SUMMARY

Keywords: Image processing, Open Source Computer Vision, k-NN, Python, Flutter, Dart, Firebase, Mobile Application, Real-Time Cloud System, Cloud Computing, Multi-Platform Mobile Application

In my study, I have realized a useful project with a mobile application interface, which aims to be used for corporate or government offices, where entrance security is provided for car parks by identifying license plates on vehicles using image processing algorithms, and where teamworkers support business solidarity with real-time messaging with each other. In my project, I aim that the personnel in the companies mentioned above will be effective in using their time more efficiently, increasing the business solidarity with each other, strengthening the bond with the company and bringing the security of the companies to higher levels.

A real-time license plate recognition system has been developed with the Python programming language using the OpenCV (open source computer vision) library, which is based on the k-NN algorithm, for reading the vehicle license plates that the employees have recorded in the database for entry from the company door. While testing the license plate recognition system, experiments were performed on images taken from different positions and from different environments. In the project, using the Flutter and Dart software languages, the application and license plate recognition system, which will allow the company employees to communicate with each other via mobile, are stored in the Firebase Cloud Firestore database system, which is a cloud-based NoSQL service.

SECTION 1. INTRODUCTION

Today, increasing vehicle usage causes difficulties in traffic and control of entry and exit from one place to another. As a solution to this problem today, much work has been done in the literature on identifying plates and determining plate regions. License plate recognition system is the license plate identification technology for reading the license plates of vehicles using pattern recognition techniques. This technology is used in many security and traffic applications such as entry-exit controls and traffic system automations. Since all the vehicles in the traffic have a license plate, the process of identifying the vehicles without the need for extra transceivers makes the license plate identification technology very popular. The most notable advantage of the plate identification system is that it can store plate information.

In this study, a project has been developed by using license plate recognition system and mobile messaging application, which are not currently available in the literature, and will meet the needs of companies. The technologies used in this project are up-to-date and highly permissive to future developments. All transactions in the project are operated instantly. When the project is considered in general terms, it is designed as a company application that enables employees in an institution to communicate with each other and where the entrances are controlled by the license plate recognition at the parking lot entrance of the company. The Python programming language, which has become very popular today, and a license plate recognition system based on the OpenCV library, K-Nearest Neighbor algorithm, which is very convenient for pattern recognition, has been developed. The project uses multi-platform Flutter UI SDK and Dart programming language, which is open source, Android, iOS, Google Fuchsia operating systems developed by Google, to allow company employees to communicate with each other via mobile. All messaging and user information are safely stored in the Firebase database, authorization, storage and cloud messaging system.

SECTION 2. ALGORITHMS

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV, a BSD licensed product, was established to provide a common infrastructure for computer vision applications.

The library has over 2500 optimized algorithms, covering both classical and modern computer vision and machine learning algorithms. These algorithms are used in applications such as detecting and recognizing faces, identifying objects, and defining characters. OpenCV-Python is a Python connector library designed to solve computer vision problems. OpenCV-Python uses the Numpy library, a highly optimized library for numerical operations with a MATLAB-style syntax. All OpenCV array structures are converted to Numpy sequences [10].

2.1 Image Classification

Image classification is the assignment of pixels to the respective classes. Thus, homogeneous pixel groups are created [22].

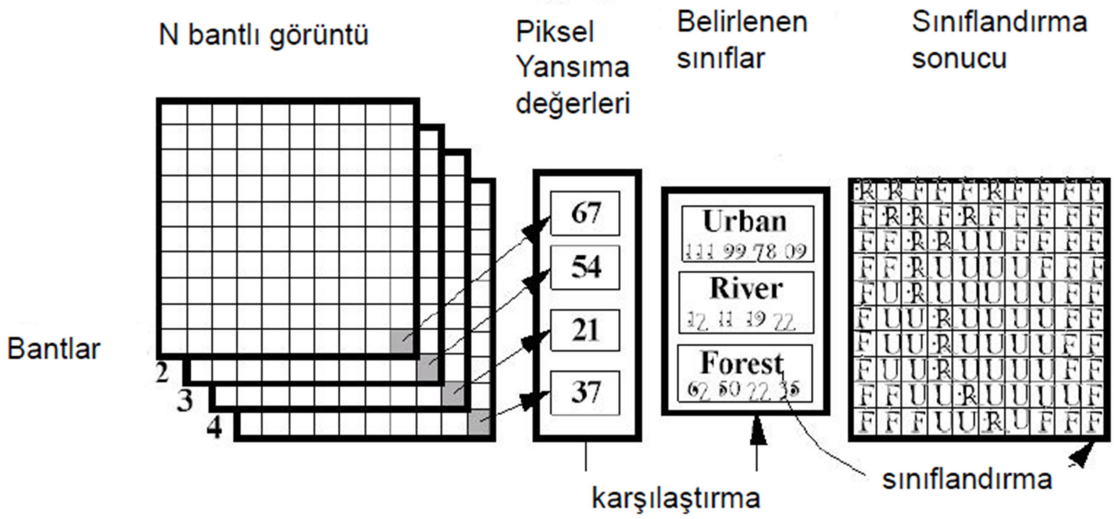


Figure 2.1. Homogeneous Pixel Groups [22]

Classification is divided into two main groups as controlled and uncontrolled classification;

• Classes are known to the user in controlled classification.

• The pixels are grouped depending on the number of algorithms and classes selected in the uncontrolled classification.

• Hybrid classifiers use both controlled and uncontrolled classification methods. [22]

2.1.1 Uncontrolled classification

Remote sensing images often contain different spectral classes. The uncontrolled classification as a premise is the identification, marking, etc. of the classes in question. used for transactions. Preliminary information on the image is not available. Each class is collected in a different set. Pixels are spectrally separated. The user determines the number of classes, the number of iterations, the convergence threshold.

Most commonly, Isodata and K-Means Algorithms are used.

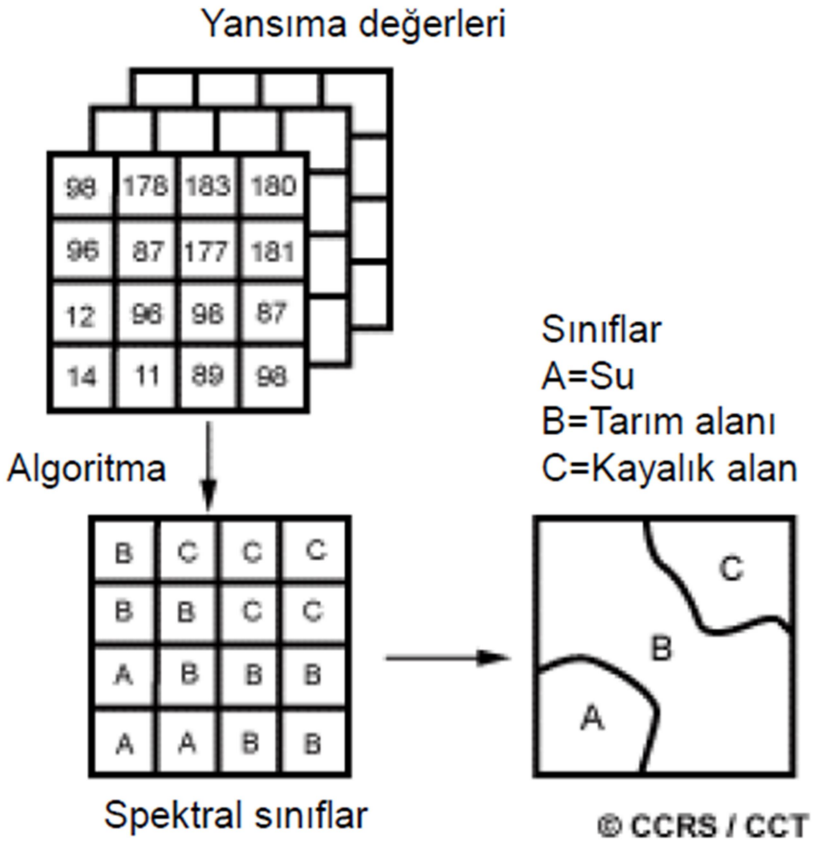


Figure 2.2. Uncontrolled Classification [22]

Weaknesses;

• Classes may not really match their counterparts in nature.

• Classes and their features cannot be controlled.

• The spectral properties of classes may change over time. [22]

2.1.2 Controlled classification

Classes are known to the user in controlled classification. Preliminary knowledge of classes is required. Classes are determined. The control area is selected for each class. Ground reality is used for testing the control areas. The analyst can check the classification. There is information about each class. With the help of audit fields, wrong classification results can be audited. [22]

Controlled learning is divided into two broad categories;

• Classification for answers of several known values, such as 'true' or 'false'. Classification algorithms are applied to nominal response values, not sequential.

• Regression for answers with real numbers such as miles per gallon for a particular car. [22]

Important characteristics of the control areas;

Pixel count; More than one control area can be measured for a class. The sample area in the control area should be in a number that can represent the class statistically.

• It depends on the number of classes, variety and resource. [22]

There are methods such as parallelepiped, Maximum likelihood, K-NN (K - nearest neighbor) algorithm. [22]

Weaknesses;

• The analyst determines the classification.

• Control areas are generally measured based on information, not spectral properties.

• Representation areas of control areas may be inadequate.

