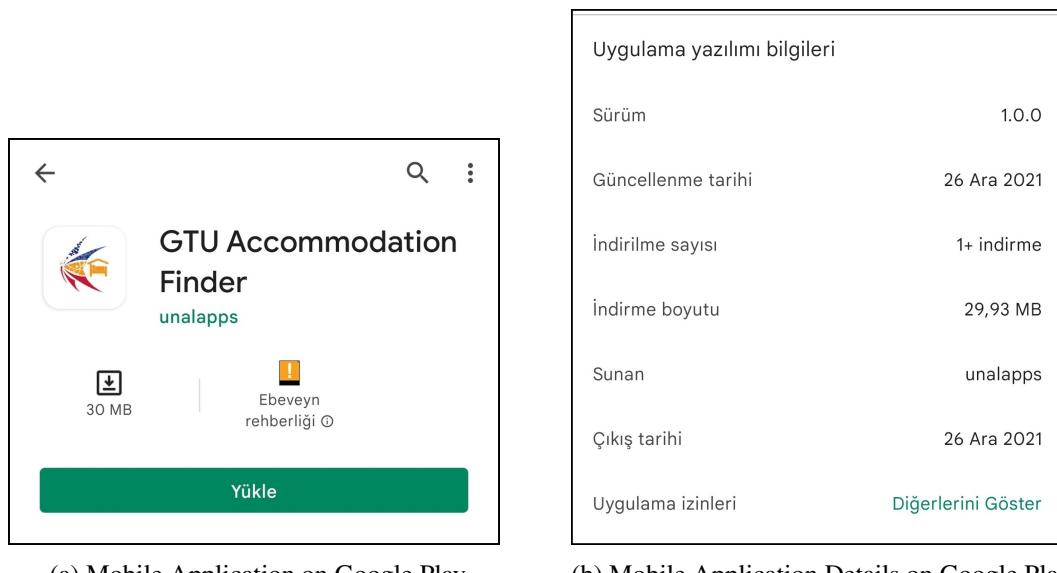


Figure 6.1: Details about Application Size in Google Play Store

In following graph, which is taken from google play console¹ we can see my total application size. according to Google play reports my total application size is **33.7 MB.**

