GTU COMP ENG CSE 495/496

Accommodation Finder Mobile Application for GTU Students

Final Presentation

Akif Kartal

Project Advisor: Prof. Dr. Fatih Erdoğan SEVİLGEN

January 2022

Project Definition

What was this project?

This was a mobile application so that,

- GTU students can find a place such as public or private dormitory, house for rent around the university to stay.
- Students can find housemates and chat with them via this application.
- Owners of private dormitories can report any new/missing information about their dormitories.
- Users can make comments, and see other comments as well as their sentiment results such as positive or negative. Also, they can like them and filter them.

GTU Accommodation Finder

Kalabileceğin yerleri keşfet!

What we did?

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

Project Architecture 1

At the beginning, we have planned creating following architecture for this project .

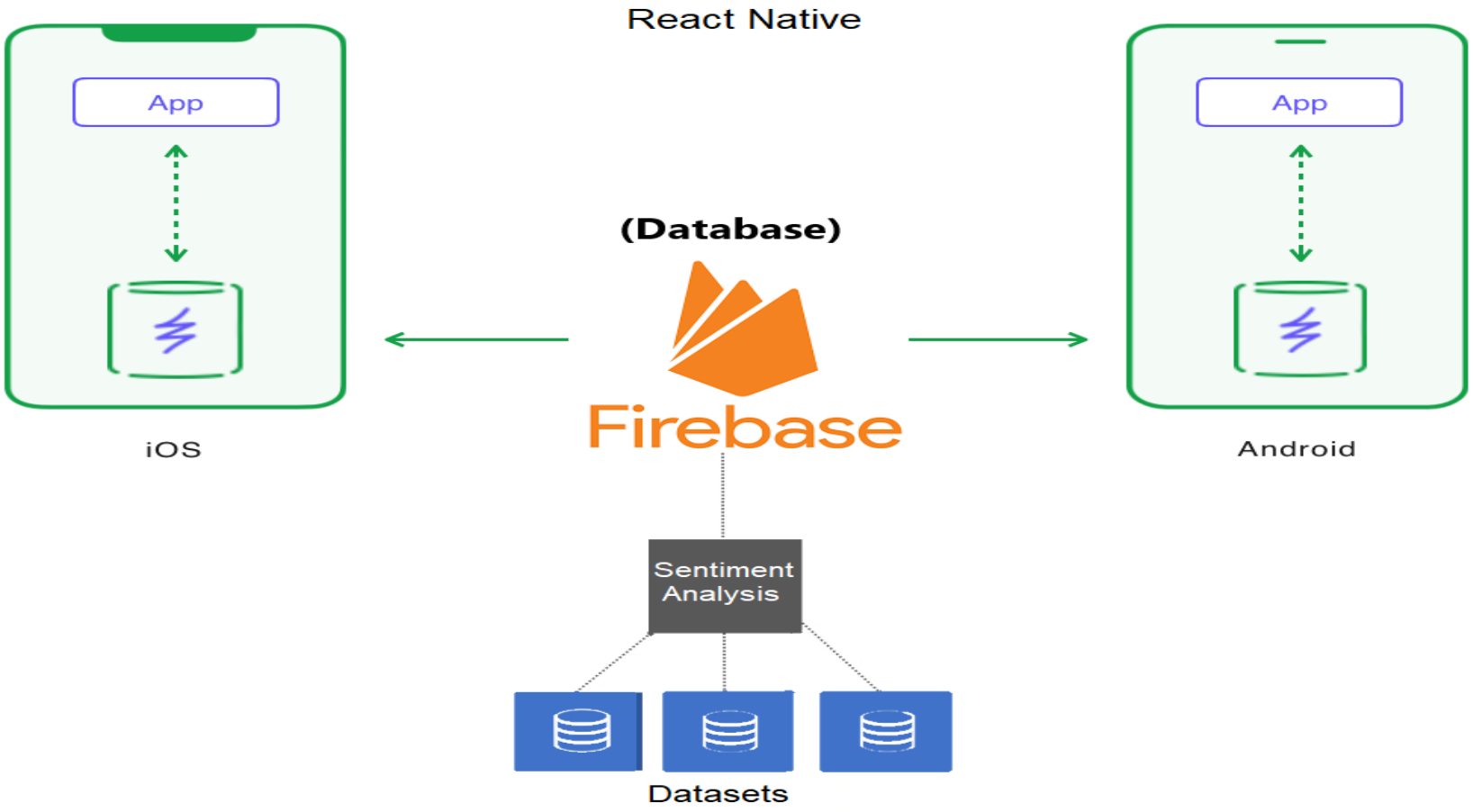


Figure 1.1: Project Architecture in the Beginning

Project Architecture 2

We created following architecture for this project.

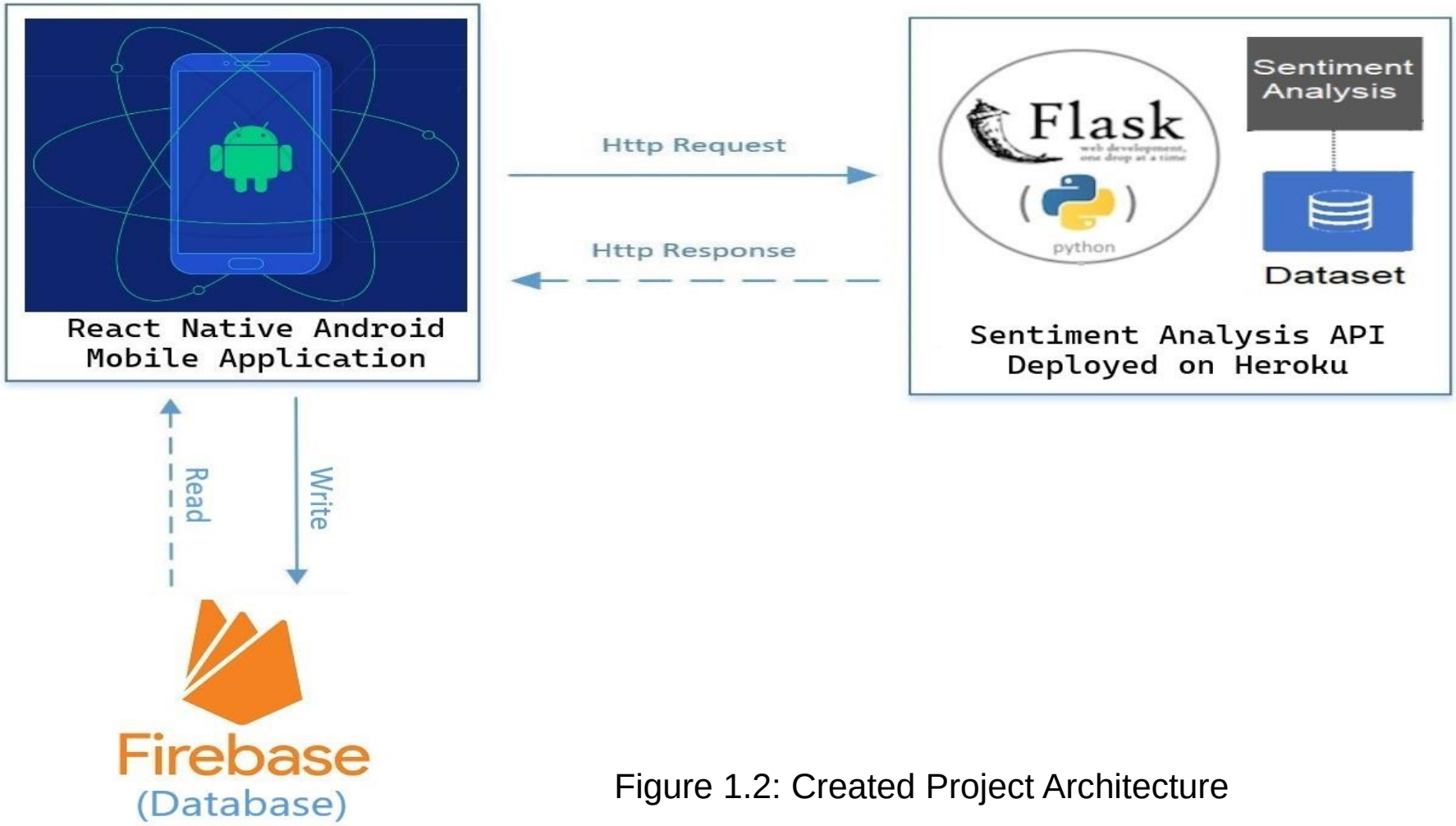
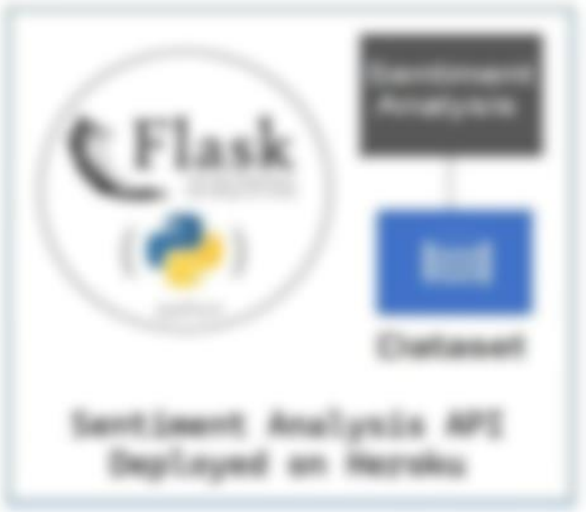