• Selection of control areas can be difficult and time consuming. [22]

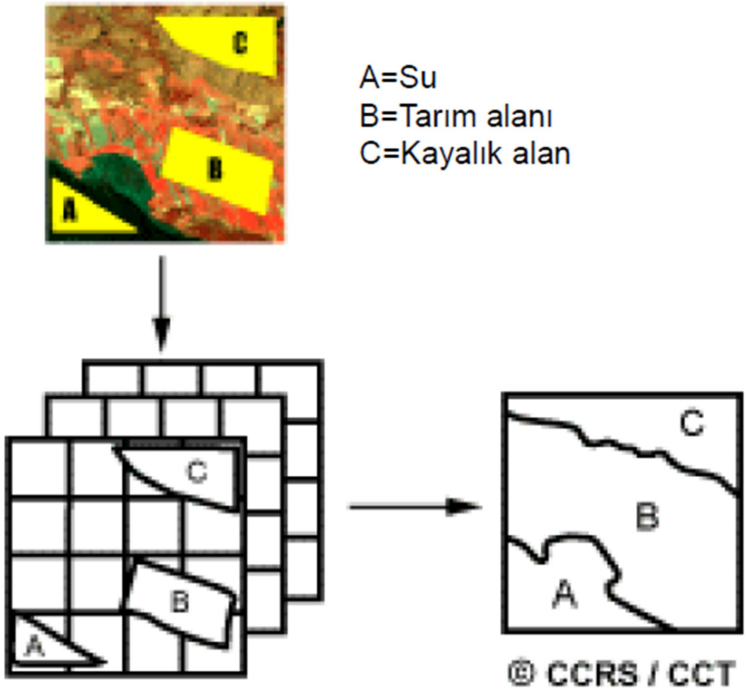
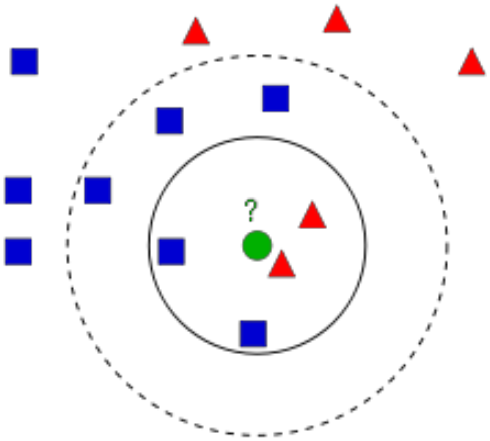


Figure 2.3. Controlled Classification [22]

2.2 K-NN (K - Nearest Neighbor) Algorithm

K-NN is one of the simplest classification algorithms for Controlled learning. The main idea of K-NN is to search for the closest match of test data in the feature field [1].



Şekil 2.4. K-NN Algoritması [1]

The picture above shows the red triangles and blue squares that belong to the two classes. Classes are shown on a city map called Makan (Feature Space). We can think of a space as an area where all the data is reflected. For example, consider a 2D coordinate space. Each data has two properties, x and y coordinates. We can show this data in your 2D coordinate space. Now consider the ones with N properties, since you will need N dimensional space, this N-dimensional area is its property area. In our picture, we can think of it as a 2D case with two features [2].

In the picture above, a new member comes to the city map as a green circle and creates a new space. This green space should be added to one of the blue and red spaces. This process is called Classification. Since we are working with K-NN, let's apply this algorithm [3].

There is a method that checks who is the closest neighbor to this green space. From the image, it is clear that the Red triangle family. So the green circle is added to the red Triangle. This method is simply called the Nearest Neighbor because the classification depends only on the nearest neighbor. It may be the closest to the red triangle. But what if there are a lot of blue squares next to it? Then the blue squares have more power in that area than the red triangle. So it's not enough just to check the closest. Instead, we check the closest places [4].

Let's take K = 3, that is, the 3 closest places in our picture. Two red and one blue (In the picture above, there are two equidistant blue squares. But due to K = 3, only one

blue square is taken.) Since it is near the green circle, red triangles are placed in its space. If we take K = 7, we will have 5 blue squares and 2 red triangles under distance and majority control. As shown in the picture, although the red triangles are close to the green circle, since the majority shows blue squares, the green circle space is added to the blue squares drum. If K = 4, there are 2 red triangles and 2 blue squares as shown in the picture. However, since it is the red triangle closest to the green circle, it joins the space of the red triangles [5].

In order to find the nearest neighbor, it is necessary to have information about existing spaces and new arrivals. If there are many places, a lot of memory and computing power are needed. A deficiency of the K-NN algorithm is that it is sensitive to the local structure of the data. The algorithm should not be confused with another popular machine learning technique, K-Means.

It requires three situations:
• Image
• A distance criterion for calculating the distance between classes
• K is the closest neighbor available

To classify an unknown pixel;
• Distances to other control areas are calculated.
• K is determined.
• Classes of the nearest neighbors are determined to assign the unknown pixel to a class.

The distance between two points;
Euclidean distance d (p, q) = $\sqrt{\Sigma}$ (pi – qi) 2
Hamming distance (overlap metric)

Determining the class from the nearest neighbor list;
• Most repeated class in K-nearest neighbors
Weight factor w = 1 / d2

SECTION 3. CHARACTER RECOGNITION STEPS AND APPLICATIONS

3.1. Plaka Okunması

3.1. Reading the Plate

In an OPT, the license plate reading was realized after 4 basic operations such as reading the license plates of the vehicles with image processing, determining the license plate, removing the license plate number, segmentation of the characters and making the character prediction in the license plate.

3.1.1 Plate Placement

In the view taken from the camera used for this system, the plate location can be determined using the edge detection technique. For slope detection, the starting and ending points of the characters are determined according to the x axis. There are upper boundary and lower boundary points relative to the Y axis. The height of the characters taken on the image is equal. If the upper limits of the detected characters increase or decrease regularly from left to right, there is a slope on the plate. This regular increase or decrease value is determined, the slope angle is found by calculating the differences on the x-axis and y-axis.

After the characters start and end points are found on the inclined plate, it is necessary to find the rate of increase or decrease of the upper and lower limits of these characters relative to each other. The license plate frame, located above and below the characters, sometimes causes problems in determining its boundaries. If the difference of adjacent pixel values used in the plate location learning method is over 20, these pixels are considered as the edge region. In this way, the problem is eliminated since the plate frame is created [6].

3.1.2 Removing the Plate Area

After that, we need to remove the plate area from the landscape or image. Plate region extraction is first converted into a binary system by detecting the image captured from the camera by 1 (white) for all pixels in the input image and the threshold values of 0 (black) pixel values for all pixels at the entrance of the image with a brightness less than the threshold value [7].

3.1.3. Plate Character Partitioning

In the segmentation of the characters in the plate, the characters are divided into its constituent parts, which are obtained separately. First, the image is filtered to improve the image and remove unwanted points. If the characters are close together, then the image is expanded to separate them. After doing the operations here, we need the splitting options, which are the enemies of the characters. We either use the regionprops () function to get the bounding box of each character, and then we crop it, or we just take and trim the highest row and highest column of each character. Each clipped character is then resized and stored in the line matrix, respectively. These datasets are used as the input of a trained neural network to test matching characters [8].

3.1.4 Optical Character Recognition (OCT)

The last step after partitioning is character recognition. For this step, the output of the partitioning process is used as input. It means that the segmented characters are fed

to the neural network of the output matrix and do some processing on the neural network and give the results in text.

Characters are normalized before the recognition algorithm. Normalization is fitting characters in a block that does not contain extra white space (pixels) on all four sides of the characters. To match the characters to the database, it must be the same size as the input images. Database characters are suitable for 20x20. The characters cut from the plate and in the database are thus brought in equal size.

Due to the similarities of some characters, there may be some errors during recognition. Mixed characters are basically B and 8, E and F, D and O, S and 5, Z and 2. In order to increase the recognition rate of the characters, some criteria tests are used in the system. Thanks to these characteristics and applied tests during the recognition algorithm, the recognition rate is increased with minimum error [11], [12], [13].

The organization of a computer vision system is highly application-dependent. Some systems are independent applications that solve a specific measurement or detection problem. Many functions are application-specific. However, there are typical functions found in many computer vision systems.

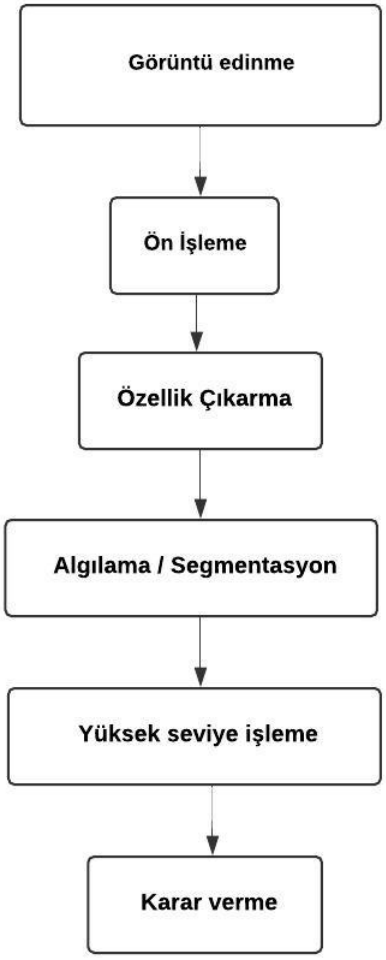


Figure 3.0. Character Recognition Flow Chart

3.2. Image acquisition

A digital image is produced by one or more image sensors, i.e. photosensitive cameras. Pixel values typically correspond to light intensity in one or more spectral bands (gray images or color images), but may also be associated with various physical measures such as the depth or reflection of sonic or electromagnetic waves.

