Receipt Analysis System With OCR

by
Ferhat Ali Tokuç

Engineering Project Report

Yeditepe University
Faculty of Engineering
Department of Computer Engineering
2023

Receipt Analysis System With OCR

APPROVED BY:

Prof. Dr. Gürhan Küçük (Supervisor)	. . . . .
Prof. Dr. Şebnem Baydere	. . . . .
Assistant Prof. Dr. Esin Onbaşıoğlu	. . . . .

DATE OF APPROVAL: 12/06/2023

ACKNOWLEDGEMENTS

First of all I would like to thank my advisor Prof. Dr. Gürhan Küçük for his guidance and support throughout my project.

Also I would like to thank my parents for their support and encouragement throughout my education up to the present.

ABSTRACT

Receipt Analysis System With OCR

Today, businesses keep track of their employees expenses by providing them with allowances and keeping records of these expenses to take advantage of tax deductions. These records are collected through receipts printed on thermal paper. Accountants manually enter the information from these receipts into the system. This manual entry process by accountants creates the following problems:

- Incorrect data entry due to human error
- Slowed down the process, which can lead to inconvenience for employees or the business
- Inefficient management of the workforce due to accountants working below their skill level

These problems indicate that this method cannot continue in its current state, and an automation system is needed. The aim of the project is to establish a system that automates the process to solve these problems. In this project, a machine learning that locates the receipt in the photo and crops it, a proposed algorithm based on Tesseract OCR and a system based on Azure Form Recognizer are designed.

ÖZET

OCR İle Fiş Analiz Sistemi

Günümüzde iş yerleri personellerin giderlerini onlara harcırah vermek ve vergi inidirmlerinden yararlanmak için bu harcamaların kaydını tutmaktadır. Bu kayıtlar termal kağıtlar üzerine basılan fişler aracılığı ile toplanmaktadır. Bu fişler muhasebeciler tarafından elle sisteme geçilmektedir. Bu sisteme fişlerdeki bilgileri girme süreci muhasebicler tarafından elle yapıldığı için şu sorunlar ortaya çıkmaktadır:

- İnsani hatalardan dolayı oluşan yanlış veri girilmesi
- Sürecin yavaşlaması , bu durumdan dolayı personelin veya işletmenin mağdur olabilmesi
- Muhasebicilerin sahip olduğu yeteneklerin altında çalıştırılmasından dolayı kaynaklanan iş gücünü verimsiz yönetme sorunu

Bu sorunlar bu yöntemin artık bu şekilde devam edemeyeceğini , bir otomasyon sistemine ihtiyaç duyulduğunu göstermektedir.Projedeki amaç bu sorunları çözebilmek için süreci otomatize eden bir sistem kurmaktır. Bu proje kapmasında fotoğraftaki fişin yerini bulan ve bunu crop eden bir makine öğrenmesi , Tesseract OCR bazlı bir algoritma ve Azure Form Recognizer tabanlı bir sistem tasarlanmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

OCR	Optical Character Recognition
PDF	Portable Document Format
JPEG	Joint Photographic Experts Group
JPG	Joint Photographic Experts Group
PNG	Portable Network Graphics
BMP	Bitmap
TIFF	Tag Image File Format
UTF-8	Unicode Transformation Format
DONUT	Document Understanding Transformer without OCR
CORD	A Consolidated Receipt Dataset for Post-OCR Parsing
EER	Enhanced Entity-Relationship
GPU	Graphics Processing Unit
BI	Business Intelligence Systems
ERP	Enterprise Resource Planning Systems
CRM	Customer Relationship Management Systems
GUI	Graphical User Interface
ML	Machine Learning

1. INTRODUCTION

Businesses around the world are trying to record their revenues and expenses. The meticulous entry of this information is of great importance due to the support they receive from governments or tax deductions. Although many revenue and expense systems have been automated and freed from human labor, tracking employees' personal expenses and entering them into the system still requires a lot of workforce today. This situation causes businesses to both not be able to use their experienced staff effectively and to spend extra money to carry out these processes.

1.1. Problem Definition

The expenses made by employees are tracked with the shopping receipts they receive. This tracking system traditionally works as follows:

(i) The staff physically delivers the receipt to the accountant.

(ii) Accountant puts the receipt in the processing queue.

(iii) The accountant reviews the receipt.

(iv) Accountant enters information into the system

Companies with the traditional method of receipt processing designate at least one private accountant for this task. In addition to the fact that this person's vacation period causes all work to stop, it is a common occurrence that incorrect information is entered into the system due to human errors while entering the information. When this process is done manually, it can take a serious time. Considering that it is a situation that affects employee morale, you can see the traditional receipt processing method in the Figure 1.1.

Figure 1.1. Traditional Receipt System Flowchart

In summary, the traditional receipt system has the following problems:

- The transition time to the system is very slow.
- It prevents the active and efficient use of skilled personnel.
- It is prone to human errors

An OCR-supported automated application will eliminate all these problems and bring the system to a much faster, less labor-intensive, and more accurate position.

1.2. Motivation

1.2.1. Productivity

Performing a standard process manually with the equipped workforce causes serious inefficient use of human experience, the processes taking too long to complete the work on time, and accordingly the dissatisfaction of the personnel in the workplace and financial losses. One of these processes is to enter the receipts into the system. By automating this process, we can both increase personnel satisfaction and gain serious financial gain by using the workforce efficiently.

1.2.2. As a Basis for Other Applications

Receipt Analysis System With OCR system can become widespread in different areas. Examples of these are invoices and contracts.

1.3. Scope And Limitations

- OCR applications do not work perfectly and replace the ”*” with ”.” Do not assume the character is the most basic example of this.
- OCR applications cannot give high-accuracy results in photos taken with too much or too little light.
- Algorithms trained with OCR often want a perfect angle. Photos of users taken from wrong angles that do not comply receipts this condition reduce the success of OCR.
- When the receipts are exposed to heat, they take on a black or similar color. This reduces accuracy.
- Receipts are papers that can be easily torn, deformed or wrinkled. This situation reduces the success of OCR.

1.4. Requirements

1.4.1. Functional Requirements

(i) The system should be able to upload and process images of receipts from a GUI.

(ii) The OCR engine can be able to extract relevant information from the receipt images such as vendor name, date, and itemized details.

(iii) The system should be able to validate the extracted information against predefined rules to ensure accuracy.

(iv) The system should transfer the extracted data to the further processing.

(v) The system should be able to process receipts in different formats including PDF, JPEG, and PNG.

(vi) The OCR engine should be able to recognize various languages and characters, including special characters and symbols like UTF-8 standart

(vii) The system should be able to handle receipts of different sizes.

(viii) The system should be able to handle receipts of different orientations.

(ix) The system should be able to handle receipts of different light conditions.

(x) The system should be able to handle receipts of different resolution conditions.

(xi) The system should be able to identify and extract data from receipts with varying layouts.

(xii) The system should be able to extract data from receipts with multiple languages.

1.4.2. Non - Functional Requirements

(i) The system should have a high level of accuracy in extracting information from receipt images.

(ii) The system should be able to handle concurrent requests without significant performance degradation.

(iii) The system should be secure and comply with data privacy regulations.

(iv) The system should have a user-friendly interface for uploading and managing receipt images.

(v) The system should be reliable and minimize downtime.

(vi) The system should be scalable and able to handle increasing numbers of users and data processing requests.

(vii) The system should be maintainable and easily upgradable to new versions of OCR technology.

(viii) The system should be cost-effective and provide value to users.

2. BACKGROUND

In this area, there are open source tools that we can use for our own algorithm, non-open source tools and their comparison.