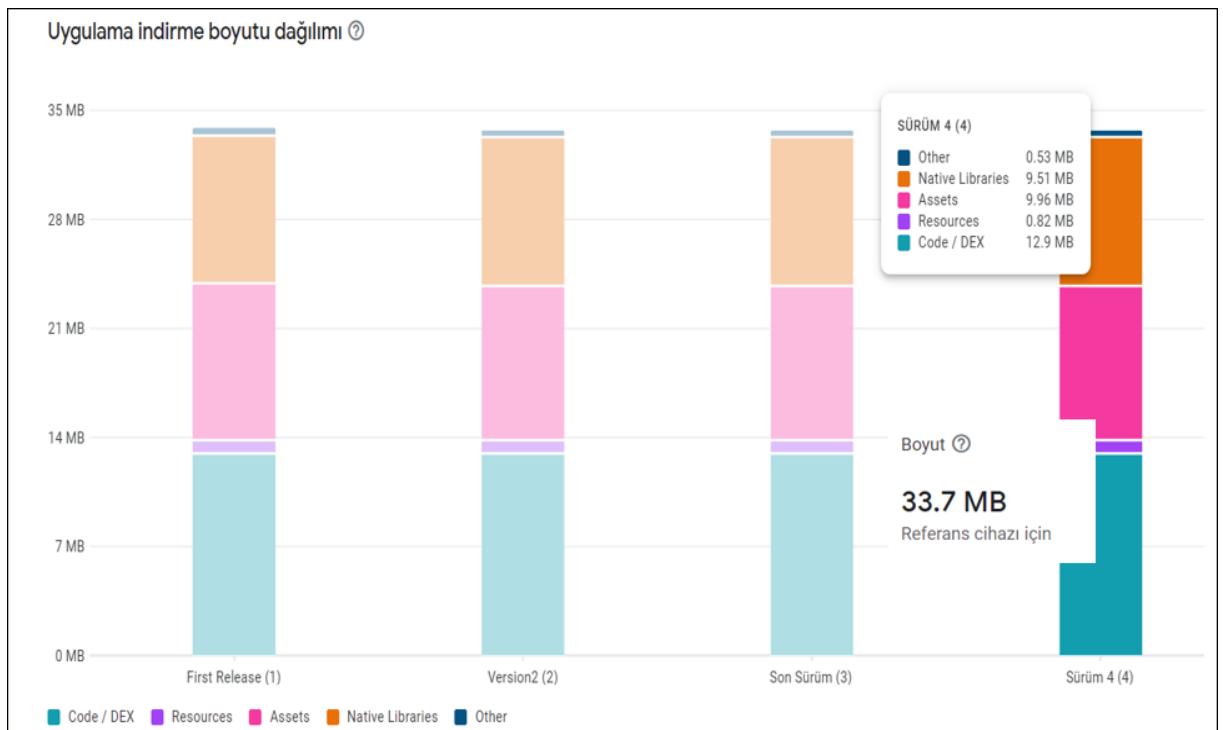


Figure 6.2: Mobile Application Size Graph on Google Play Console

¹<https://play.google.com/console/about/>

6.3. Criterion 3

- At least %60 sentiment analysis accuracy.

I have accomplished this criterion successfully. In order to show this, I tested my model with 2 different test method. Below we will see this methods one by one.

6.3.1. Testing with Confusion Matrix

In this method, I will test my model accuracy by using Confusion Matrix in figure 6.3 with test data from original dataset. In this test I will use %10 of dataset as a test data and then we will get accuracy result. Note that our dataset in english therefore our test data will be english.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 6.3: Confusion Matrix Representation¹

Success Accuracy and Error Rate formula in Confusion Matrix

$$\text{Accuracy} = (TN + TP) / (TN + TP + FN + FP)$$

$$\text{Error Rate} = (FN + FP) / (TN + TP + FN + FP)$$

6.3.1.1. Splitting Data

In this test, I will split our train and test data. My test size is 0.1 which means %10 of train set will be my test data.

¹<https://emilia-orellana44.medium.com/breakdown-confusion-matrix-2cf25842f1ae>

6.3.1.2. Creating Confusion Matrix and Testing

In the following code, we will create confusion matrix and calculate the result.

```
1 from sklearn.metrics import confusion_matrix
2 from sklearn.metrics import accuracy_score
3
4 verdict = model.predict(attribute_test)
5 confusion_matrix(verdict, target_test)
6
7 print("Accuracy : ", accuracy_score(verdict, target_test))
```

Listing 6.1: Testing Model with Confusion Matrix

Test Method	Accuracy
Confusion Matrix with English Sentences	%88

Table 6.1: Confusion Matrix Accuracy Result

6.3.2. Testing with Turkish Sentences

In this method, I will test my model accuracy by using my Turkish 100 sentences. In this test first I will determine each sentence value such as positive or negative and after labeling each sentence we will give these sentences into our model and we will compare the results and check whether they produced correct results or not.

6.3.2.1. Turkish Sentences and Success Results

I have created 100 Turkish dormitory comments and tested them in my mobile application. In the following table, you can see the results for some of the comments.

Comment	Result
hapishane sanki gece girip çıkmak yasak	Pass
güzel vakit geçiyor öneririm	Pass
oda değiştirmek çok zor önermem	Pass
uzak durun buradan pişman olursunuz	Fail
bir deneyin burayı pişman olmazsınız	Fail
süper bir mazarası var	Pass
burda kalacağınızı bir parkta yatın	Fail
hayat tecrübe kazanmak isteyen gelsin	Pass
şehir merkezine ve çarşıya çok uzak dag başı resmen	Fail
cadırda kalırım daha iyi	Pass

Table 6.2: Some of My Test Turkish Dormitory Comments and Sentiment Results

As we have seen, we have 100 Turkish sentences and some of them are failed in our sentiment analysis model. In order to get the accuracy we only need to count how many successful sentences we have. In the following image, you can see the number of successful sentences. Since we have 76 successful sentences out of 100 sentences we have **%76 accuracy** on 100 Turkish sentences.