Figure 3.1. Color Image and Plate

3.3. Pre-processing

Before a computer vision method is applied to the image data to extract a particular piece of information, it is usually necessary to process the data to ensure that the following methods fulfill certain assumptions implied.

Noise reduction to ensure sensor noise is not misleading.

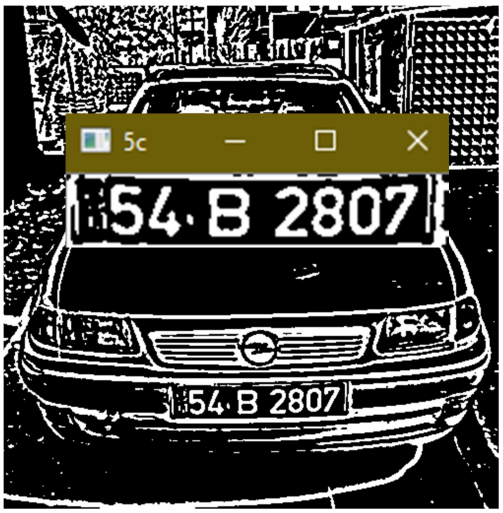
• Enhance contrast to ensure perception of relevant information.

• Resampling to make sure the image coordinate system is correct.

• Scaling area display to improve image structures at locally appropriate scales.

From the previous image, to reduce noise, Grayscale (Figure-3.2) and also the image is converted to the threshold image (Figure-3.3 and Figure-3.4).

Şekil 3.2. Gri Görüntü ve Plaka



Şekil 3.3. Eşik Görüntüsü ve Plaka

Şekil 3.4. Eşik Kenar Görüntüsü ve Plaka

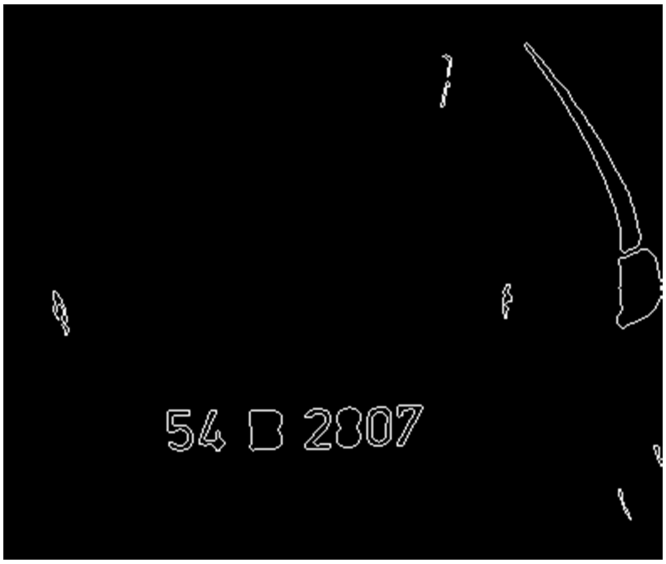
3.4. Feature Extraction

Görüntü karmaşıklığındaki çeşitli karmaşıklık seviyelerinde görüntü özellikleri elde edilir.

Bu özelliklerin tipik örnekleri:

- Çizgiler, kenarlar ve sırtlar.
- Köşeler, noktalar vb. yerelleştirilmiş ilgi noktaları.
- Daha karmaşık özellikler doku, şekil veya hareketle ilgili olabilir.

Eşik görüntüden, olası karakterleri tanımlamak için çizgiler, kenarlar, sırtlar vb. özellikler birbirinden ayrılır (Şekil-3.5).



Şekil 3.5. Feature Extraction

3.5. Detection / Segmentation

At some point in the process, it is decided which image points or regions of the image are suitable for later processing. For example;

• Selection of a particular interest group.

• Partitioning of one or more image regions containing the object of interest.

In Figure 3.6, possible characters are matched to trained characters.

Şekil 3.6. Segmentation

3.6. High level processing

At this stage, the input is typically a small dataset, a series of points, or an image region, which is supposed to contain a particular object. The remainder deals with processing.

• Verification of data based on model-based and application-specific assumptions.

• Estimation of application-specific parameters such as object exposure or object size.

• Image recognition - classify a detected object into different categories.

• Image recording - comparing and merging two different images of the same object.

In Figure 3.7, all possible plates are identified and treated as rectangles, rectangles without correct characters are discarded.

Şekil 3.7. Possible Plate on the Stage

3.7. To decide

The final decision required for the following applications has been made. These are as follows;

• Successful / unsuccessful in auto review.

• There is no match / match in recognition.

The last rectangle with the appropriate characters from the previous stage (Figure-3.7) is matched with the trained characters and printed on the original image (as shown in Figure 3.8).

Figure 3.8. Plate Numbers on Original Image

Figure 3.9. Cutting Fields of Characters

SECTION 4. MOBILE APPLICATION PLATFORM

The project was developed using multi-platform Flutter and Dart software languages developed by Google, which enables applications to be developed for Android, iOS and Google Fuchsia operating systems.

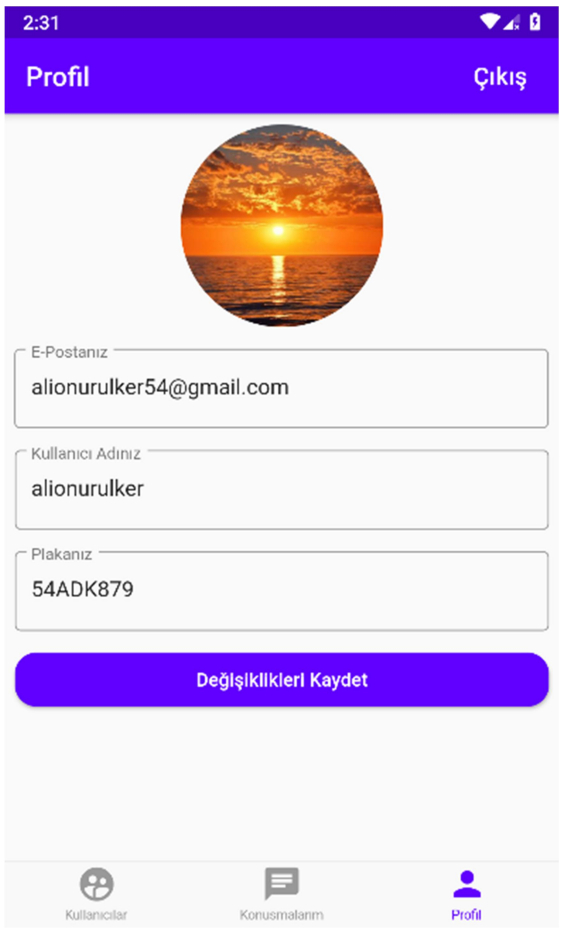
4.1. Technology Definitions and Usage Scenario

Flutter is a rich SDK with Mobile-first 2D Skia rendering engine, react-style framework, plug-in support used in Android and IOS platform, which can be run in Windows, Linux or Mac environments with its consistent and flexible use. Unlike Javascript, Flutter uses Dart, a compiled language. Dart is a language that can be compiled both JIT and AOT. Dart JIT is compiled when developing a Flutter application. Material Design and Cupertino (iOS style) widgets allow developing effective applications for both platforms with the help of themes. It has a real-time editing feature called Hot Reload, which is not available in many development environments. With Hot Reload, developers can instantly see the reflections of their changes in the application while they are preparing or editing their applications. With the performance of Flutter, it is now noticeable that we do not notice and not see what we call animation in mobile applications. Lists, switches, checkboxes, page transitions, dialogs, gesture effects etc. even in the most common application. in fact, everywhere is filled with animation. Thus, high and stable FPS becomes critical for a mobile application. Ease of customization In Flutter, the interface layer is not dependent on native SDKs, nor are some abstractions, interfaces, bridges etc. built on them. dependent. Any user interface that can come to our mind from a native, hardware level, single codebase with a single language is possible. Moreover, Flutter offers a very rich catalog of ready-made interface components and is of course open source and a Go To Definition distance from our editor [14] [15] [19].