2.1. Donut: Document Understanding Transformer Without Ocr

Speed is very important when analyzing receipts. Therefore, it was considered to use Donut methodology in the project. The usual contribution of the Donut project to our project is as follows:

Although such OCR-based approach promise reasonable performance, they suffer from critical problems induced by the OCR, e.g., expensive computational costs and performance degradation due to the OCR error propagation.You can see the speed difference with this methodology in the Figure 2.1.

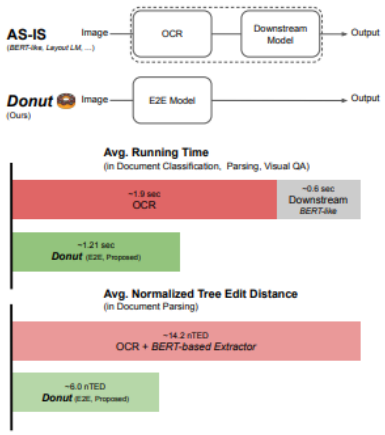

Figure 2.1. The Performance Comparison Of Conventional Visual Document Understanding Architectures and The Proposed Method (Donut). The Lower The Metric Score Is, The Better Performance It Is.

Donut, directly maps an input document image into a desired structured output. Unlike traditional methodologies, our method does not depend on OCR and large-scale real document images.

Although the donut methodology brought cost and speed advantages to our project, it was decided not to be used at this stage of the project due to insufficient community support. [1]

2.2. Cord: A Consolidated Receipt Dataset For Post-ocr Parsing

It is a methodology that helps us classify data on the basis of the cord methodology. The difference of Cord from a standard OCR system is explained in the article as follows.

CORD provides line annotations for the serialization tasks. Current semantic parsing techniques are not usable because current semantic parsing techniques only handle well-ordered texts. We can use this method to transfer the resulting texts from two dimensions to one dimension. This methodology will be used in the algorithm that we will write ourselves to have the ability to classify important information such as product names, store, tax amount.You can see in the Table 2.1 how the data in the example dataset is stored. [2]

Table 2.1. Example Dataset Statistics

No.	Superclass	# Subclasses	Proportion	Example
1	store info.	9	0.134	store name, address, telephone number
2	payment info.	2	0.092	visiting time, card company
3	menu	16	0.510	menu name, quantity, price, submenu
4	void menu	6	0.0002	menu name, quantity, price
5	subtotal	8	0.073	subtotal price, discount, service charge, tax
6	total	8	0.145	total price, amount of credit/debit card
7	void total	4	0.00015	void total, void tax
8	etc.	1	0.045	table number, membership points, repeated symbols
	Total	54	1.0	

2.3. Extending Trocr For Text Localization-free Ocr Of Full-page Scanned Receipt Images

In this methodology, the chips are divided into different chunks and a more strategic data mining is done. The biggest problem in this application is that the target value is 22.8% character error rate. This ratio is far from an acceptable level for a financial record. The chunk logic in this methodology can be used in operational activities such as getting a store name.In the Figure 2.2 you can see the simplified version of the process.[3]

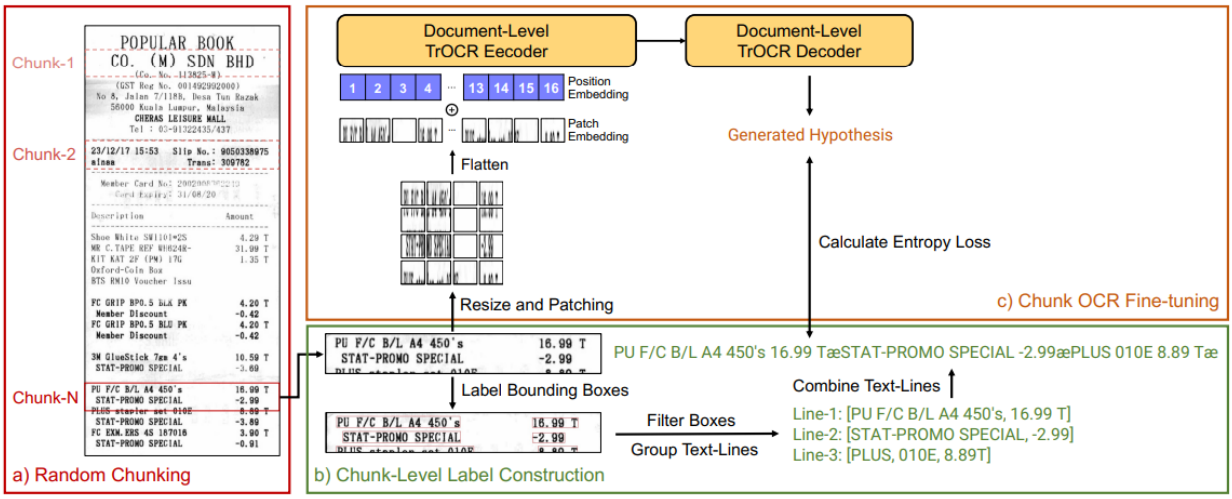

Figure 2.2. Finetuning The TrOCR Model For Chunk-Level OCR.

2.4. Tesseract Ocr Engine

Tesseract OCR Engine is an up-to-date OCR Engine that is supported by many companies and communities, especially Google, and has a low error rate compared to its competitors. We decided to use this Tesseract OCR Engine while developing our own algorithm.

Main reasons for choosing this app:

- Google and Community support
- Easy to be implanted
- Having a proven OCR Engine
- Easy to read and maintain
- Baseline Fitting (This process is very critical since the user cannot take a picture at a perfect angle.)
- Fixed Pitch Detection and Chopping (Shopping receipts can lose their original structure and print quality very easily, since thermal paper is sensitive to humidity and heat, so this feature is very critical.) [4]

2.5. Object Detection with Deep Learning: A Review

Users may not take enough care or create the perfect conditions while taking photos. This is a common problem. For example, showing the keyboard while taking a photo of a receipt and the writings on the coffee cup facing the camera are examples of these situations. These situations pose a problem in extracting the text of a receipt from an image with ocr. Preventing this situation significantly increases the success rate. For this, a receipt database and an artificial intelligence model should be trained and this artificial intelligence model should determine the location of the receipt. OCR operations should only be performed in this specified area. In this way, even if there are many deceptive objects (noise) in the area where the photo was taken, OCR will not be affected by them. [5]

2.6. Tools Without Open Sorce

In this section, the OCR Receipt tools used by the following sites are examined: Pay as you go logic is the method most used by small and medium-sized companies, and in such a use, companies charge per page as follows(US Dolar):

- Nanonets 0.3
- Mindee 0.10
- Veryfi 0.08
- Taggun 0.08
- Azure Form Recognizer 0.01

3. ANALYSIS

3.1. Performance Analysis

First, the alternatives in the Tools Without Open Sorce section were analyzed and it was decided that Azure Form Recognizer was the most advantageous service in terms of success rate and price. In addition to its affordable price and high accuracy, its structure that can be easily integrated into the systems it offers and the improvement of its performance by constantly training with new datasets by Microsoft were effective in this decision. A proposed algorithm was written, inspired by the methods mentioned in the Background section and using the Tesseract library.

These two codes have been tested with the datasets detailed below. Tested results were quantified as a result of manual verification of all results by 4 different people. In order to prevent human error, the workload was not divided into 4 people, and everyone repeated the process from start to finish.

The receipts and receipts used in the comparison may contain sensitive information such as the customer's name, credit/debit card number, date/time of the transaction. This information has been removed due to the protection of personal data.200 images of receipts found on the internet 50 photos taken with different phones in ideal conditions 50 low-light photos with different phones 50 high-light photos with different phones 50 low resolution photos with different phones. The average success of the proposed algorithm I wrote is 37.75%. The average success of the Azure Form Recognizer tool is 99.5%.The Figure 3.1 clearly shows how big this difference in success is.