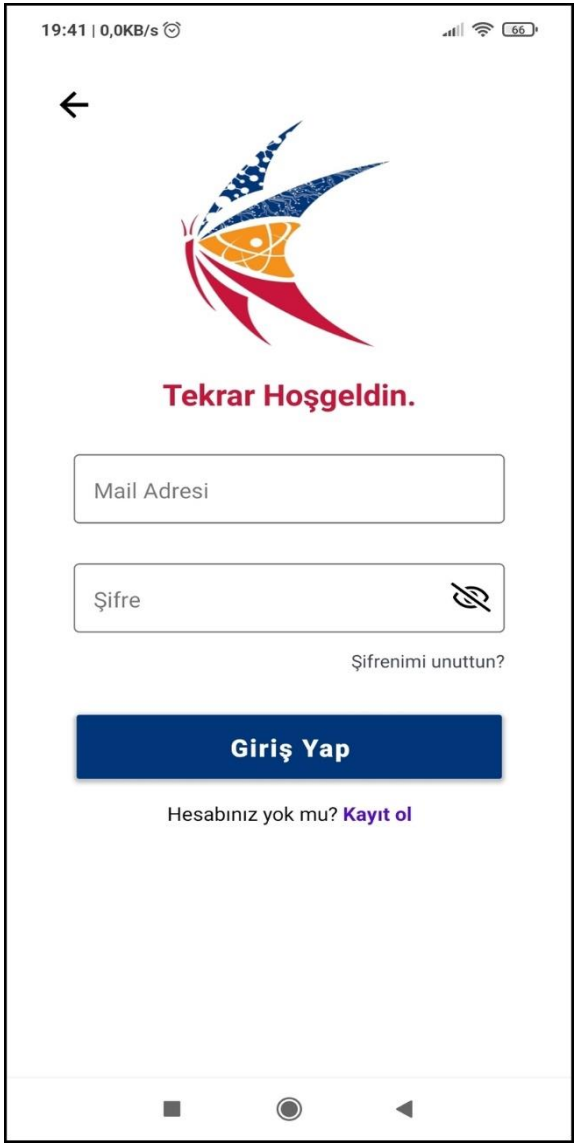
Figure 1.2: Created Project Architecture

Mobile Application



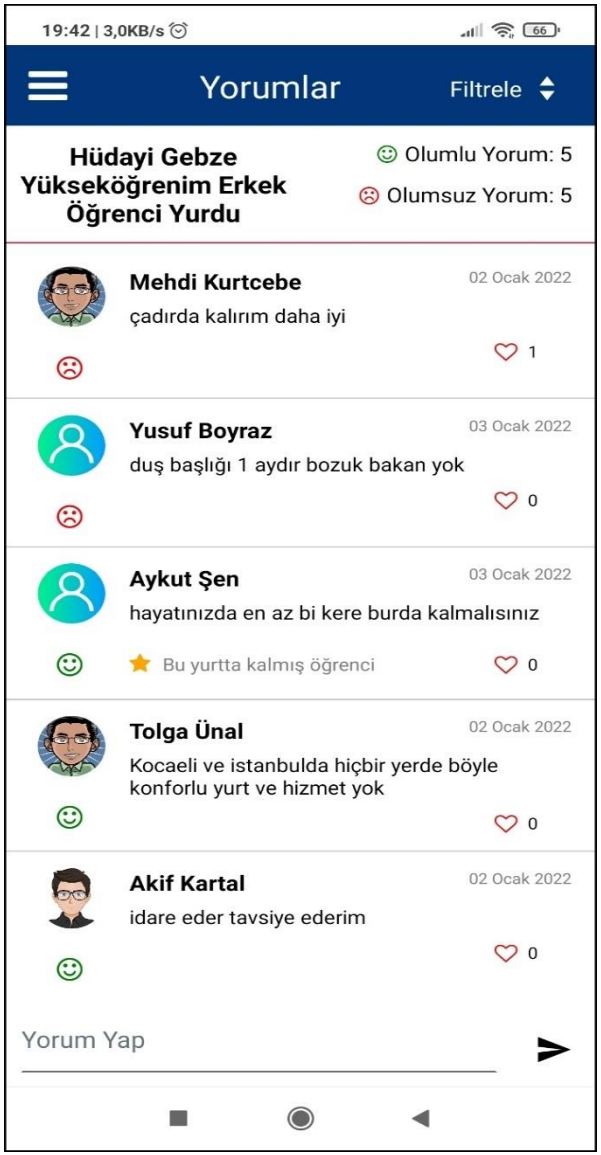
Mobile Application Part 1

- The user should be able to sign up and log in
- The user should be able to see the list of dormitories.
- The user should be able to see information about dormitories.