Firebase Cloud Firestore; According to the announcement from its official blog, Firebase, the cloud-based service provider purchased by Google in 2014, Google is a NoSQL service developed as an alternative to the Realtime Database service, which Cloud Platform launched in beta last year [18]. It is a document-based NoSQL

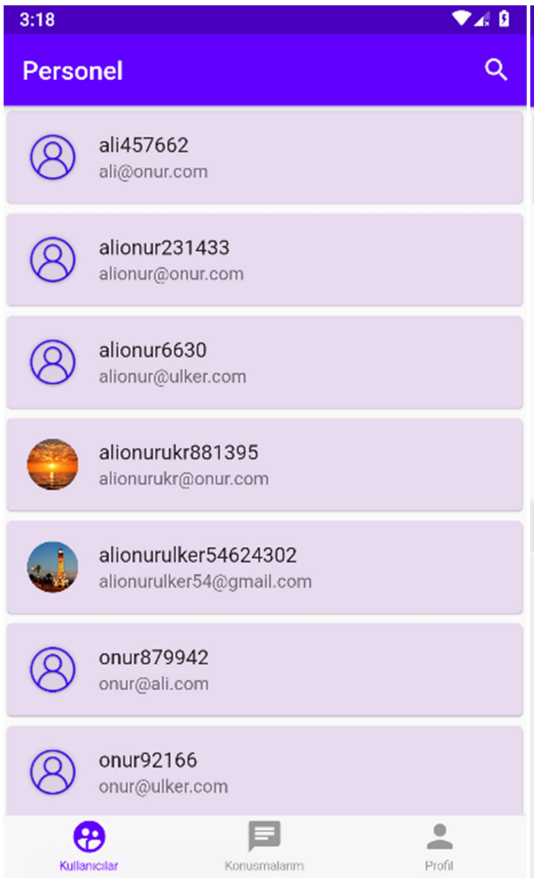
database. The data is stored in structures called documents. The document community is called collections. When storing and querying Nested data, Cloud Firestore provides a more performance, scalable and flexible structure. While there is a Realtime Database for storing plain data, it is seen that Cloud Firestore is also successful in plain data and offers advanced features. Thanks to the real-time update feature (realtime listeners), when the database changes, it automatically synchronizes, allowing the data to be updated automatically in the mobile application or on the website. With the support of offline mode, data can be read and written from the device's local when there is no internet. When the internet comes, it can automatically synchronize to the cloud database. It is fully compatible with Firebase Authentication. Thus, access or restriction on the data can be provided according to the user's authority. You can easily store complex, hierarchical, nested data and make quick queries. Each document can contain different numbers and names, since you do not have to store the data according to a certain scheme. Rich platform support (Web, IOS, Android, Java, Node.js, Php,
Go, Python ..) Data is stored as Key-Value. Each document must have an ID. Whether you give the ID or Firestore will create it automatically. [16] [19]

Şekil 4.1. Kullanıcı Kayıt / Oturum Açma

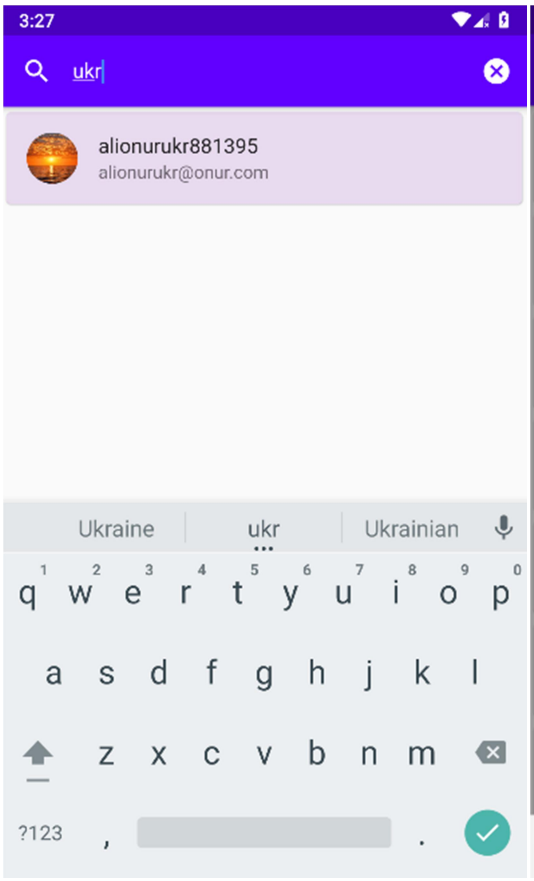


Şekil 4.2. Profil Sayfası

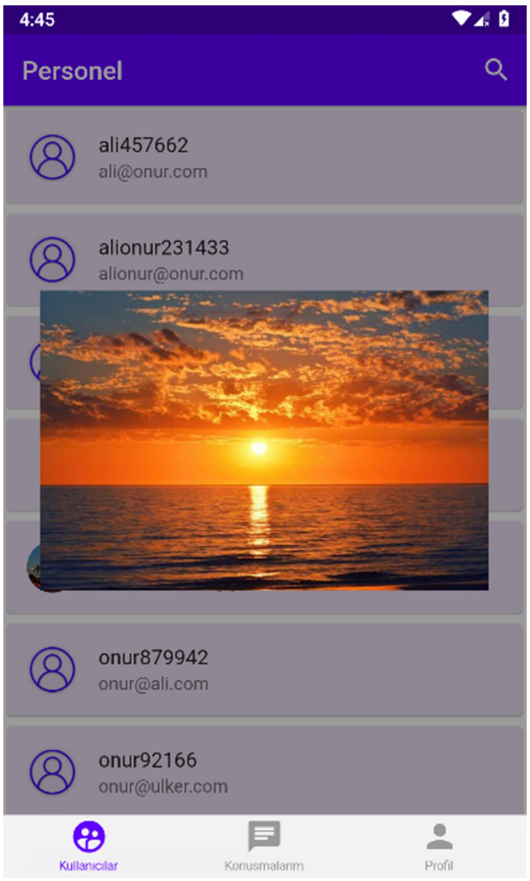
As can be seen in Figure 4.1, the user can login to the application with his own Gmail or Facebook accounts or by normally registering his e-mail and password to the system. In this login screen, the necessary validation commands have been applied to enable the user to log in the correct format. This screen is enriched with opening animations to appeal to the user. The user is able to update his / her own user name and profile photo from the Profile page shown in Figure 4.2, and the security of the license plate recognition system has been created by considering the license plate information of the vehicle of the user from the same page and by entering the company from the very base of this license plate information while entering the company. Unless the Save changes button is pressed, updates and downloads are not recorded in the database. The user can save the profile photo to the application instantly either from his camera or from his gallery.



Şekil 4.3 Personel Sayfası



Şekil 4.4. Personel Sayfasında Arama Yapma



Şekil 4.5. Kullanıcının Profil Fotoğraf Gösterimi

A list of all users is created on the personnel page shown in Figure 4.3. All the information is taken from the database by parallel programming (multi threding) process. As you go down to the bottom of the page in the personnel page, the information is pulled from the database 10 times. In other words, pagination infrastructure is available in practice. As seen in Figure 4.4, users can search by e-mail or username on the personnel page of the application. While searching, all users on the list are displayed on the staff page. If the letters entered in the search are in the email or user name of the users, they are on the list. If there is no user name or e-mail address consisting of these letters, these users are removed from this list and prevented from being seen in the list. As seen in Figure 4.5., Detailed profile photos of the users can be viewed using the structure called Hero widget.