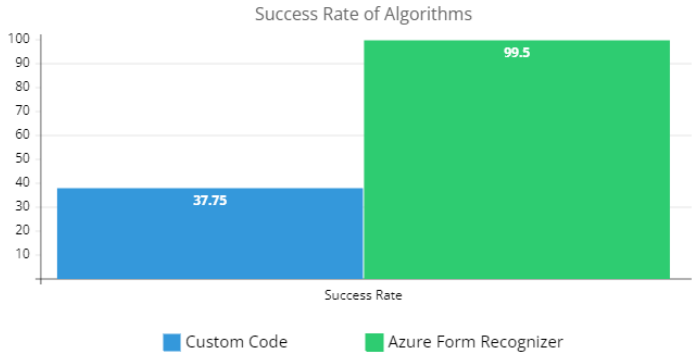

Figure 3.1. Success Rate of Algorithms

There is a serious difference in accuracy between the code we wrote and Azure Form Recognizer. The main reason for this difference is the limited data set. The accuracy rate is the most critical data for a system that will affect the accounting. In addition, since the cost of hosting the code we wrote is more expensive than using Azure Form Recognizer, it was decided to use the Azure Form Recognizer Tool in the project. In this way

- Receipts will be scanned with a higher accuracy
- It will save cost

Thanks to all the improvements Microsoft will make, our system will improve itself to increase accuracy without any data collection, training or testing.

If the form recognizer's fee increases or stops serving, we need a backup plan. For this, we need to increase the success of our proposed algorithm. For this, the following way can be followed. In order to increase this success, we need a better working ML model. The way to make this model better is to train it with more datasets. It is not easy to create such a large dataset. Therefore, this dataset will be developed using the receipt photos and information previously processed with the form recognizer. The main purpose in making this decision is that this dataset will be more compatible with real-life scenarios, as this dataset will consist entirely of photos to be taken by 100% real users.

3.2. Cost Analysis

According to information indeed.com [6] , the average salary of an accountant is $61,709 per year. Side rights are not included in this figure and are excluded from the calculation. Since storage costs are the same in the traditional system and in the Receipt Analysis System, they are excluded from the calculation.

When it is calculated that an employee works for 12 months and works 20 days per month (status like public holidays, annual holidays are excluded from the calculation). This employee works 240 days in a year. When we divide this amount by the salary of 61709 US dollars, the result is approximately 257.12 US dollars per day.The Receipt Analysis System costs:

- $0.01 per page (If you do not have a Commitment. We will base this situation in the calculation.)
- $0.0095 per page (If you have a Commitment.)

.For an accountant to cost the same as a Receipt Analysis System, he needs to process 25712 receipts per day, despite assumptions in favor of the accountant. This is a situation that is far beyond a person's limits. In order to understand this situation more easily, we can give an example as follows. For an accountant to process 25712 receipts (assuming that he works 24 hours a day continuously)

- 1071.3 per receipt in an hour
- 17.8 per receipt in a minute

must be entered into the system. In the light of this information, we see that the Receipt Analysis System has a serious cost advantage.

4. DESIGN AND IMPLEMENTATION

In this section, the design and implementation phases of the project are introduced. The first subsection contains mockup designs and diagrams of the application. The second subsection describes how the implementation is done.

Simplicity is the most important thing while designing the project. In this context, both the GUI to be used by the users and the architecture have been kept as simple as possible. Keeping the GUI side simple has two main purposes. The first reason is that users can easily use this application. This should be avoided as training employees to use a software is a serious waste of time and additional cost for companies. The key to this is the simple GUI design. The second reason is that companies should be able to easily update this application with their own logos, color palette designs or easily add them to their own panels. Because of this, minimal button clicks and a simple GUI are essential.

The reasons for keeping the architecture simple are:

- Understandability: Simple architecture makes it easier to understand the system.
- Maintenance Ease: It simplifies code maintenance and bug fixing.
- Scalability: Simple architecture allows for easier system scaling.
- Testability: It facilitates software testing and ensures higher quality.
- Performance: Simple architecture often leads to better performance.
- Debugging: It simplifies the process of identifying and fixing errors.
- Collaboration: Simple architecture promotes effective teamwork.

The architecture should be very understandable, error-free, flexible, and simple so that integration to different systems or connectors can be done easily. This not only reduces the initial investment cost of the project but also provides speed and savings in the later stages of the project.

4.1. Design

The design of the project was made for two different situations. The first case is receipt recognition with Azure Form Recognizer, a paid service. The second situation is the receipt recognition with proposed algorithm. Since the receipt recognition stage is the most critical stage for this project, the Design stage is divided into two different branches. These are, in order:

- The System Created Using Form Recognizer
- The System Created Using Proposed Algorithm

You can check it from its subsections. Design Assumptions, Mock-up and Final GUI are common to both.

Making it with proposed algorithm is not recommended due to low accuracy , low speed and needing more CPU power .

4.1.1. Mock - Ups

It may not be a system that every workplace uses, or it may not want to spare effort and resources to integrate it into its system. Therefore, the application must have a login page. You can see the mock-up of the login page in the picture.

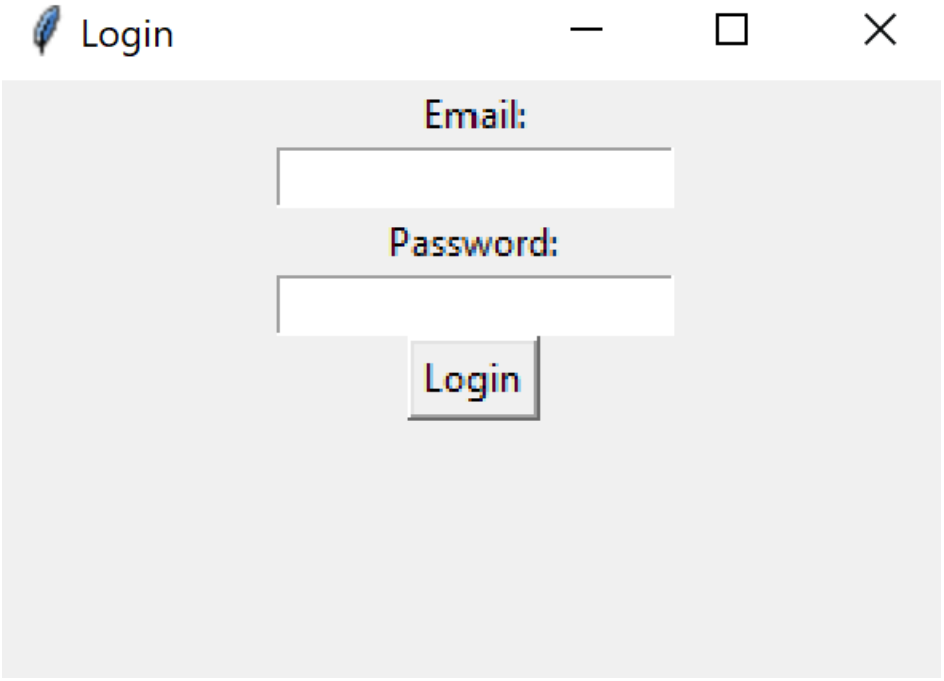

Figure 4.1. Mock-Up Login Page

The user should be redirected to a home page after successfully logging in. On this page, they should be able to choose which receipt image to employ, send it to Form Recognizer, see the result and send it to the database. The reason why the Send Form Recognizer and Send to Database keys are separate allows the user to observe the results and notice a possible error situation. This is the preferred design choice to subsidize the possible errors of the Form Recognizer, which operates with a 99.5

If the user presses the Send Database button more than once, sending the same receipt information to the database more than once is a serious problem. This problem may cause a receipt to be processed more than once and accordingly miscalculation. In order to prevent this situation, the Send Databese button is designed in such a way that it cannot be pressed repeatedly. It is only active once with the data from the Form Recognizer, and after being used it passively waits for another result from the Form Recognizer. You can see this status in the main menu mock-up.