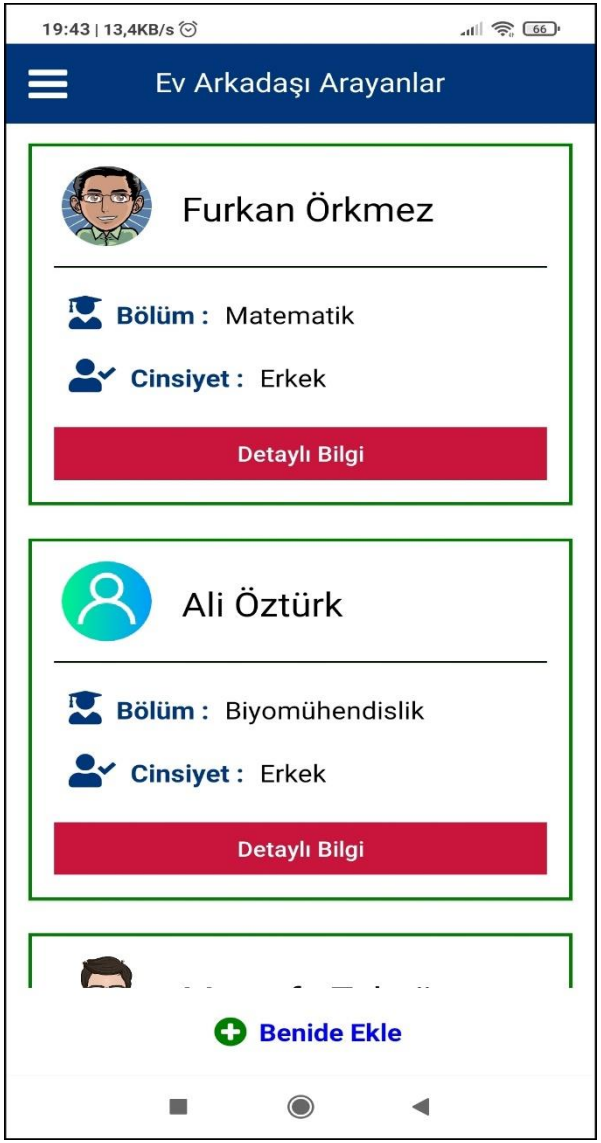
Mobile Application Part 2

- The user should be able to comment and see other comments about dormitories.
- The user should be able to like or unlike comments for dormitories
- The comments for dormitories should be sorted by their number of like
- The user should be able to see positive and negative comments about dormitories and filter them



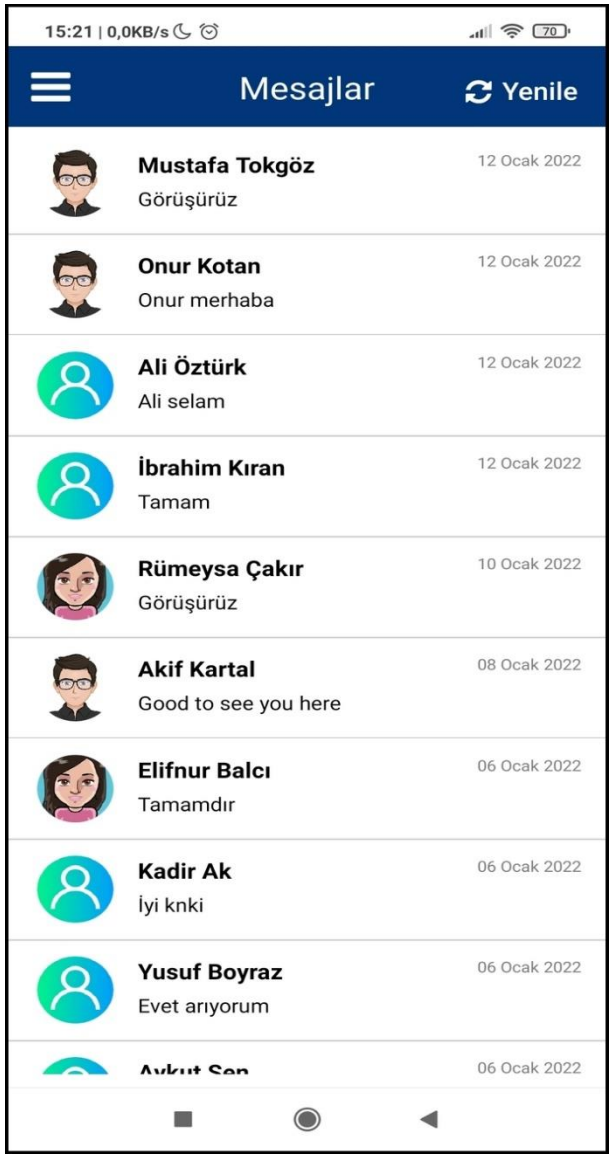
Mobile Application Part 3

- The user should be able to see a list of students who are looking for a housemate and see their information(age, gender etc.)