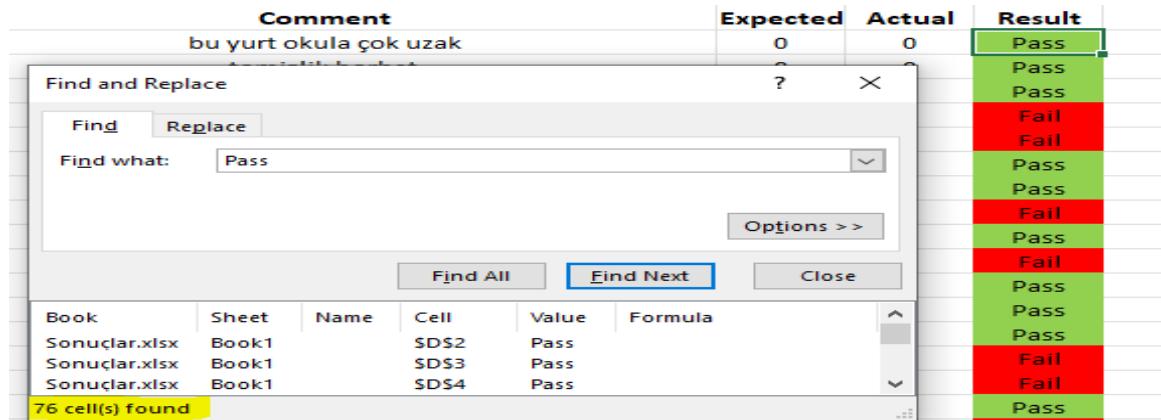


Figure 6.4: Counting How Many Successful Sentiment Results We Have

Test Method	Accuracy
Confusion Matrix with English Sentences	%88
100 Turkish Dormitory Comment	%76

Table 6.3: Overall Accuracy for Sentiment Analysis

As a result, we had good accuracy in 2 test method but still there are critical errors to be solved in sentiment analysis module of this project.

6.3.3. Sentiment Analysis Results in Literature

When I search for a similar solution to my sentiment analysis module I found this article¹ which is very similar to my case. Note that they didn't share their dataset. In this article, they have used **Word2Vec and Recurrent Neural Networks(RNN)** instead of TF-IDF and Logistic Regression. In the following table, you can see the comparison between my solution accuracy and their accuracy.

Test Method	Accuracy
Private Word Vector Model	%91
Public Word Vector Model	%92

Table 6.4: Sentiment Analysis Results in Literature [3]

Test Method	Accuracy
100 Turkish Dormitory Comment	%76

Table 6.5: My Sentiment Analysis Result

As a result, we can improve our sentiment analysis with better datasets and algorithms.

6.4. Criterion 4

- At most 3.5 seconds (3500 ms.) backend(database) response time.

I have accomplished this criterion successfully. In order to show this, I have used 2 technique. These techniques are

1. I will measure database response time by measuring taken time in source code.
2. I will measure sentiment analysis response time by using postman program and by measuring taken time in source code.

¹<https://dergipark.org.tr/tr/download/article-file/852974>

6.4.1. Login Response Time

In the following code, I am measuring the login control response time of the firebase database. I am calculating time using javascript performance.now()¹ method and after successful login, we print the result on the console. As you can see we **successfully logged in 1.57 seconds.**

```
1  var t0 = performance.now();
2  Firebase.auth()
3  .signInWithEmailAndPassword(email.value, password.value)
4  .then((userCredentials) => {
5      var t1 = performance.now();
6      console.log("Login time " + (t1 - t0) + " milliseconds.");
7      ...
8  })
```

Listing 6.2: Measure Login Response Time in Source Code

Test Function	Result
Login Response Time	1.57 seconds

Table 6.6: Database Login Response Time Result

6.4.2. Updating Student Profile Response Time

In the following code, I am measuring the updating of the student profile information and writing them in the firebase database. Below, I am using the updateUser method which uses the firebase update method and writes the new information(passed by parameter) into the database. After successfully updating, we printed the result on the console. As you can see **we can update the database in 1.20 seconds.**

```
1  var t0 = performance.now();
2  updateUser(info.id, info)
3  .then((docRef) => {
4      var t1 = performance.now();
5      console.log("Update Student Time: " + (t1 - t0) + " ms.");
6      ...
7  })
```

Listing 6.3: Measure Updating Student Profile Response Time in Source Code

¹<https://developer.mozilla.org/en-US/docs/Web/API/Performance/now>

Test Function	Result
Login Response Time	1.57 seconds
Updating Student Profile Response Time	1.20 seconds

Table 6.7: Database Response Time Results

6.4.3. Sentiment Analysis API Response Time

In order to measure this first, I will use Postman ¹ program.

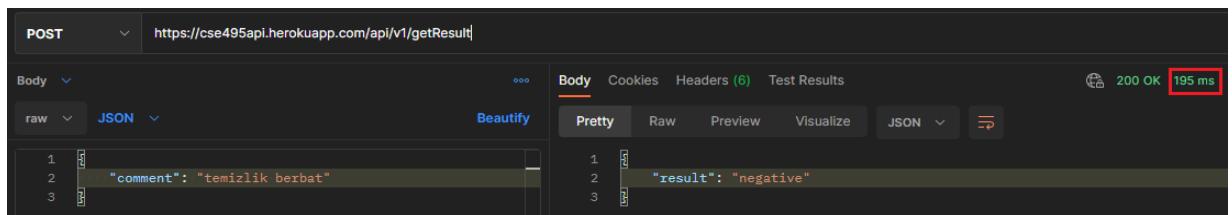


Figure 6.5: Sentiment Analysis API Response Time in Postman

As you can see in the above image we have **195ms** response time which is very low from my API².