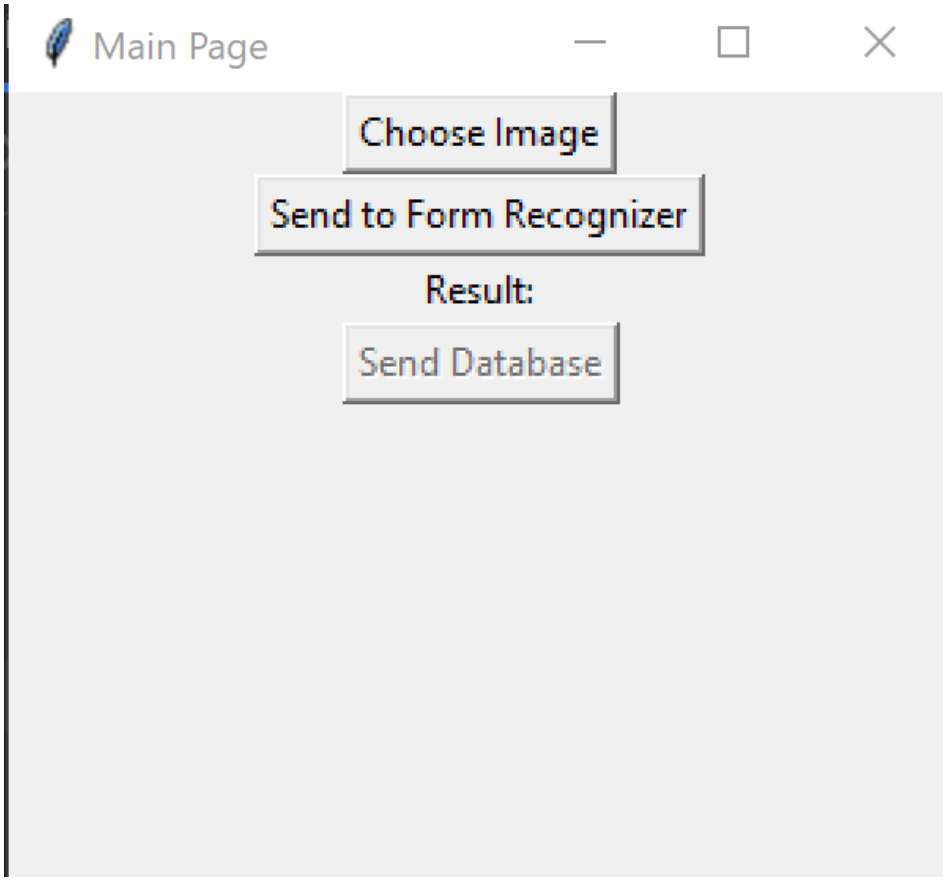

Figure 4.2. Mock-Up Main Menu

4.2. Final GUI

The user interface has been tried to be kept as simple as possible. Companies can easily put their own logos on this interface, and the easy customizability of the source code allows to easily develop independent guis.

4.2.1. Login Page

At the top of the login screen, the company logo and the necessary inputs (mail, password) that the user will enter are boxes. The user can log in after entering this information.You can see this design in the picture

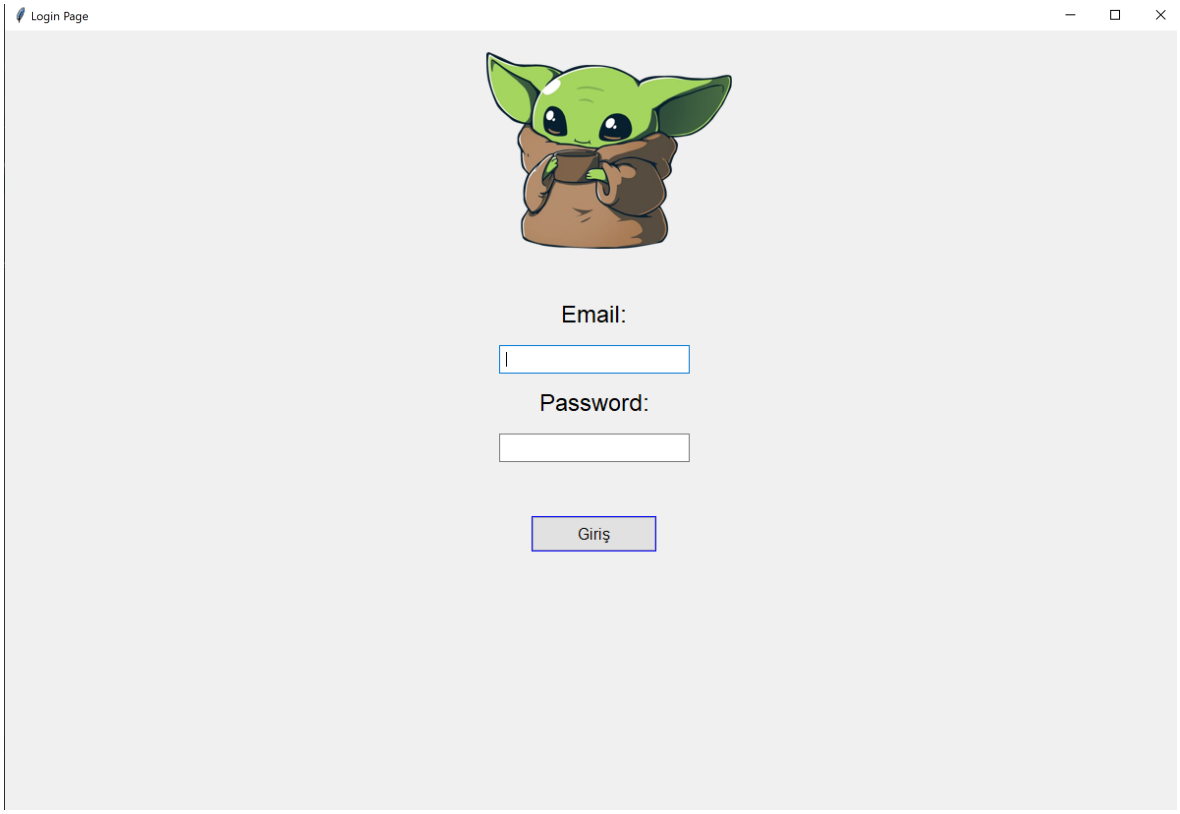

Figure 4.3. Login Page

4.2.2. Main Menu

Users can easily select the file to be processed in the Main Menu, after making sure that they have selected the right file (This selected file is indicated by the path. The main purpose of making this design is to prevent the wrongly selected files from being sent to the Form Recognizer and causing unnecessary costs.) The user can view the answer after submitting the Form Recognizer and After checking, it can send it to the relevant database. In order for users to use this interface easily, the GUI (buttons and table) has been ordered from top to bottom in order of operation. You can see this design in the picture