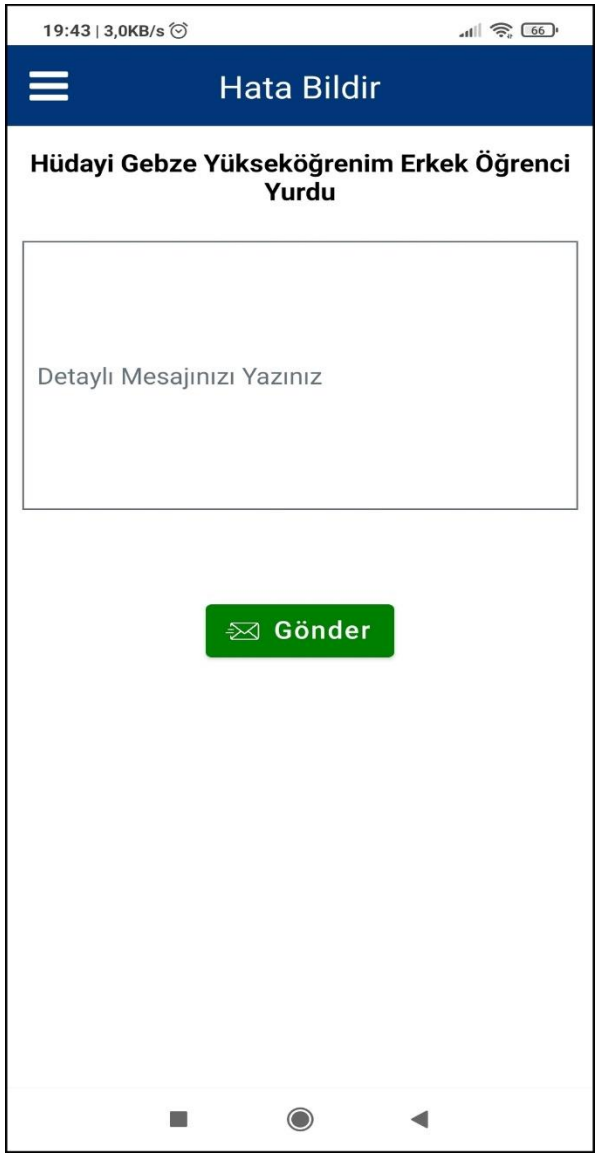
Mobile Application Part 4

- The user should be able to chat with other students



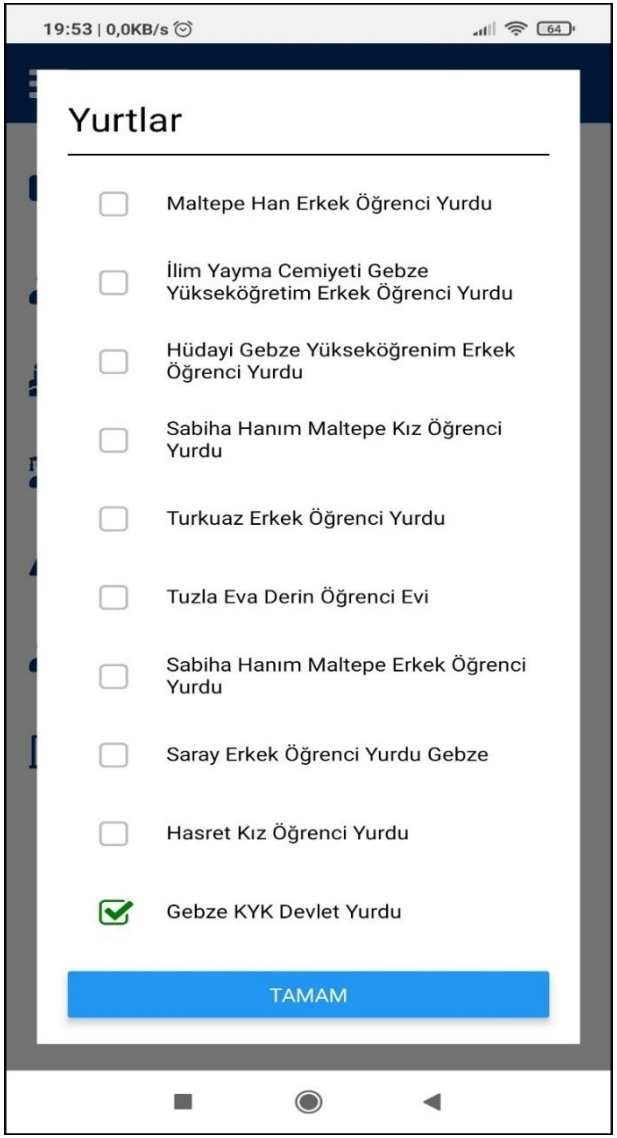
Mobile Application Part 5

- The user should be able to report missing/wrong information about dormitories
- The user should be able to follow status of the his/her reports



Mobile Application Part 6

- The user should be able to update his/her profile information



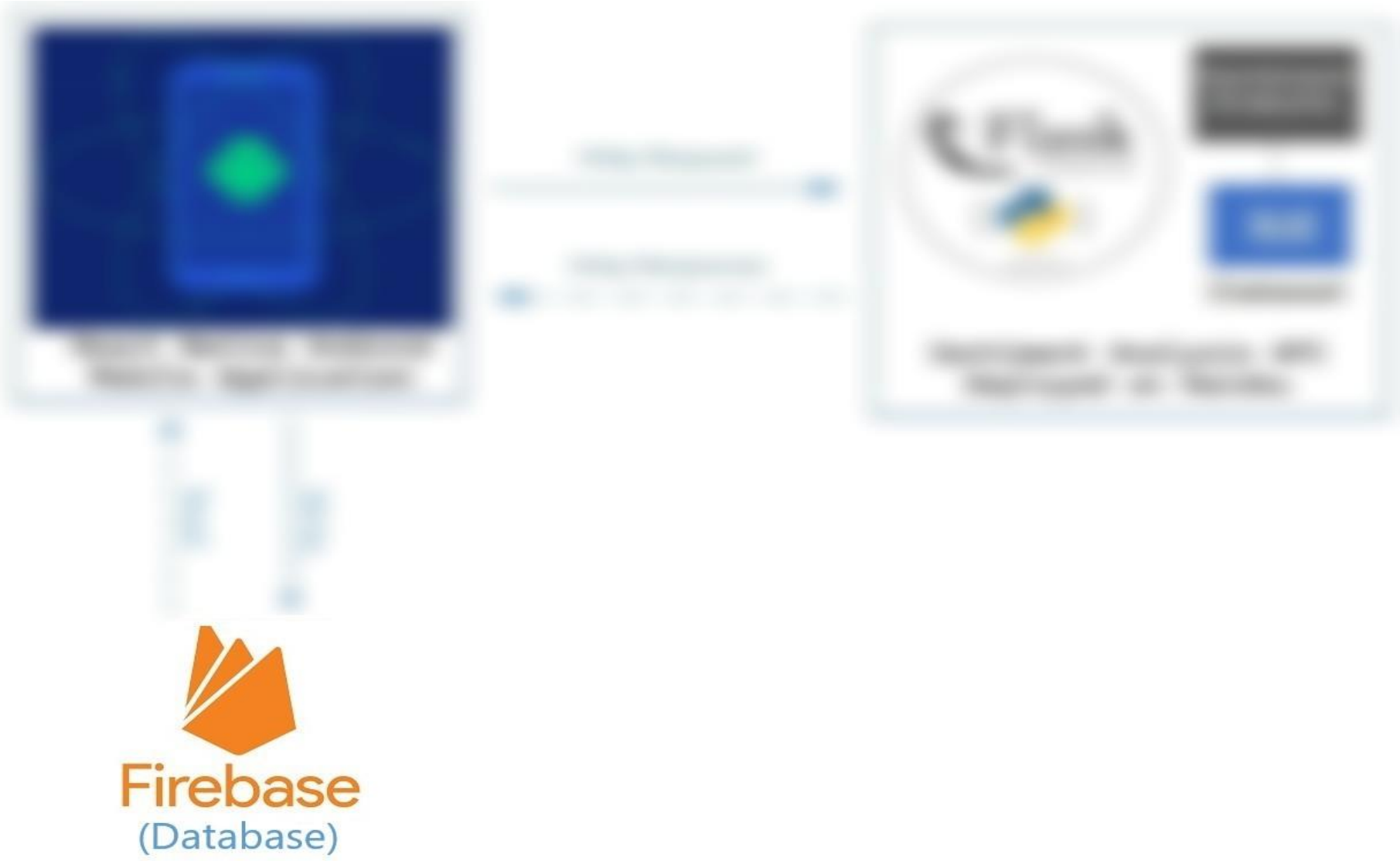
Incomplete Requirements

- The user should be able to see a list of houses for rent and details about them.

I couldn't finish this requirement.

Use Case Diagram

Database



Database Design

We have stored our data in the firebase database and we created relations between data. In following image, we can see relations between tables in the database.