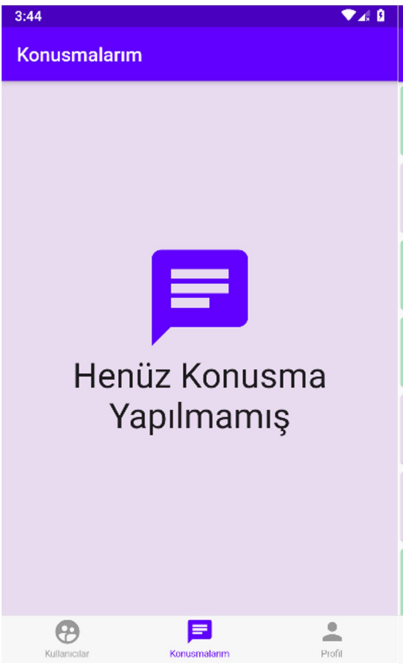


Figure 4.6. Display of My Conversations Page Empty

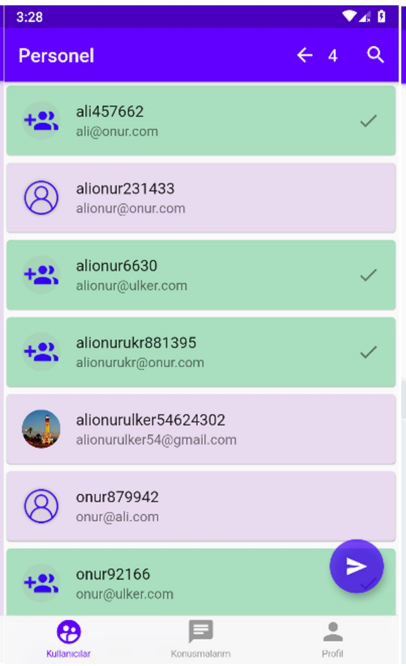


Figure 4.7. Multiple Selection on Staff Page

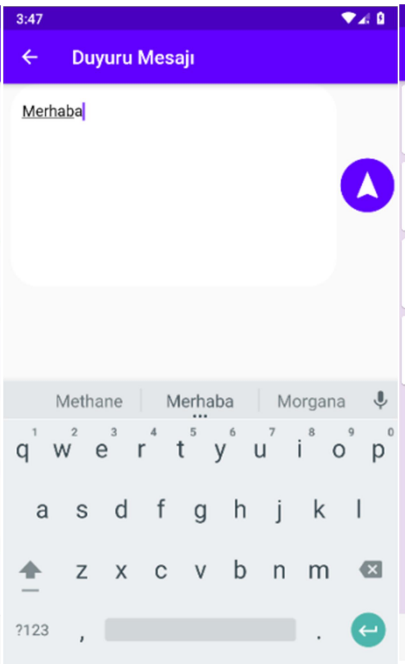


Figure 4.8. Announcement Message Page

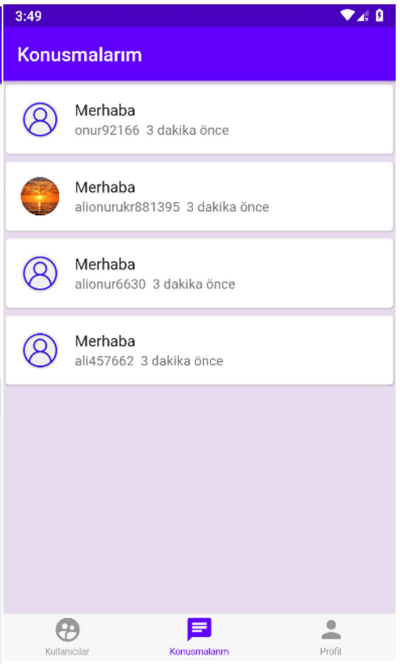


Figure 4.9. Screening of the My Speaking Page When Mass Speaking

If any user is clicked once, routing is done by sending the parameters of the current user and the chat user to the Chat page. It is possible to make multiple selections in the application. If any user is selected by long click, as shown in Figure 4.7, the number of selections in the upper part called appbar and next to it, the selection selection button and also the action button in the lower right corner will appear on the phone screen. If the selected user is long clicked again, the selection of that user will be canceled. If the selection is made and the action button is clicked, all selected users are directed to the Announcement Message page. Mass messages can be sent to the users selected in the Personnel page from the Announcement Message page, seen in Figure 4.8. Messages sent to any user or to users selected collectively are listed on the My Conversations page shown in Figure 4.9.

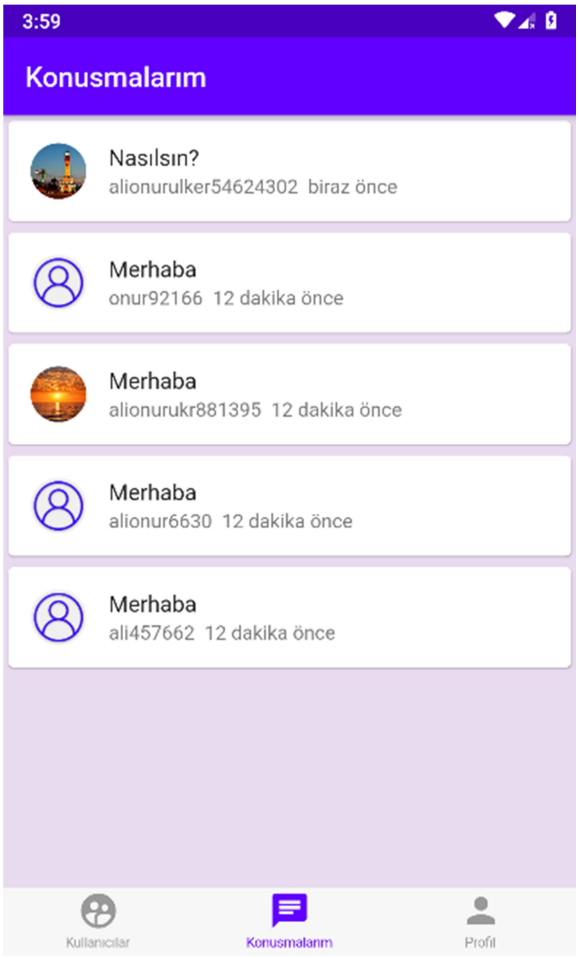


Figure 4.10. Screening of the My Talk Page When Speaking

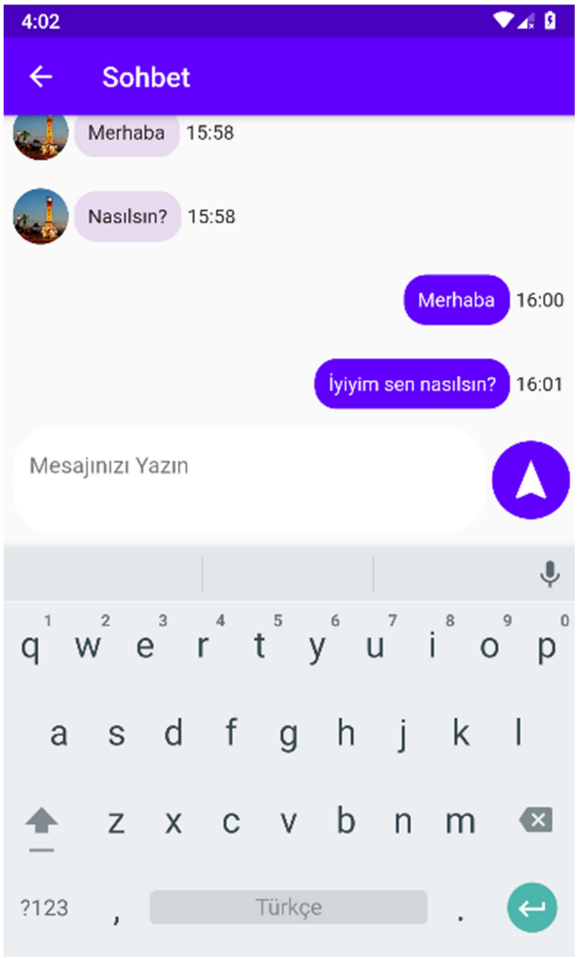


Figure 4.11. Chat Page

Pagination structure is used in the My Conversations and Chat pages as in the Staff page. The chat flows from the bottom up. When the user comes to the Chat page, he / she can view the previous messages by 10 and 10 by pulling the messages from the database, not all of them. As seen in Figure 4.11, if a message is received or sent, this message and the timing in the hour and minute format are displayed at the bottom of the page. As seen in Figure 4.10, the list flow is from top to bottom on the My Conversations page. If someone sends a message to the user or sends a message to the user, the message will be displayed at the top of the list on this page, with the username and how long this message was posted..

4.2. Data security

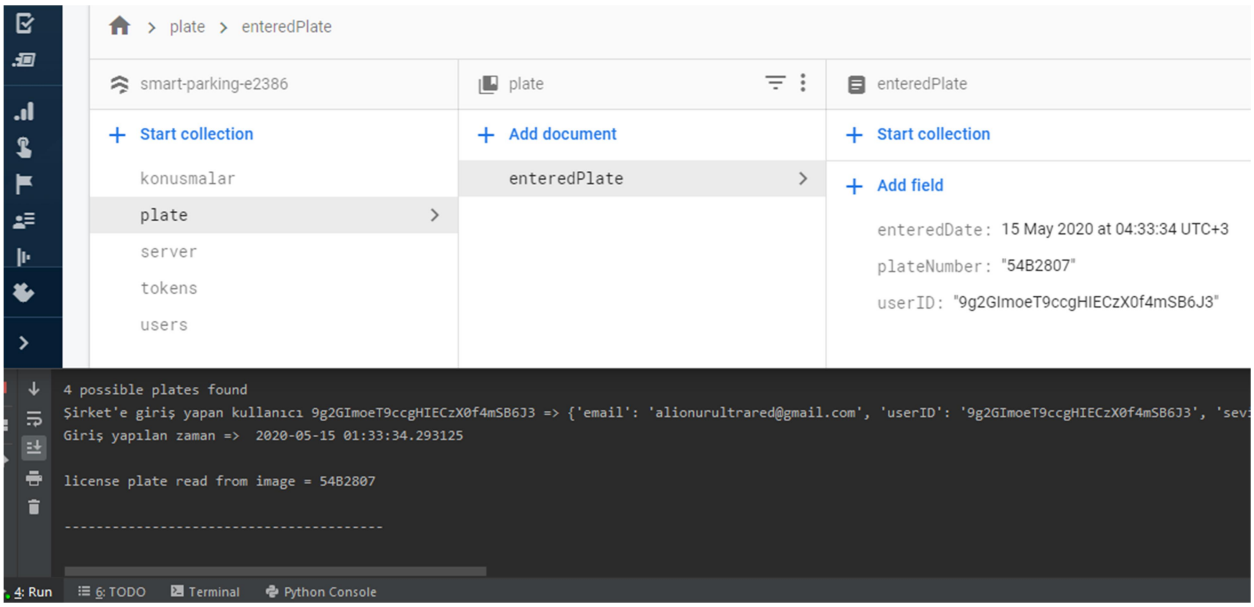


Figure 4.12. Firebase Database and Python Output Screen

Profil sayfasında kullanıcı kendine ait aracının plaka bilgisini kaydederek şirketine giriş esnasında bu plaka bilgisinin veri tabanından kontrolü ile şirketin otoparkına girişini sağlayabileceği düşünülerek plaka tanıma sisteminin güvenliği oluşturulmuştur. Şekil 4.12.'de görüldüğü gibi giriş yapan kullanıcının plaka bilgisi ve giriş yapılan tarih veri tabanına aktarılmaktadır.