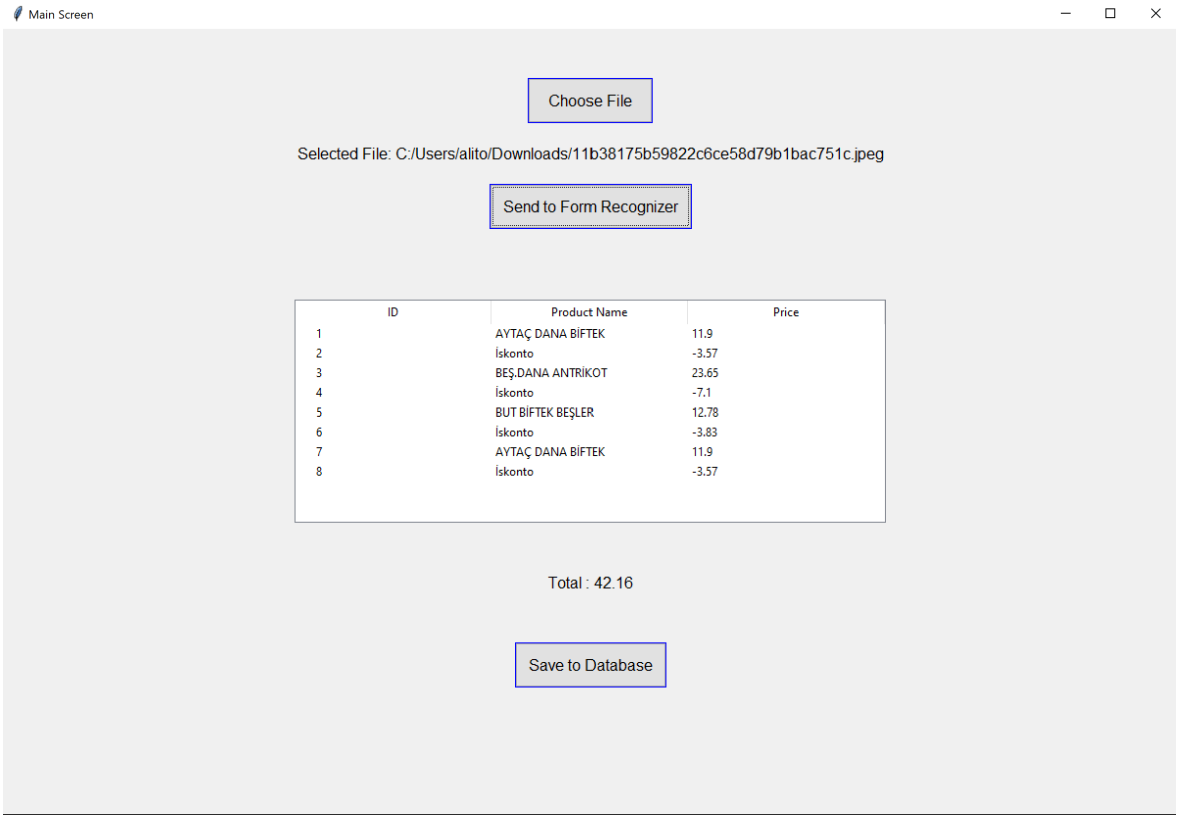

Figure 4.4. Main Menu

4.3. Design Assumptions

While designing, the following conditions were assumed and taken into account.

75-80 percent of people in the world use Windows operating system on their business computers and/or personal computers. Therefore, the main target operating system of the program is the Windows operating system.

The age range of accountants is 25-50. This situation requires us to stay away from the use of GUI with complex interface.

Office computers generally do not have external hardware that we call GPU. This obliges us to work only with the processing unit we call the CPU. Since office computers are generally not very powerful or up-to-date devices, the processing power required for the program to run should be as low as possible. Below are the minimum and recommended system requirements for this program to run.

Recommended System Requirements

- Processors:Intel® Core™ i5 processor 8th Gen
- 16 GB of RAM
- Operating systems: Windows® 10

MINIMUM SYSTEM REQUIREMENTS

- Processors: Intel Atom® processor
- 4 GB of RAM
- Operating systems: Windows® 7

Since office computers usually have Windows operating system, application requirements are determined for Windows operating system. Python code can be easily adapted to Linux, MacOs operating systems.

4.3.1. Roles Of Users

It has been assumed that the application has one type of user. This is the user role.The duties / responsibilities of the User role while using this program are as follows:

(i) This user must enter his e-mail address and password to access the system.

(ii) After gaining access to the system, he should select the relevant image file via the GUI and send it to the Form Recognizer by clicking the button.

(iii) The data from Form Recognizer will be reflected to the GUI. Here, the user can check the products with his eyes or just look at the total amount.

(iv) After these processes, the user can click the ”Send To Database” button and send the receipt information to the company’s database.

No Accountant role is defined in this program. The main reason for this is that people in this role can control this data from MySQL databases or view or analyze this data through other applications when collaborating with other systems.

4.4. Design Based On Form Recognizer

In this section, we will examine the diagrams drawn on the basis that the receipt reading, interpretation and analysis part of the project is done by Azure Form Recognizer. The main reason for separating these parts as Azure Form Recognizer or proposed algorithm is that receipt reading, interpretation and analysis part is a very critical stage for this project, accordingly we want to increase the success rate by adding additional stages such as proposed algorithm a Receipt Detector Machine Learning Algorithm.

4.4.1. Diagrams of The System Created Using Form Recognizer

The diagrams of the application presented in Figures 4.5, 4.6, 4.7 and 4.8 The events described in the diagram are briefly as follows:

The user first enters the username (Mail address) information and password on the login screen and performs authentication by pressing the login button.If the authentication process is successful, the second screen welcomes it. On this screen, the user selects the picture of the receipt they want to trade. This selected photo is stored in azure blob storage. The url of the stored file is taken from azure blob storage and sent to azure form recognizer. The response is in the form of json. After this is parsed, it is reflected in the GUI and the receipt information is sent to the database when the user clicks the Send to Databese button.The place to be noted in each diagram is indicated under the relevant heading.

4.4.1.1. Use Case Diagram - Form Recognize Verison . In this diagram we see that the user is using the following resources:

- Authentication Server
- Blob Storage
- Azure Form Recognizer
- Receipt Database

We can see how and with which inputs and outputs they use these resources, on the Figure 4.5.