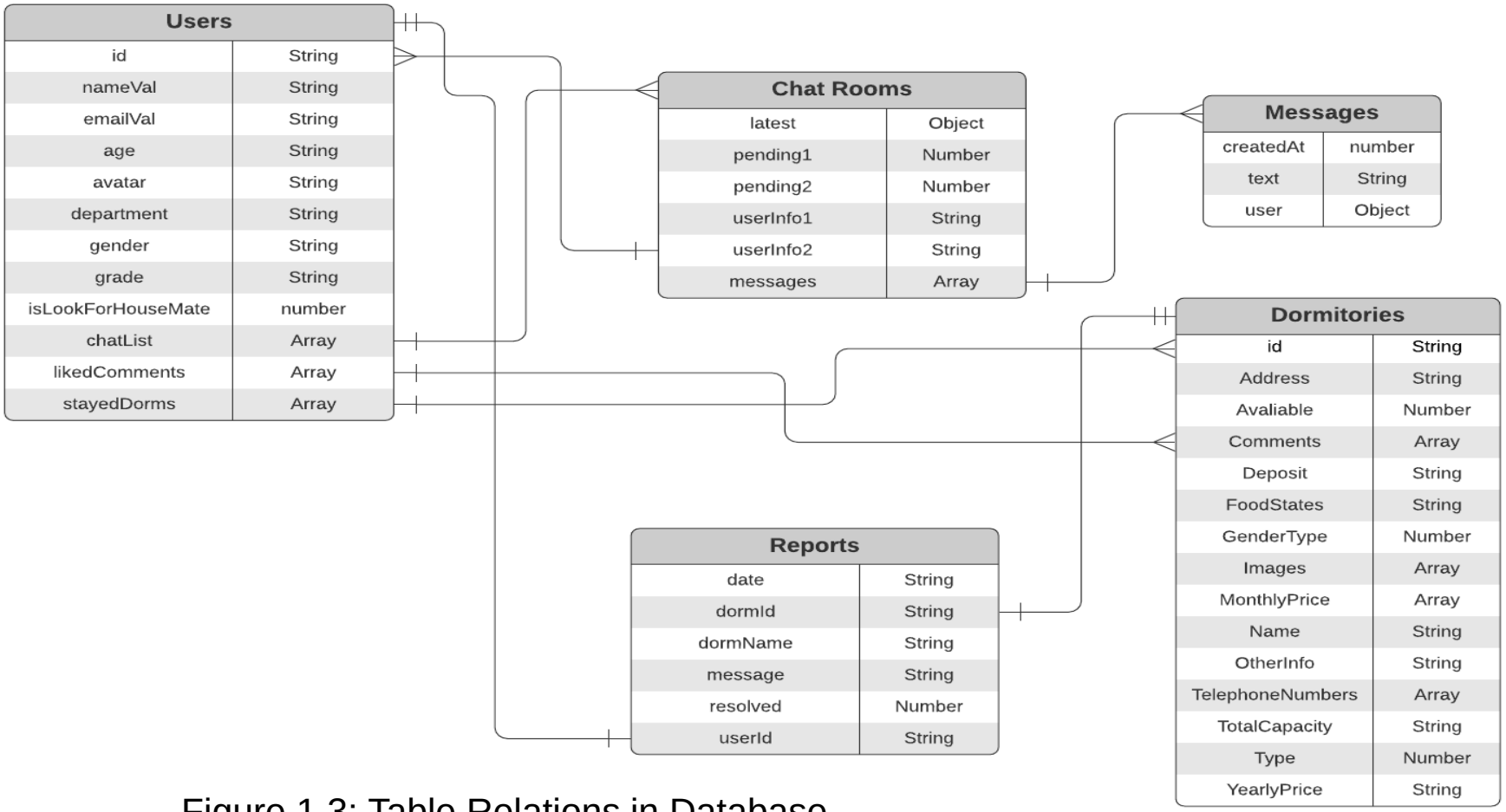


Figure 1.3: Table Relations in Database

Database

In following images, we can see the real tables and data in firebase.

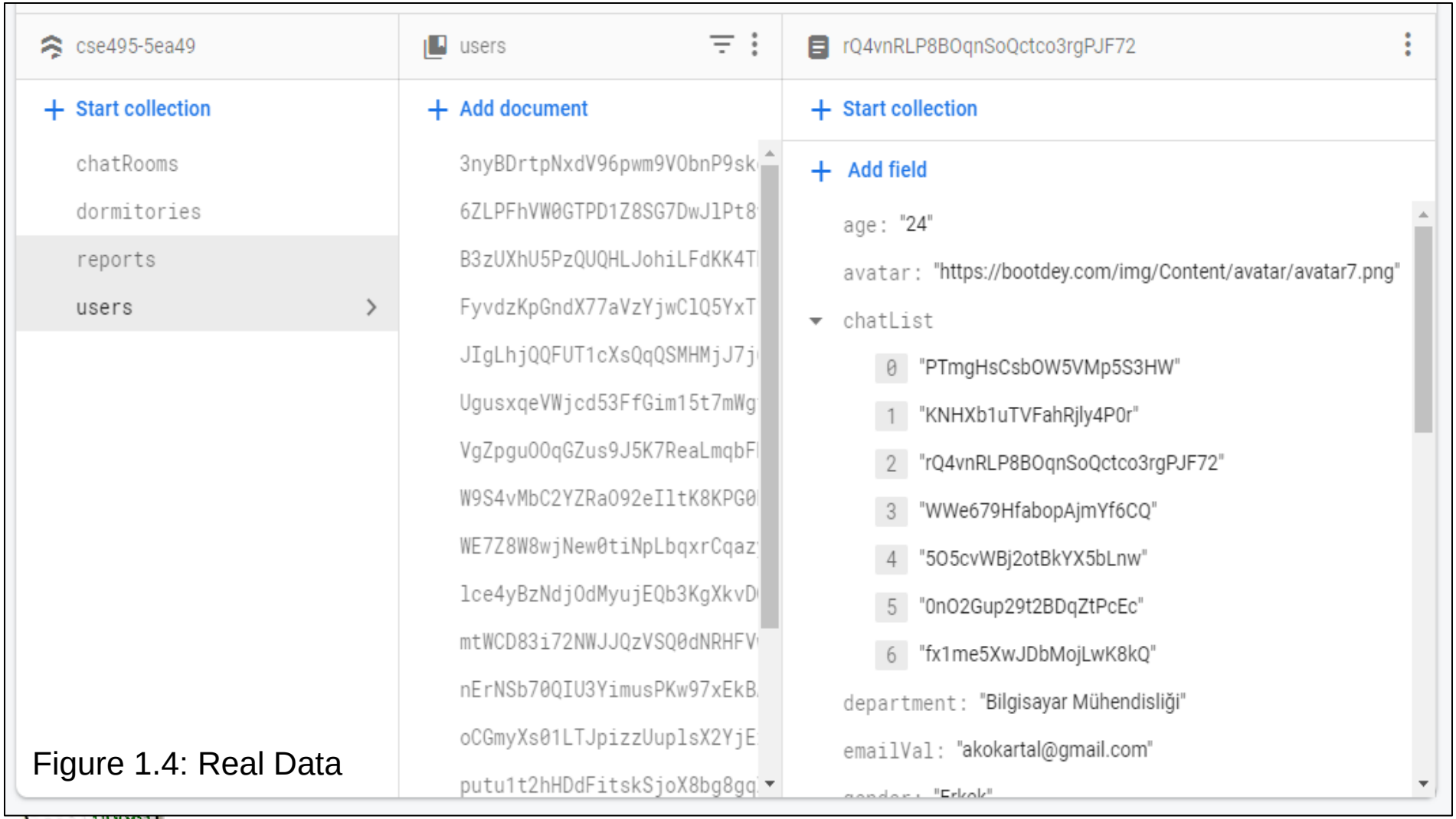
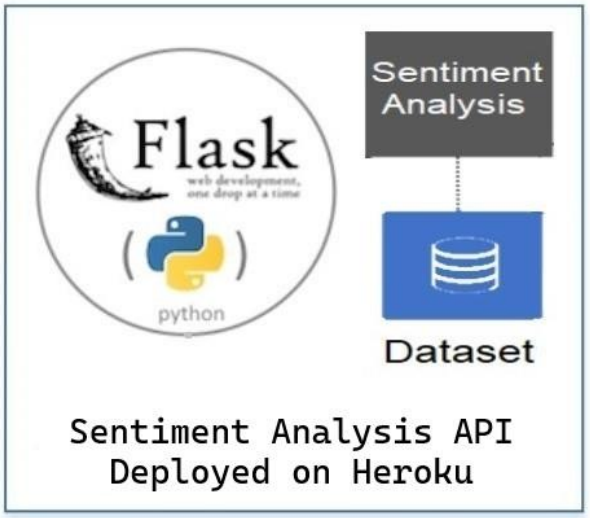