Firestore'da veri tabanını ilk oluşturduğunuzda, varsayılan olarak erişim public (genele açık) olarak ayarlanmaktadır. Eğer bu şekilde projenizi yayınlarsanız kısa zamanda veri tabanınızın ele geçirilmesi, silinmesi mümkündür. Bu problemi giderebilmek için Firebase güvenlik kuralları oluşturmamız gerekiyor. Kullanıcıların Firestore'a yapacağı bütün istekler öncelikle tanımladığınız güvenlik kurallarından geçecek. Eğer Kuralları sağlıyorsa istek veri tabanına ulaşacaktır. Kuralı sağlamıyorsa veri tabanına erişim sağlanmadan reddedilecektir [16][17].

Bütün kurallar match sözcüğü kullanılarak meydana gelir. document=** tüm dökümanlar için bu işlemi uygula anlamına gelmektedir. Eğer kullanıcı login olduysa

tüm okuma ve yazma işlemlerinin yapılması aşağıdaki Şekil 4.13.'teki kural sayesinde olacaktır [16][17].

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth.uid != null;
    }
  }
}
```

Figure 4.13. Example Firestore Rule [16] [17]

4.3. Layered Architecture

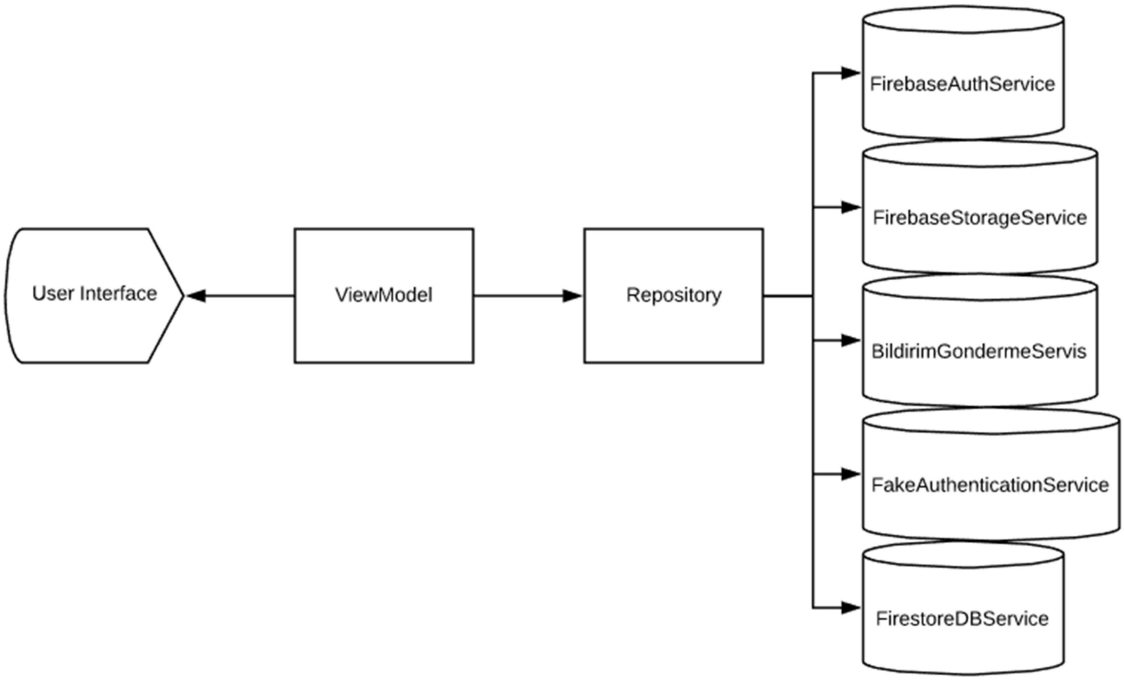


Figure 4.14. Interface - Access Architecture Between Database

When the application first runs, it adds a UserModel to the tree and when it is desired to make large changes, this UserModel class is used to determine the definitions of the methods.

For example, if there is a change in User, he may be logged in. To transmit such requests, notifyListeners is defined to the set state of the model. With this set state method, changes on the screen are detected. The point to be considered here is to check the ViewState with the state set method, not with the _state variable. Otherwise, the changes are not detected at the interface.

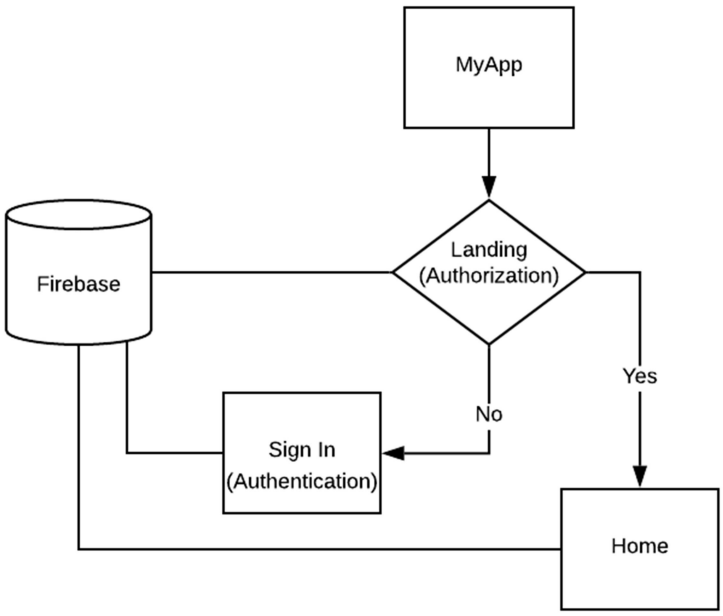


Figure 4.15. Block Diagram of the Application's Layered Architecture

The getIt package in the State Management section saves the classes that will be required as singleton or refactor, and then these classes are called without a constructor if needed. LazySingleton works with call logic when needed.

Username update has been done. If the username is updated, if we close and open the application instead of reading it from Firebase again to print the current username, the new data will be displayed on the screen. Storage_base abstract class has been created for saving photos to the database. The FirebaseStorageService class was created by taking implements from storagebase as a concrete class. Storage service was started in Firebase. Save changes button has been made for upload. Unless it is printed, new data is not added to the data base.

4.4. Database Communication

```
Stream<List<Mesaj>> getMessages(String currentUserID, String
sohbetEdilenUserID) {
  var snapShot = _firebaseDB
      .collection("konusmalar")
      .document(currentUserID + "--" + sohbetEdilenUserID)
      .collection("mesajlar")
```