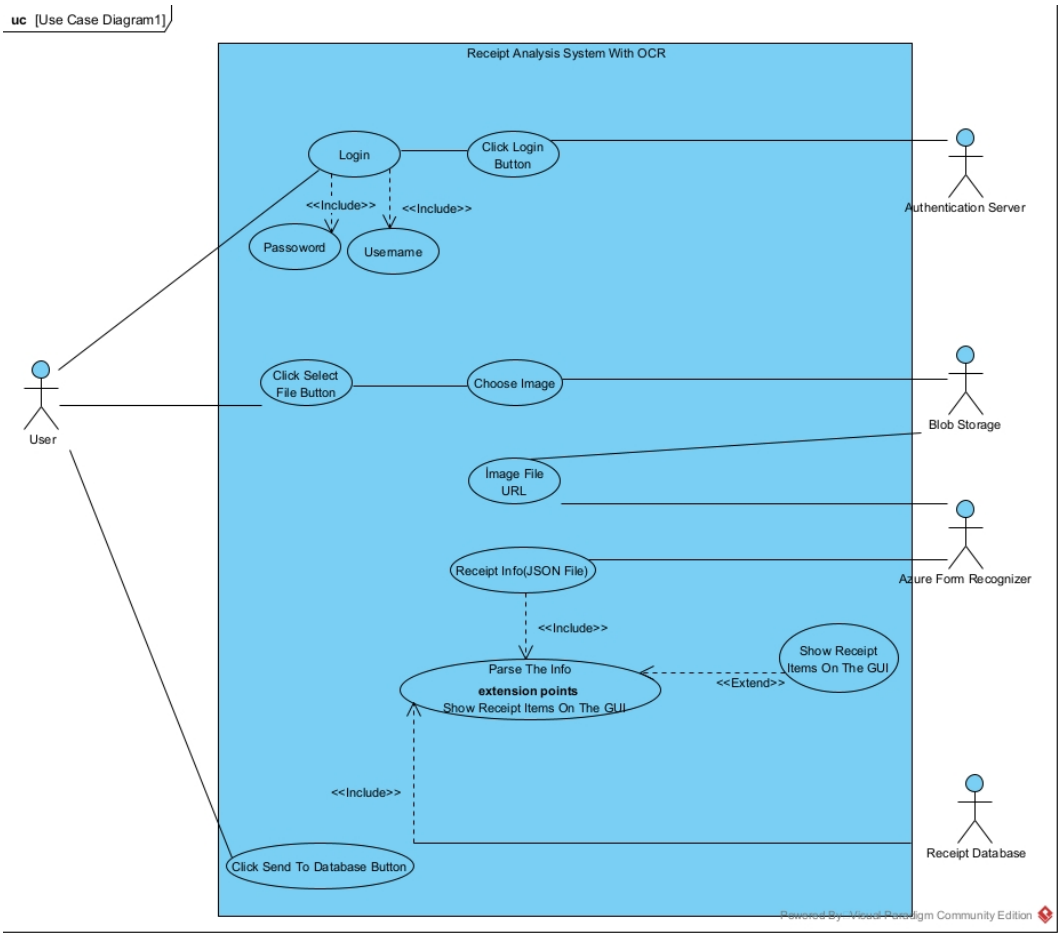

Figure 4.5. Use Case Diagram

4.4.1.2. State Machine Diagram - Form Recognize Verison . In this diagram, we see what keys and operations the user must do to use the application. What we need to pay attention to here is that it can go back to the ”User Click Choose File Button” stage without going back to the beginning. This shows that the program has the ability to process as many receipts sequentially as it wants, without the user having to exit the program after a single login. You can follow this order with the arrows in the Figure 4.6.

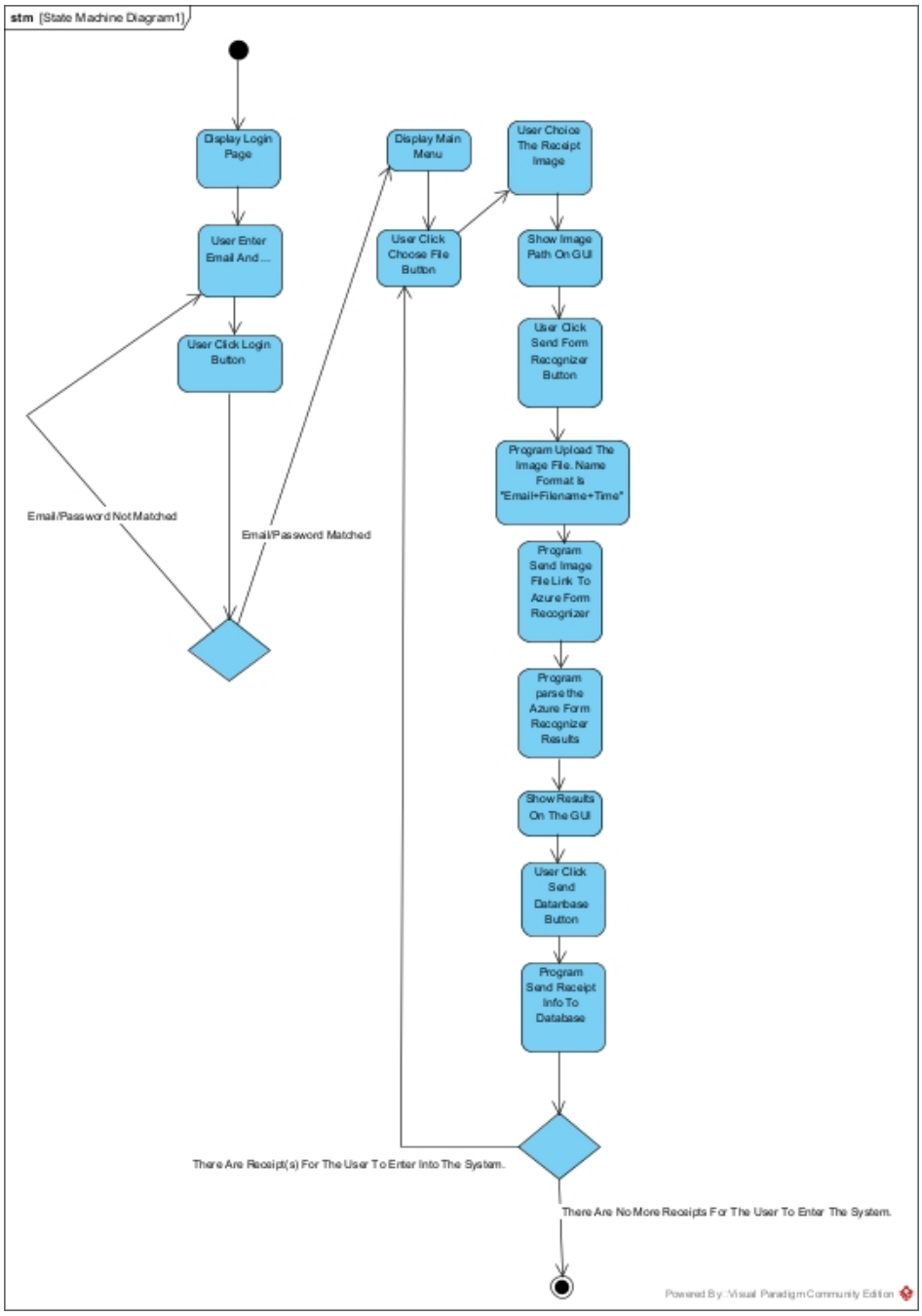

Figure 4.6. State Diagram

4.4.1.3. Sequence Diagram - Form Recognize Verison . In this diagram, the user can clearly see what stages the information entered, from which sources and data flow.In addition, the working logic of the authentication server is explained in Sequance Diagram. You can see the stages of the program by following the arrows in Figure 4.7.

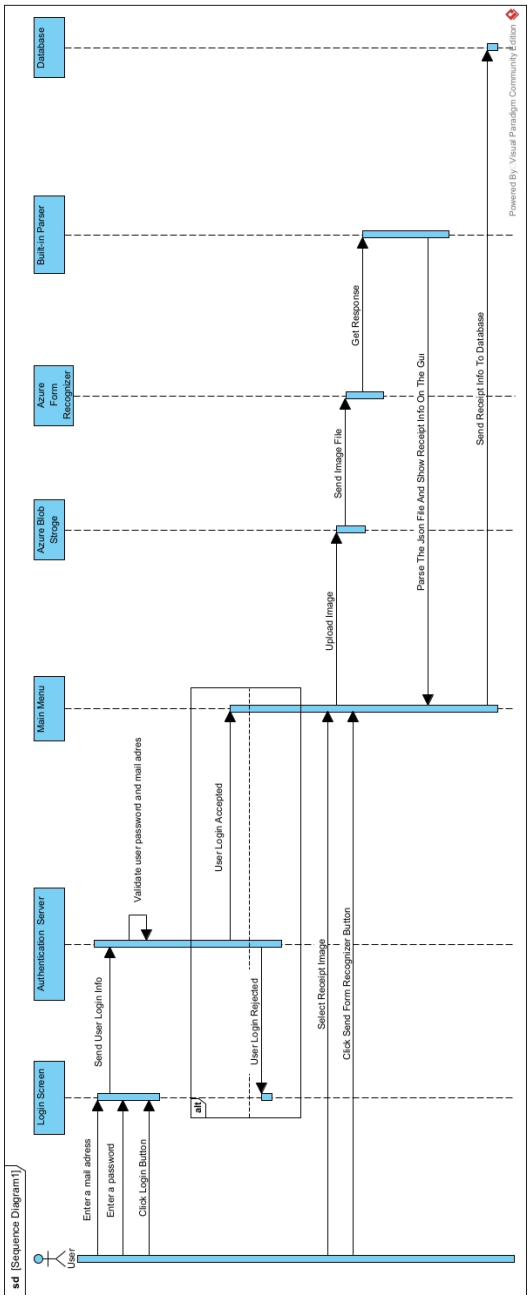

Figure 4.7. Sequance Diagram

4.4.1.4. EER Diagram. **EER Diagram in both methods (Form Recognizer and Proposed Algorithm) are the same.**

There are 4 tables in the MySQL diagram.In the Users table, each user has a unique id.