Figure 1.4: Real Data

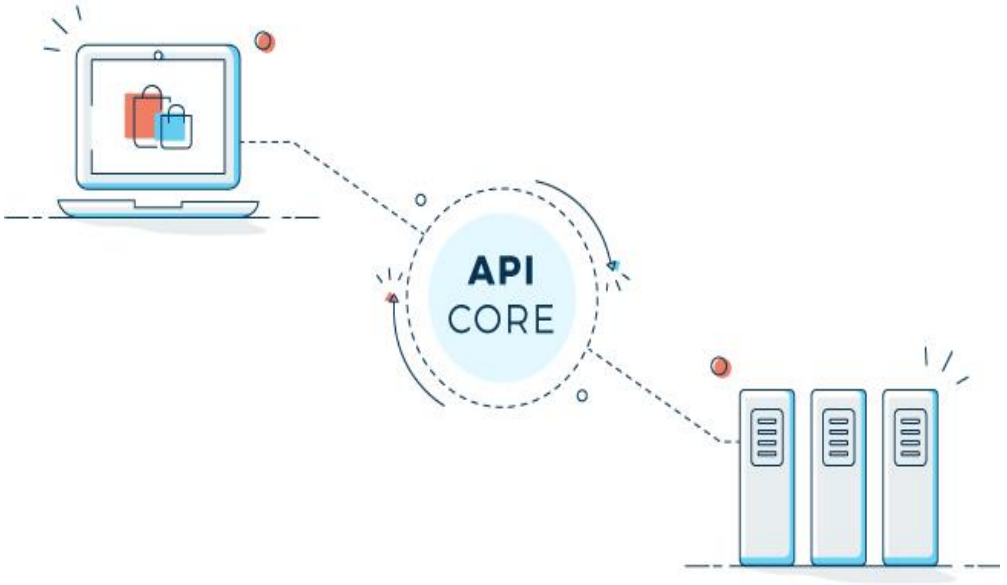
Sentiment Analysis



Sentiment Analysis

In order to finish sentiment analysis following steps were completed.

- Finding a dataset about comments.
- Training a simple model
- Exporting model to use in API.
- Creating an API with that model in order to use in mobile application
- Deploying that API so that everyone can reach sentiment analysis.



Sentiment Analysis

Dataset

- I have used the hotel comments dataset in English. In this dataset we have **38.932** hotel comments with response(label). Size of dataset is **35.7 MB.**

Detail | Compact | Column — 5 of 5 columns

User_ID	Description	Browser_U...	Device_Us...	Is_Response
id10326	The room was kind of clean but had a VERY strong smell of dogs. Generally below average but ok for a...	Edge	Mobile	not happy
id10327	I stayed at the Crown Plaza April -- - April --, ----. The staff was friendly and attentive. The ele...	Internet Explorer	Mobile	not happy
id10328	I booked this hotel through Hotwire at the lowest price I could find. When we got there the front de...	Mozilla	Tablet	not happy

Figure 2.1: Dataset Sample

Sentiment Analysis

Creating a model

- I created a simple model by using TF-IDF(term frequency–inverse document frequency) Vectorizer and the Classifier using Logistic Regression on Google Colab. Before training I cleared data and also I changed hotel words with dormitory word.

Paramaters

- In tfidf vectorizer, I only set the **ngram_range** to (1,1) it means it will only extract the **unigrams**. Other parameters are default.
- In Logistic Regression all parameters are default.

- Definition: A *n*-gram is a chunk of *n* consecutive words.
  - unigrams: "the", "students", "opened", "their"
  - bigrams: "the students", "students opened", "opened their"
  - trigrams: "the students opened", "students opened their"
  - 4-grams: "the students opened their"

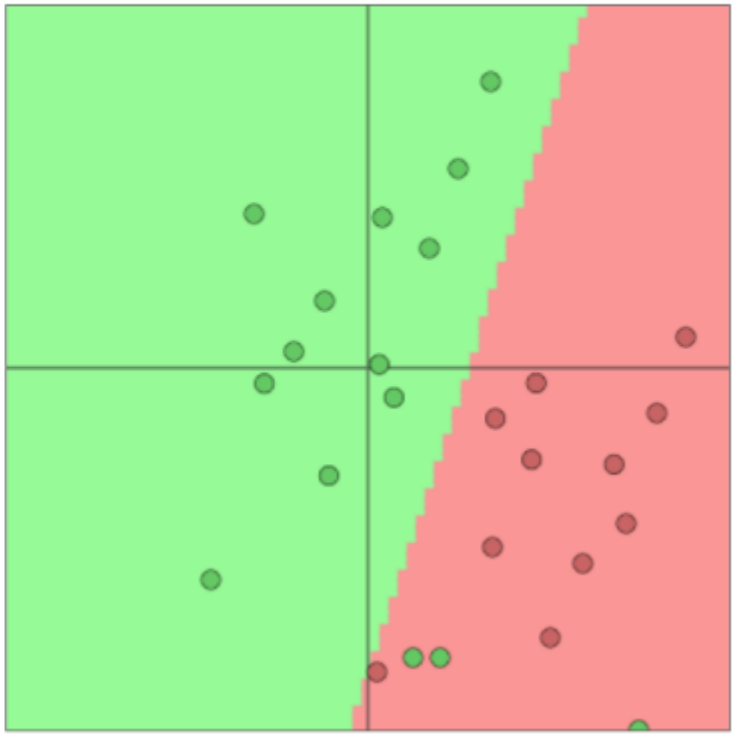
$$banking = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

* https://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture06-rnnlm.pdf

Classification

Classification intuition

- Training data: $\{x_i, y_i\}^N_{i=1}$
- Simple illustration case:
  - Fixed 2d word vectors to classify
  - Using logistic regression
  - → linear decision boundary →



*https://cs224d.stanford.edu/lectures/CS224d-Lecture4.pdf

Sentiment Analysis

Exporting and Deploying Model to the public

- In order to deploy API, Python Flask Github and Heroku was used.

```python
import pickle
from flask import Flask
from flask import jsonify
from flask import request
from googletrans import Translator

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
translator = Translator()


@app.route("/")
def index():
    return "Welcome to CSE495 API"
```

https://cse495api.herokuapp.com/

Figure 2.2: API Source Code

*Since Heroku is free it goes to sleep after a time if it is not used, and it takes about 20 second server to wake up but after that it takes at most 3 second to get result.

Creating Test Data

- In order to test application, I have created at least 100 comments ,at least 30 user and 10 dormitory data.