```
      .where("konusmaSahibi", isEqualTo: currentUserID)
      .orderBy("date", descending: true)
      .limit(1)
      .snapshots();
  return snapShot.map((mesajListesi) => mesajListesi.documents.map((mesaj)
=> Mesaj.fromMap(mesaj.data)).toList());
}
```

Figure 4.16. It is instantly resting with its Stream structure.

It is aimed to continuously bring the whole model instead of bringing the following field from the map. getMessages have been added to the firestore. The messages were printed on the screen, with the last sent to the top. This place is listened to instantly with Stream structure. In Figure 4.16, a snapshot of the database on how to listen to messages is created instantly.

Firebase is subscribe to the conversations in Firebase and whatever is added to it, Firebase provides us with its snapshot streaming structure to rest with its server using TCP protocol. Users can transmit their messages simultaneously, that is, asnychronously. Similarly, operations such as pulling or adding data from the database take place as asc.

The Add field section shown in Figure 4.17 below is used to list the people I speak to. If there are too many users, it will not be possible to take the conversations of only one person in the conversations part, it requires a lot of reading, so add field is used. A call was made that the owner of the speech brings the ones equal to the following id.

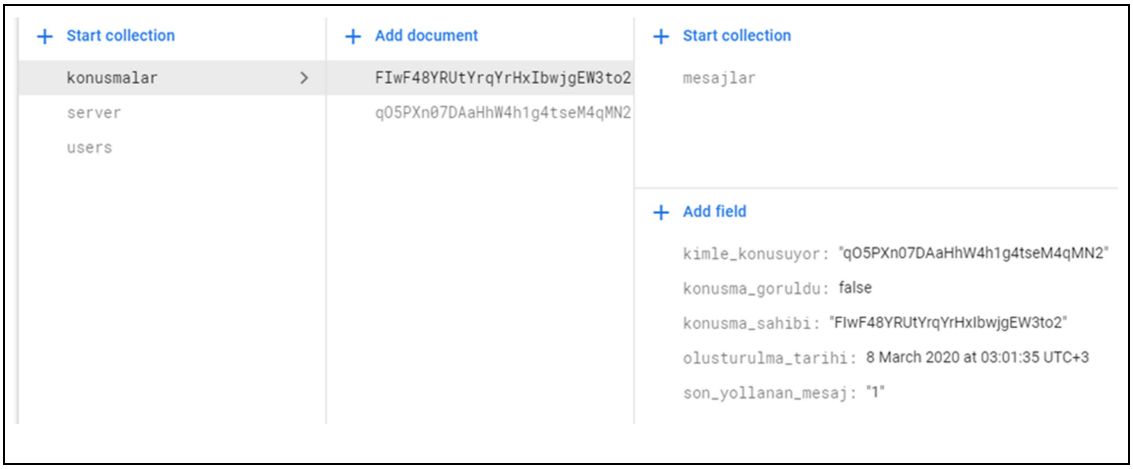


Figure 4.17. Recording of Messages to Firebase

FCM (Firebase Cloud Messaging) is used for operations such as sending notifications to the mobile application. The flow control of the message is described below with FCM [20].

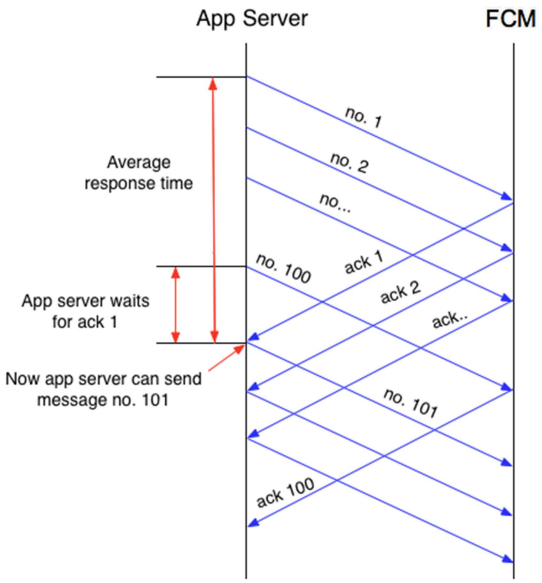


Figure 4.18. Flow control of the message with FCM [20]

Each message sent to FCM receives an ACK or NACK response. Messages that do not receive any of these replies are considered pending. If the number of pending messages reaches 100, the application server should stop sending new messages and wait for FCM to approve some of the pending messages as shown in Figure 4.18. [20]

To avoid overloading the application server, FCM stops sending if there are too many disapproved messages. For this reason, the application server must "ACK" upstream messages received from the client application via FCM to ensure a continuous flow of incoming messages. The above-mentioned pending message limit does not apply to these ACKs. Even if the number of pending messages reaches 100, the application server continues to send ACK to messages received from FCM to prevent the transmission of new upstream messages. [20]

ACKs are valid only in the context of a connection. If the connection is closed before a message is ACKed, the application server is waiting for the FCM to send the ACK message again. Similarly, all pending messages that do not receive ACK / NACK from FCM must be resubmitted before the connection is closed. [20]

If I close the application and send the following notification information via postman, the notification is not sent. Notifications must be specified as Top lavel for this to happen. Classes that do not depend on any class are called top level.

If a notification is to be sent, it will be sent while the message is being sent. For this, the opposite party has a token value. Each device has a token value. Here, instead of pulling the

token value out of the database, I save it to local. Thus, when sending a notification for the message, the token value of the user is not read from the database and local.Each message sent to FCM receives an ACK or NACK response. Messages that do not receive any of these replies are considered pending. If the number of pending messages reaches 100, the application server should stop sending new messages and wait for FCM to approve some of the pending messages as shown in Figure 4.18. [20]

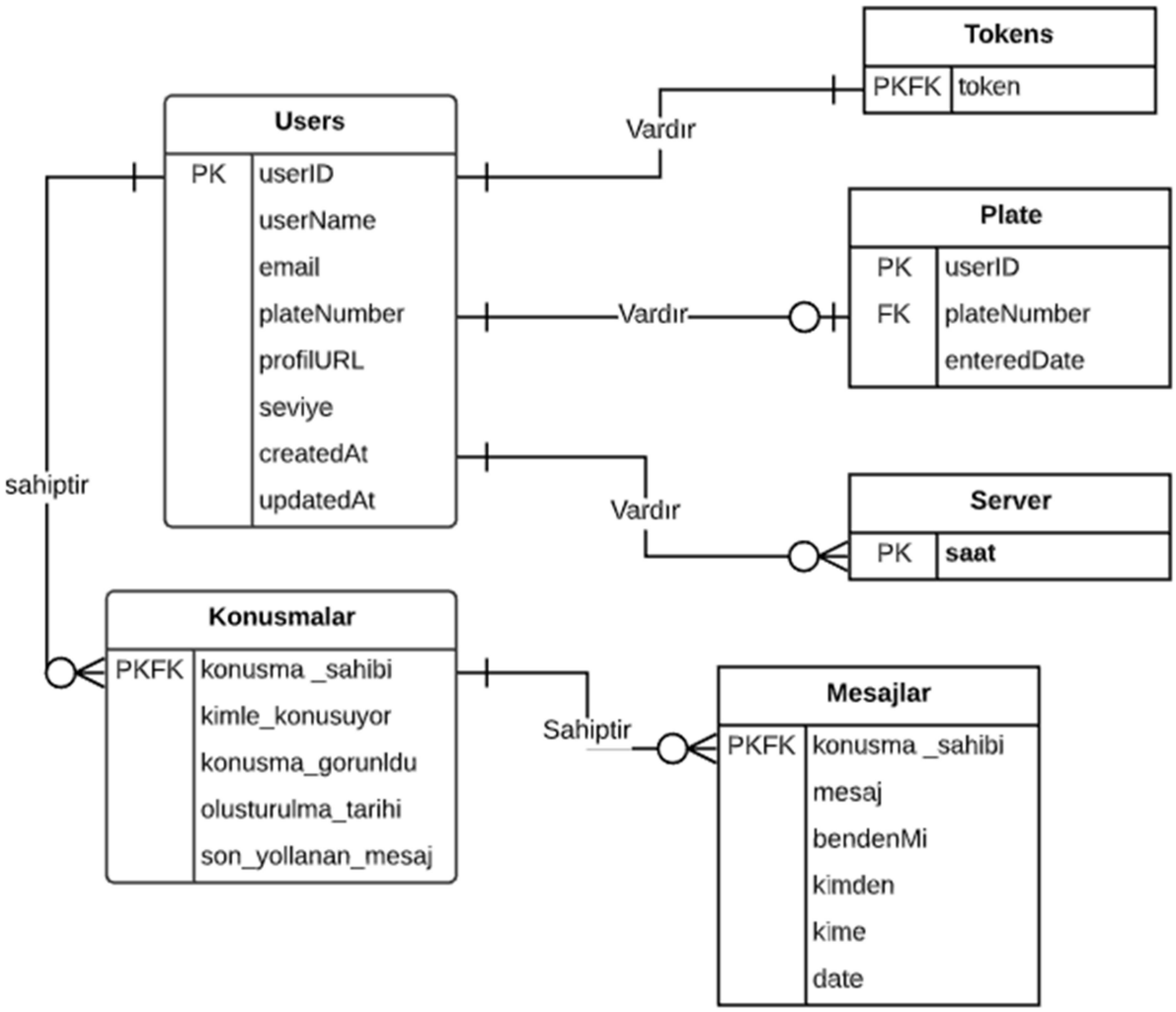


Figure 4.19. Representation of Conceptual Model with Existence Relationship Diagram (Crow's Foot)

To avoid overloading the application server, FCM stops sending if there are too many disapproved messages. For this reason, the application server must "ACK" upstream messages received from the client application via FCM to ensure a continuous flow of incoming messages. The above-mentioned pending message limit does not apply to these ACKs. Even if the number of pending messages reaches 100, the application server continues to send ACK to messages received from FCM to prevent the transmission of new upstream messages. [20]

ACKs are valid only in the context of a connection. If the connection is closed before a message is ACKed, the application server is waiting for the FCM to send the ACK message again. Similarly, all pending messages that do not receive ACK / NACK from FCM must be resubmitted before the connection is closed. [20]

If I close the application and send the following notification information via postman, the notification is not sent. Notifications must be specified as Top lavel for this to happen. Classes that do not depend on any class are called top level.

If a notification is to be sent, it will be sent while the message is being sent. For this, the opposite party has a token value. Each device has a token value. Here, instead of pulling the token value out of the database, I save it to local. Thus, when sending a notification for the message, the token value of the user is not read from the database and local.

<table>
<tr><td>No</td><td></td><td></td></tr>
<tr><td>1</td><td>
<table>
<tr><td colspan="3">Cloud Firestore</td></tr>
<tr><td>Stored data</td><td>$0.18/GiB</td><td>1 GiB total</td></tr>
<tr><td>Network egress(N1 high-memory machine types)</td><td>$292.00</td><td>10GiB/month</td></tr>
<tr><td>Document writes</td><td>$0.06/100K</td><td>20k/günde</td></tr>
<tr><td>Document reads</td><td>$0.06/100K</td><td>50k/günde</td></tr>
<tr><td>Document deletes</td><td>$0.02/100K</td><td>20k/günde</td></tr>
</table>
</td><td rowspan="3">In Firebase, pricing is made with the use and pay logic. Accordingly, monthly average
$350</td></tr>
<tr><td>2</td><td>
<table>
<tr><td colspan="3">Firebase Hosting</td></tr>
<tr><td>GB stored</td><td>$0.026/GB</td><td>1 GB</td></tr>
<tr><td>Network egress(N1 high-memory machine types)</td><td>$0.15/GB</td><td>10GiB/month</td></tr>
</table>
</td></tr>
<tr><td>3</td><td>
<table>
<tr><td colspan="3">Cloud Functions</td></tr>
<tr><td>Invocations</td><td>$0.40/million</td><td>125K/month</td></tr>
<tr><td>GB-seconds</td><td>$0.0025/thousand</td><td>40K/month</td></tr>
<tr><td>CPU-seconds</td><td>$0.01/thousand</td><td>40K/month</td></tr>
<tr><td>Outbound networking</td><td>$0.12/GB</td><td>Google services only</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>monthly expenses and salary required for the project staff</td><td>$1090</td></tr>
<tr><td></td><td>Total monthly expenses according to the current rate</td><td>$1185</td></tr>