- The username and password values in this table are not mandatory. The reason it was designed this way is to provide flexibility for small businesses to use this service without an extra authentication server.
- In the Receipt table, the receipts have a unique ID, the date of purchase, the total tax amount and the total amount are kept in this table.
- The receipt and accuracy values from Azure Form Recognizer are stored in the Item Table. The reason for storing accuracy values is that it gives an option to start an investigation starting from the lowest accuracy rate in possible suspicious situations.
- Each item has its own unique id. Since there can be one or more products in a receipt, the Item has Receipt table allows us to establish a logic between the receipt and the products. You can examine the relationship between the tables on Figure 4.8.

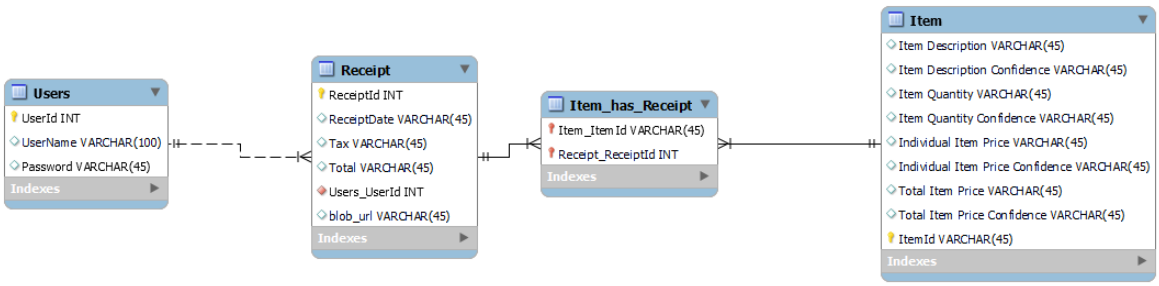

Figure 4.8. EER Diagram

4.5. Design Based On Proposed Algorithm

In this section, we will examine the diagrams drawn on the basis that the receipt reading, interpretation and analysis part of the project is done by proposed algorithm. The main reason for separating these parts as Azure Form Recognizer or proposed algorithm is that receipt reading, interpretation and analysis part is a very critical stage for this project, accordingly we want to increase the success rate by adding additional stages such as proposed algorithm a Receipt Detector Machine Learning Algorithm.

4.5.1. Receipt Detector Machine Learning Algorithm

It is quite common for users to have objects other than the receipt, which we can call noise when taking receipt pictures. If the photo is worked with OCR without any processing, the texts on the other objects that are not in the photo will be processed and this situation:

- It will reduce the accuracy value of the receipt
- It will complicate the analysis of the receipt.
- It will reduce the processing speed of the receipt

In order to prevent this situation, a machine learning algorithm that scans the photo, determines the location of the receipt and runs OCR only in this location is essential for this project.For this machine learning algorithm, photos taken from different countries on the internet were collected in a database and a machine learning algorithm was developed over it. You can see on Figure 4.9 of what the receipt detection looks like below. The writings on the products that we may encounter frequently in daily life such as glasses, books, medicines and berets are skipped and only the location of the receipt is marked.

Figure 4.9. Receipt detected with Receipt Detector Machine Learning Algorithm

A total of 250 photos were used to train the machine learning model. These images were taken from various locations and different lighting conditions to give a wider perspective and more general understanding of the model. In the process of labeling images, the LabelImg tool was used, which provides a fast and error-free labeling process.

In the training phase of the model, the YOLOv5 algorithm was used, along with the 'patience' argument. The 'patience' argument allows the training to stop if there is no improvement in the model's performance over a certain number of epochs. This helps to avoid unnecessary training cycles that might lead to overfitting.

In this instance, the 'patience' value was set to 5 epochs. This means that if there was no observed improvement in the model's performance across 5 consecutive epochs, the training process would halt. By setting a low 'patience' value, we ensure that the model optimizes its learning efficiently without getting stuck in prolonged training cycles.

The training process was conducted with a batch size of 32, where a batch represents a subset of the training dataset. An epoch, on the other hand, refers to a full pass through the entire training dataset. By specifying the batch size, we can control the number of training samples processed before updating the model's parameters.Furthermore, the model attained its maximum learning capacity at the 69th epoch without succumbing to overfitting. This means that the training was able to achieve the desired performance within 69 epochs, indicating successful training.

To elucidate, an epoch is a full pass through the entire training dataset, and a batch is a subset of the training dataset. Overfitting is a common problem in machine learning, where

a model becomes too attuned to the training data and performs poorly on unseen data. The 'patience' argument, therefore, helps to guard against this by stopping training when no further improvement is being made, preventing unnecessary complexity and promoting model generalizability.

As a result, the machine learning algorithm developed through this process is capable of autonomously identifying the locations of receipt photos. This is a significant advancement for automation processes, improving operational speed and efficiency.A different dataset consisting of 50 receipt photos was used for the test. The success rate of this dataset is 94.7%

Ultimately, the use of 250 diverse images for training, the employment of the LabelImg tool for tagging, the implementation of the YOLOV5 algorithm, and the careful setting of the 'patience' argument to prevent overfitting, all culminate in a robust and reliable machine learning solution for the automated identification of receipt photo locations.

4.5.2. Receipts Data Extraction with Tesseract

After the location of the receipt is determined with the Receipt Detector Machine Learning Algorithm, we need to read the texts on the receipt there with the Tesseract library (pytesseract) of the python software language. This library processes the text written in the text and turns it into an output. While doing this, it uses many different techniques, especially grayscaling. The techniques I use in my code are as follows:

- Grayscaling
- Detect Regions
- Text box detection
- Text recognition
- Regular Expression
- Filtering

After going through these stages, we have the products and their prices. If the photo quality is below the expected standard, these detected contents may be inaccurate or incomplete. Expected features are:

- Shot the photo with the right light and angle, clear enough to run OCR on
- The receipt paper is not damaged
- Texts are not damaged

The determination of these products and prices alone is not enough. This information may be incomplete or inaccurate. For this we need to use an algorithm. This algorithm detects possible abnormal situations. These detected situations must be reported to the user and possible reversible / irreversible errors must be prevented.

4.5.3. Making sense of data detected with Tesseract OCR

When we do OCR scanning with Tesseract, the data we get is in the form of a blob. This blob consists of a collection of letters placed side by side or under each other. Being able to correctly interpret and process this data is the most critical part of the whole process. This blob data must be text-mined through a parser. This is the most basic way to make sense of OCR data verified with Tesseract. For example, one of the parsers that can be used to text mining a fee on this blob is: amounts = re.findall(r'/d+./d2/b', text)

The basic things we will do while doing this interpretation process are as follows:

- We need to separate this parceled data into product and price for later use.
- We should search for keywords such as Total , Tax , Date in the blob data and criticize if found.
- Since the accuracy of the data received with Tesseract OCR is very low compared to the form recognizer, we should try to detect potential errors with some analysis applications.

If the receipt information does not encounter any errors during the interpretation process, this information can be sent to the database containing the receipt information.

4.5.4. Analyzing Receipt Information

It is very important to analyze the receipt information and, accordingly, to confirm the information. Tesseract OCR can read numeric and punctuation data incorrectly. For example, it can read the number 6.8 as 7.8. In this case both ”,” instead of ”.” sign is read and 7 instead of 6 is read. These situations seriously reduce the accuracy of the information read from the receipt photo and make it unusable in financial systems. In order to prevent this situation, we are trying to detect possible errors with two different analyzes.