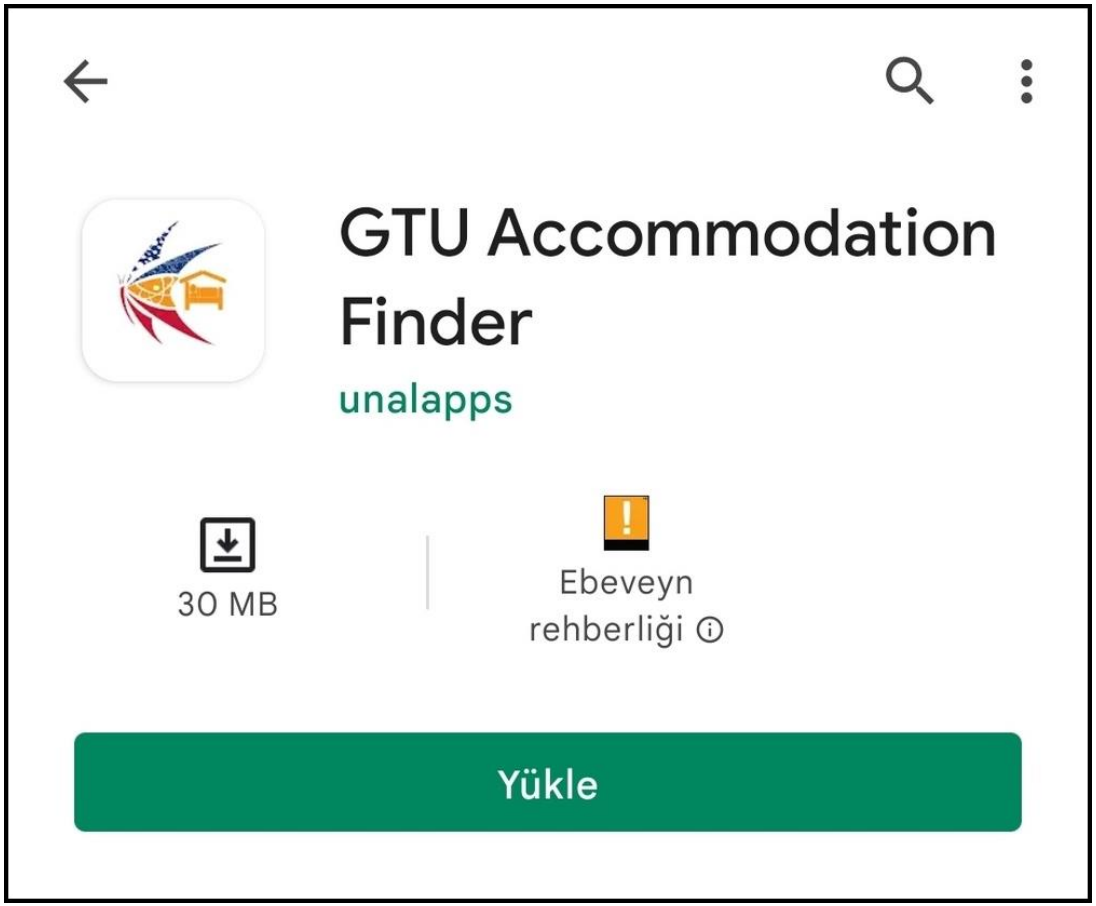
	Comment
2	bu yurt okula çok uzak
3	temizlik berbat
4	idare eder tavsiye ederim
5	fiyatlar çok pahalı
6	oteller burdan ucuzdur


```
age: "24"
avatar: "https://www.bootdey.com/img/Content/avatar/avatar2.png"
▸ chatList: ["PTmgHsCsbOW5VMp5S3HW", "...]
department: "Bilgisayar Mühendisliği"
emailVal: "aa@aa.com"
gender: "Erkek"
grade: "4"
id: "wmQLy0NyFXS3jgOb3pmF4pI7tcK2"
```

```
Address: "Muallim, Gebze Yurt Md., 41400 Gebze/Kocaeli"
Available: 0
▸ Comments: [{type: 1, date: "Sun Jan ...]
Deposit: 520 (number)
FemaleRoommateNumber: 4
FoodStates: "Sabah, Akşam"
GenderType: 3
▾ Images
```

Uploading App to Play Store

- In order to show my application size, I have uploaded my application to the Google Play Store. I have used my friend's developer account for this.



Success Criteria

1. At least 8 different screens in the mobile application.

   - As we have seen, I have **14** different screen in mobile application.

Success Criteria

2. Mobile Application size will be less than 100 MB.

- In order to show this, I have uploaded my application to the google play.

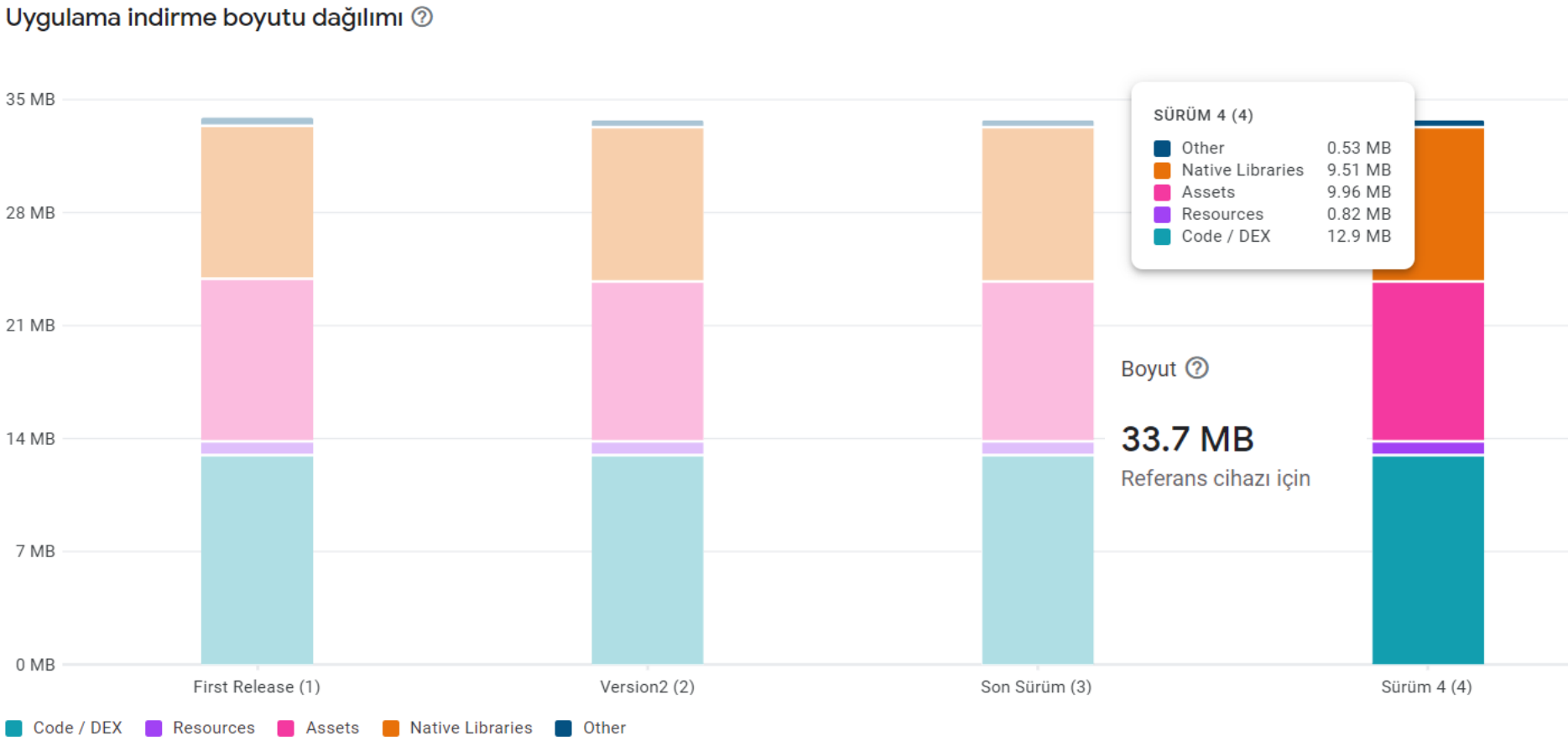


Figure 2.3: Mobile Application Size on Google Play Console

Success Criteria

3. At least %60 sentiment analysis accuracy.

➢ In order to test I have used 2 method. Confusion matrix and Turkish Sentences.