<tr><td>5</td><td>One-time fee to publish apps on the Play Store</td><td>$25</td></tr>
<tr><td>6</td><td>Apple App Store Individual / Organization Developer Account for public download apps</td><td>$99/year</td></tr>
<tr><td>7</td><td>Sum of Macbook, Apple iPhone, iPad, high-capacity android operating system phones and tablet hardware to be used for the development and testing of the project, as well as software costs that need to be licensed as well.</td><td>$20185</td></tr>
<tr><td colspan="2">This is a chart arranged for large companies. contains sample data. it is advisory.</td><td></td></tr>
</table>

Table 4.1. BUDGET NEEDS TABLE

SECTION 5. TESTS

For license plate recognition solution, it was tested on static snapshots of vehicles divided into several sets by difficulty. Fuzzy and crooked snapsets provide worse recognition rates than a clear set of snapshots. The main purpose of this project is not to find a 100 percent recognizable snapshot cluster, but to test its invariance in random snapshots systematically classified into clusters according to the characteristics of the algorithms.

Random tests were performed on different pictures on the plate, the results are shown below

Şekil 5.0. Original Image

Şekil 5.1. Gray Image

Şekil 5.2. Eşik Kenar Görüntüsü

Şekil 5.3. Eşik Görüntüsü

With the support of IP Webcam program, the images taken from the camera were obtained by passing them through the Pyhton software and transferred to the web environment [21].

The sill image is obtained from the grayscale image and processed separately to reduce noise.

Figure 5.4. Threshold Edge Image on Plate

In threshold images, features such as lines, edges and ridges are reserved to identify possible characters (Figure 5.4). After the separation of all lines, only the determined character set edges and ridges are determined.

Figure 5.5. Drawing Rectangle with Correct Character

All possible plates of the entire image are defined as rectangles, and the rectangles without correct characters are discarded by processing (Figure 5.5).

Figure 5.6. Cutting Fields of Characters

The last rectangle with the appropriate characters from the previous stage is matched with individually trained characters (shown in Figure 5.6.) And the extracted text is printed on the original image as well as on the screen (as shown in Figure 5.7).

Figure 5.7. Correctly Detected Plate Numbers on the Original Image

Figure 5.8. Low Resolution Faulty License Plate Recognition Example Compilation Time: 3.45e-05

Figure 5.9. Incorrect Plate Recognition Example Taken from Cross Compilation Time: 5.85e-05

Figure 5.10. Incorrect Plate Recognition Example Taken from Cross Compilation Time: 4.23e-05

Messaging is carried out smoothly and in real time. If an improvement is to be made here, it can be used more effectively by an interface update on the Announcement Page. Apart from that, the application can be more user-friendly with the loading and updating events such as photos by making an application page in the form of page loading.

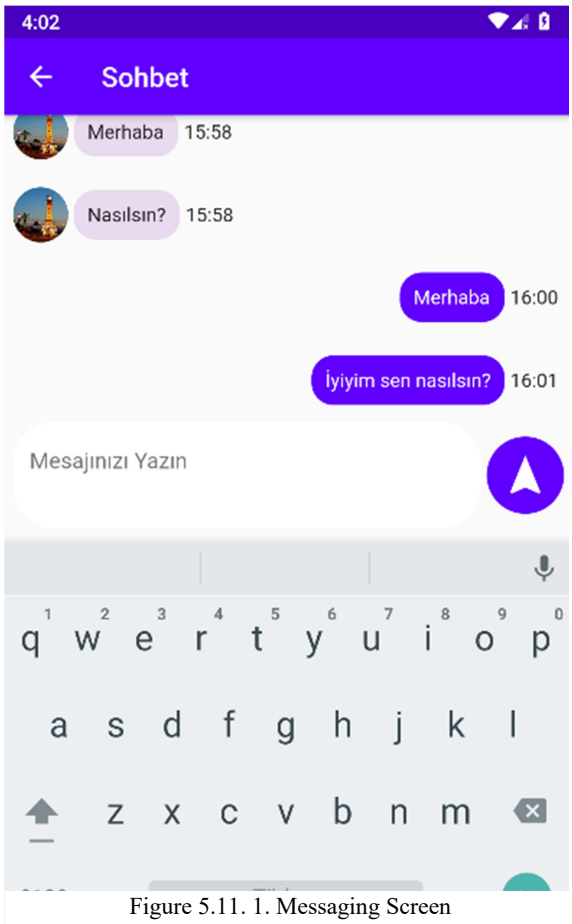


Figure 5.11. 1. Messaging Screen

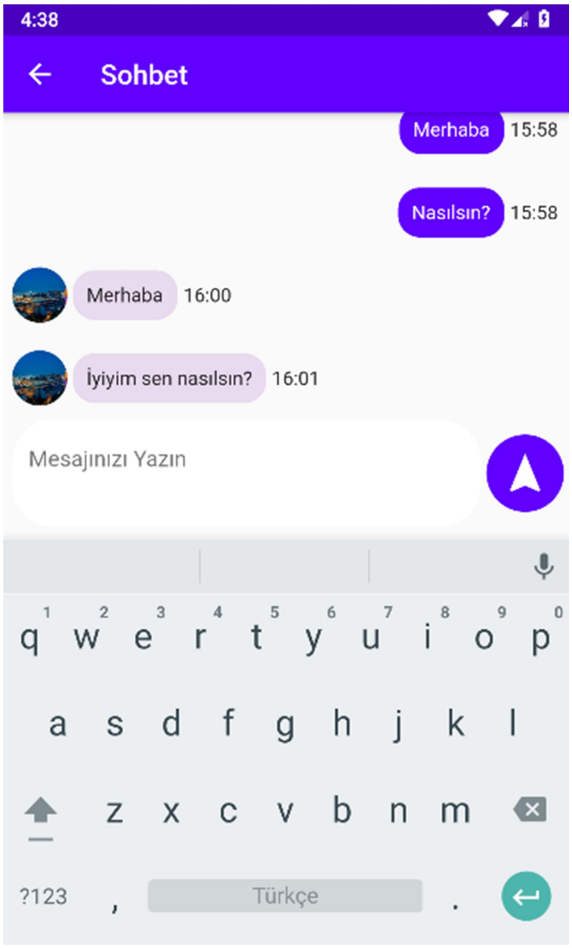


Figure 5.12. 2. Messaging Screen

SECTION 6. RESULTS AND RECOMMENDATIONS

The application range of application is quite wide and it is a project developed to meet the working needs of the personnel of any organization in my work, which is highly possible to work in real time, integrated with privatizations, for the institution

and organization, where communication and vehicle inputs / exits are to be controlled. The methods used are based on open source based OpenCV library and real-time license plate recognition system developed with Python programming language based on K-NN algorithm and open source, Android, iOS, Google Fuchsia operating systems developed by Google for communication between company employees and mobile. The multi-platform Flutter UI SDK, the Dart programming language, and all messaging and user information that have been used, are used to safely store them in the Firebase database, authorization, storage and cloud messaging system.

The system was tested for both local and foreign license plate images with various license plate orientation, plate localization with 92% accuracy, including image quality and size, and character segmentation with 88%. The plate location was determined by using ready-made picture filtering methods throughout the studies on plate systems. These filters are filters such as turning an image in the rgb space to gray, getting a median, applying a ticketeral filter, histogram equalization, edge finder filter, determining a threshold value, contour operation, masking process. These filters are ready filters in the OpenCV library. During the discovery of the license plate area, the brand name, logo or similar characters similar to the license plate on the vehicle and the error of determining these places as license plates were sometimes encountered. For the license plate recognition system to be efficient in the images taken at night, the efficiency is increased by the good adjustment of the light in the environment. The low resolution of the images used in license plate recognition affects the system negatively. According to studies, as the resolution rate increased, the success rate in finding the plate region increased linearly. Stretching can be applied to the image as a recommendation to create a frontal image of the crossed photos. In order for the license plate definition to be successful in real-time systems, the camera must be well positioned at the transition point of the vehicles, the camera's resolution must be high, and the camera must be well protected from external factors.

With the product called Spark Plan, which is a free and limited use offered by Firebase, there are many restrictions on the cloud system such as speed, monthly data entry / exit restriction, storage limitation. It is recommended that if the number of users of companies that use the application for this ease of use by finding the number of users of the companies that use the application, for the ease of use, by passing the speed limits and storage space called Paid Blaze Plan offered by Firebase.

RESUME

Ali Onur Ülker graduated from Recepbey Anatolian Technical High School, Web Programming in the field of Information Technologies in 2015. Then, he started his

university life at Sakarya University in the same year. He completed his hardware internship in industrial automation and PLC programming at TIO Automation. In addition, he worked at ENKA Systems with technologies such as Angular 7, OData, DevExtreme, MsSql C #, etc. and did his software internship on Web API development. He joined the Erasmus program in October 2018 and completed her two-semester education at Rzeszów University.

Best regards…