Below, I am measuring sentiment analysis API response time in source code. In this code, I am using the getResult method which connects my API and gets the result. After getting the result in the finally block, we printed the result on the console. As you can see we can get sentiment analysis results from my API in **2.38** seconds.

```

1  try {
2      var t0 = performance.now();
3      const response = await getResult(comment);
4      result = await response.json();
5  } catch (error) {
6      alert("Bir hata olustu. Lutfen tekrar deneyin.");
7  } finally {
8      var t1 = performance.now();
9      console.log("API Response Time: " + (t1 - t0) + " ms.");
10 }

```

Listing 6.4: Measure Sentiment Analysis API Response Time in Source Code

¹https://www.postman.com/downloads/?utm_source=postman-home

²<https://cse495api.herokuapp.com/>

Test Function	Result
Login Response Time	1.57 seconds
Updating Student Profile Response Time	1.20 seconds
Sentiment Analysis API Response Time in Postman	0.19 seconds
Sentiment Analysis API Response Time in Source Code	2.38 seconds

Table 6.8: Overall Backend Response Time

As a result, in all tests we don't have any response time bigger than 3.5 seconds.

6.5. Summary of Success Criteria Results

In the following table, we can see the all success criteria expected and actual results. Note that in the sentiment analysis success criterion we got the expected accuracy but **it is open to improvement**.

Success Criterion	Expected	Actual	Result
At least 8 different screens in the mobile application	At least 8	14	Successful
Mobile Application size will be less than 100 MB	At most 100 MB	33.7 MB	Successful
At least %60 sentiment analysis accuracy	At least %60	%76	Successful
At most 3.5 seconds(3500 ms.) backend response time.	At most 3.5s	1.57s, 1.20s, 2.38s	Successful

Table 6.9: Success Summary of All Results

7. CONCLUSIONS

In this project, I have tried to solve and make it easy to find a place to stay while studying in university for GTU Students. As we have seen this problem is mostly a software engineering problem, therefore, I had to take action according to this and I did mostly.

In order to solve this problem I have applied the following steps;

1. Define Requirements
2. Make a design(both visual and architectural)
3. Divide design into modules
4. Code modules one by one
5. Test each module
6. Combine modules
7. Deploy application
8. Maintenance application

As an engineer in the design step, I have considered both visual design and architectural design. My visual design is user-friendly and meets the requirements as expected. On the other hand in the architectural design step, I have chosen the most appropriate technologies for the project and heavily used ones. For example, in order to integrate the sentiment analysis module with this project, I have used a request-response mechanism with HTTP protocol by using python flask technology.

By dividing the big project into modules I have conquered each small piece so that my problems were small to solve and test. After finishing the modules I have created this project.

As a result, while making such a project the key point is being agile about both changes and learning technologies since time is limited and you can't avoid changes.

BIBLIOGRAPHY

- [1] https://github.com/RaihanAk/Hotel-Review-Sentiment-Analysis_MachineLearning.
- [2] <https://github.com/BaharYilmaz/turkce-duygu-analizi>.
- [3] <https://dergipark.org.tr/tr/download/article-file/852974>.
- [4] <https://towardsdatascience.com/a-beginners-guide-to-sentiment-analysis-in-python-95e354ea84f6>.
- [5] <https://medium.com/analytics-vidhya/sentiment-analysis-on-amazon-reviews-using-tf-idf-approach-c5ab4c36e7a1>.
- [6] <https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/>.
- [7] <https://github.com/venits/react-native-login-template>.
- [8] <https://www.bootdey.com/react-native-snippet/35/Comments-list-ui-example>.
- [9] <https://www.npmjs.com/package/react-native-gifted-chat>.
- [10] <https://reactnavigation.org/>.
- [11] <https://reactnativepaper.com/>.
- [12] <https://reactnativeelements.com/>.
- [13] <https://fontawesome.com/v5.15/icons?d=gallery&p=2>.
- [14] <https://github.com/bezkoder/react-firebase-database-crud>.
- [15] <https://www.crowdbotics.com/blog/add-real-time-database-to-react-native-app-with-firebase>.
- [16] <https://www.npmjs.com/package/react-native-image-slider-box>.
- [17] <https://dev.to/chinmayhatre/how-to-generate-apk-using-react-native-expo-kae>.
- [18] <https://www.veribilimiokulu.com/hata-matrisini-confusion-matrix-yorumlama/>.