- Testing with Confusion Matrix

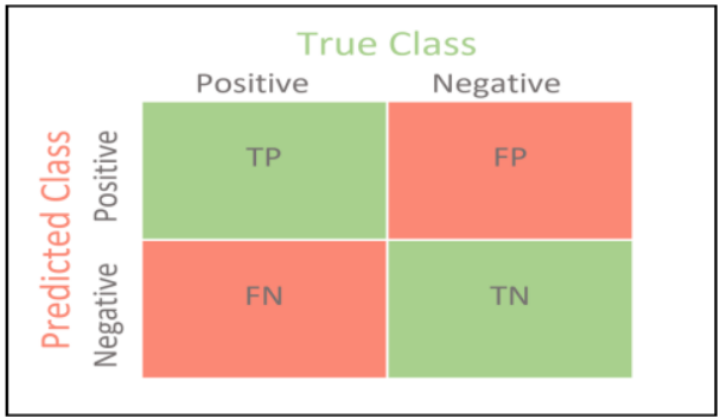


Figure 2.4: Confusion Matrix Representation

Success Accuracy and Error Rate formula in Confusion Matrix

Accuracy = TN + TP / Total Number

Error Rate = FN + FP / Total Number

```
from sklearn.metrics import confusion_matrix

verdict = model.predict(attribute_test)

confusion_matrix(verdict, target_test)
```

```
array([[1017,  154],
       [ 306, 2417]])
```

ing the result.

```
from sklearn.metrics import accuracy_score, p

print("Accuracy : ", accuracy_score(verdict,
```

```
Accuracy :  0.881869542886492
```

Figure 2.5: Code for Using Confusion Matrix

Success Criteria

3. At least %60 sentiment analysis accuracy.

- Testing with Turkish Sentences

I have created 100 Turkish Sentences. See the results we got **%76** accuracy.

Success Criteria

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

- In order to test I have used 2 method.
  - ✓ By measuring taken time in source code for database and Sentiment Analysis.
  - ✓ By using postman program for Sentiment Analysis API.

```
59      try {
60        var t0 = performance.now();
61        const response = await getResult(comment);
62        result = await response.json();
63        setSpinner(false);
64      } catch (error) {
65        alert("Bir hata oluştu. Lütfen tekrar deneyin.");
66      } finally {
67        var t1 = performance.now();
68        console.log("Sentiment Analysis Response Time: " + (t1 - t0) + " ms.");
69        setSpinner(false);
70      }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Sentiment Analysis Response Time: 2385.3181760013103 ms.
```

Success Criteria

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

```
49      var t0 = performance.now();
50      Firebase.auth()
51        .signInWithEmailAndPassword(email.value, password.value)
52        .then((userCredentials) => {
53          var t1 = performance.now();
54          console.log("Login time " + (t1 - t0) + " milliseconds.");
55          const user2 = userCredentials.user;
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Login time 1574.0831240005791 milliseconds.
```

```
71      var t0 = performance.now();
72      updateUser(info.id, info)
73        .then((docRef) => {
74          var t1 = performance.now();
75          console.log("Update Student Profile Time: " + (t1 - t0) + "ms.");
76          setLoad(false);
77        })
78        .catch((error) => {
79          alert("Bir hata oluştu. Lütfen tekrar deneyin.");
80        });
81      Alert.alert("Başarılı", "Bilgileriniz Güncellendi.", [
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Update Student Profile Time: 1203.9378639999777ms.
```

Conclusions

While making this project, I have experienced

- How to approach and create such a big project.
- How to divide a project into modules and conquer each of them separately so that problems will be small.
- Searching and learning new technologies within a short time.
- We can't avoid changes, we have to be prepared for this.

Contributions

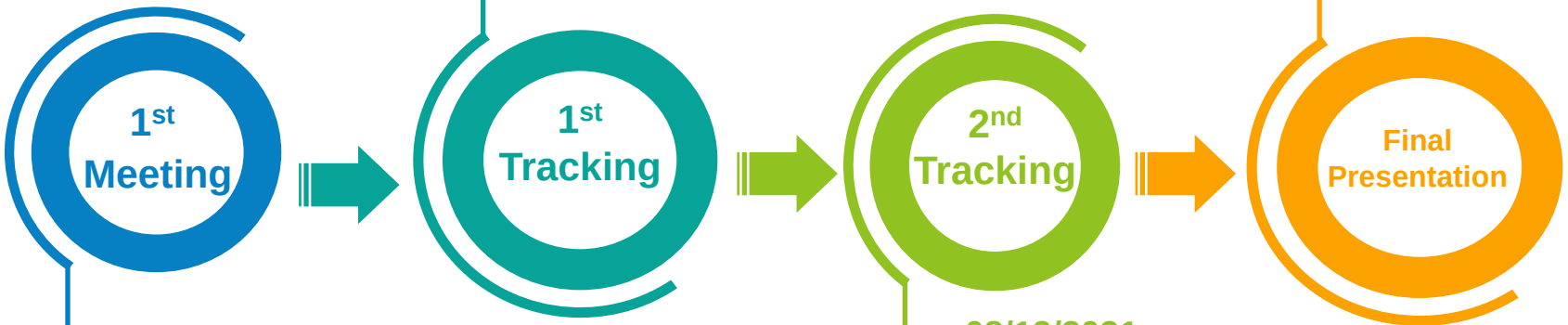
We made an end to end system that includes

- Mobile Application
- Database
- Machine Learning

Anyone with these needs can benefit from this project.

Also, this project is open to improvements so that it can turn into a startup easily and remove the next picture from the universities.

Timeline



21/10/2021

- Determine requirements and tools for project.
- Setup environments and tools.
- Design frontend pages.
- Start to collect data.

10/11/2021

- Continue to collect and update data.
- Create database and tables.
- Implement login and sign-up page, dormitory list page, and detail page. (roughly)

08/12/2021

- Implement a comment mechanism for the dormitories.
- Implement report feature for dormitory owners.
- Implement a list of students looking for housemates.
- Start to implement Sentiment Analysis for comments.

19/01/2022

- Finish sentiment analysis.
- Implement chat feature for students.
- Implement house for rent list page and detail page.
- Create test data for demo.

References

1. https://towardsdatascience.com/a-beginners-guide-to-sentiment-analysis-in-python-95e354ea84f6
2. https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/
3. https://www.kaggle.com/anu0012/hotel-review?select=train.csv
4. https://github.com/BaharYilmaz/turkce-duygu-analizi
5. https://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture06-rnnlm.pdf
6. https://cs224d.stanford.edu/lectures/CS224d-Lecture4.pdf
7. https://dergipark.org.tr/en/download/article-file/852974
8. https://github.com/RaihanAk/Hotel-Review-Sentiment-Analysis_MachineLearning