4.5.4.1. Control Of Products With The Same Name. The products in the markets have a unique name. The prices of products with this unique name may be the same or different, but the prices of products with the same name cannot be different. There are two main exceptions to this. These are :

- The first exception is products sold on a weight basis. These products are written as weight X unit price.
- The second exception is the multiple purchase exception. This exception occurs when we receive the same product more than once. Products in this situation are written as Item count X unit price.

The common point of these two exceptions is that they are arranged in the form of a number at the beginning, an X in the middle and a repeating number. When we scan all the lines that do not have such information in the top line, we can understand that there is a problem in the receipt information if a product has the same name and different price.

4.5.4.2. Tax - Total Control Mechanism. Countries have set various tax rates for various product categories. For example, this situation has been determined as a minimum of 1% and a maximum of 18% in Turkey. Based on this situation, we can use the tax amount to confirm whether the total is calculated incorrectly or not, but this cannot give definite results since we do not calculate on a fixed percentage rate. However, we use it in our project because it helps us find errors.

For example, a $100 purchase should be taxed between $1 and $18, according to the taxation conditions of Turkey, which I mentioned above. All receipts that do not meet these conditions must be re-examined. This should be customized for each country, as tax rates do not have fixed values all over the world.

4.5.5. Using Form Recognizer to Develop a Proposed Algorithm

We can increase our proposed algorithm success by using Form Recognizer. This may occur in order not to share sensitive data with the outside or not to depend on a 3rd party service.

The working logic of the form recognizer is as follows: Form Recognizer takes a photo as input. Form Recognizer handles this. Form Recognizer stores the information it processes as json.

The locations of the words are kept in this json file. If we create a new dataset by keeping the original photo and this json file, we can bring the low performance rate of the proposed algorithm closer to the 99.5% success rate of the Form recognizer. The representation of this process as a Flowchart is as follows:

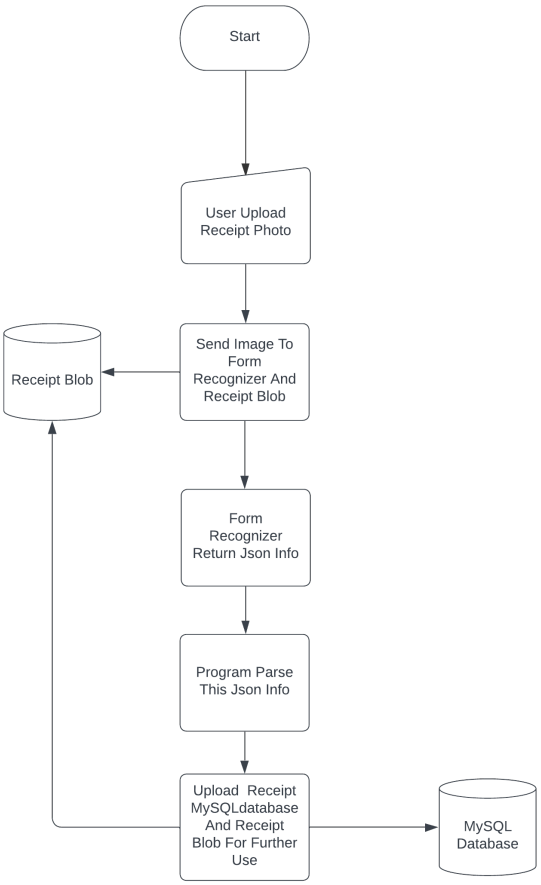

Figure 4.10. Using Form Recognizer to Develop a Proposed Algorithm

The response of the system created with proposed algorithm and Tesseract and the response of Form Recognizer are compared with the following parameters: date

(i) Store Name

(ii) TOTAL

(iii) Total Tax

(iv) Product Count

(v) Product price (Each Item)

4.5.6. Diagrams of The System Created Using Proposed Algorithm

4.5.6.1. Use Case Diagram - Proposed Algorithm Version. When the use case diagram below is examined, we see that the Receipt Detector Machine Learning algorithm processes the receipt photo, and as a result of this processing, it creates a photo in which only the receipt is visible. Afterwards, this photo is sent to the proposed algorithm and after the necessary operations, the user clicks the Send Database button and is sent to the database.You can see this in the figure below.

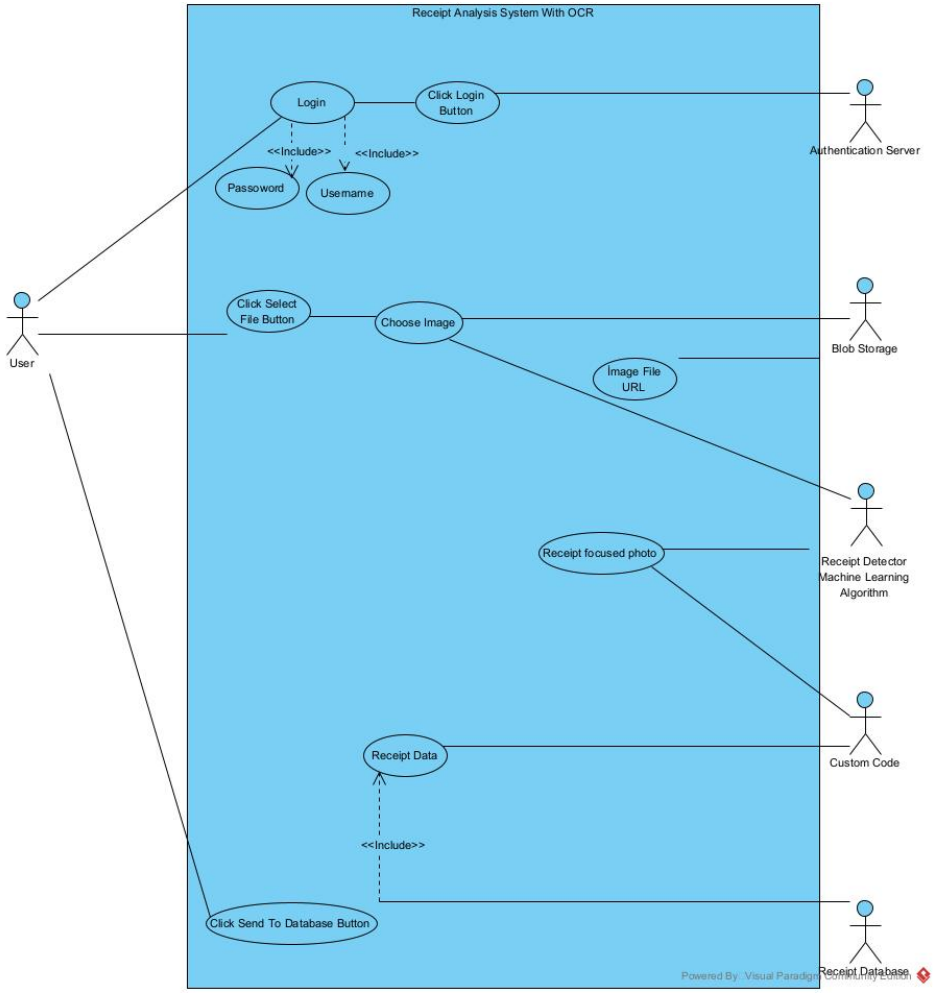

Figure 4.11. Use Case Diagram - Proposed Algorithm

4.5.6.2. State Machine Diagram - Proposed Algorithm Version. In this diagram, we see what keys and operations the user must do to use the application. What we need to pay attention to here is that it can go back to the ”User Click Choose File Button” stage without going back to the beginning. This shows that the program has the ability to process as many receipts sequentially as it wants, without the user having to exit the program after a single login. You can follow this order with the arrows in the State Diagram.

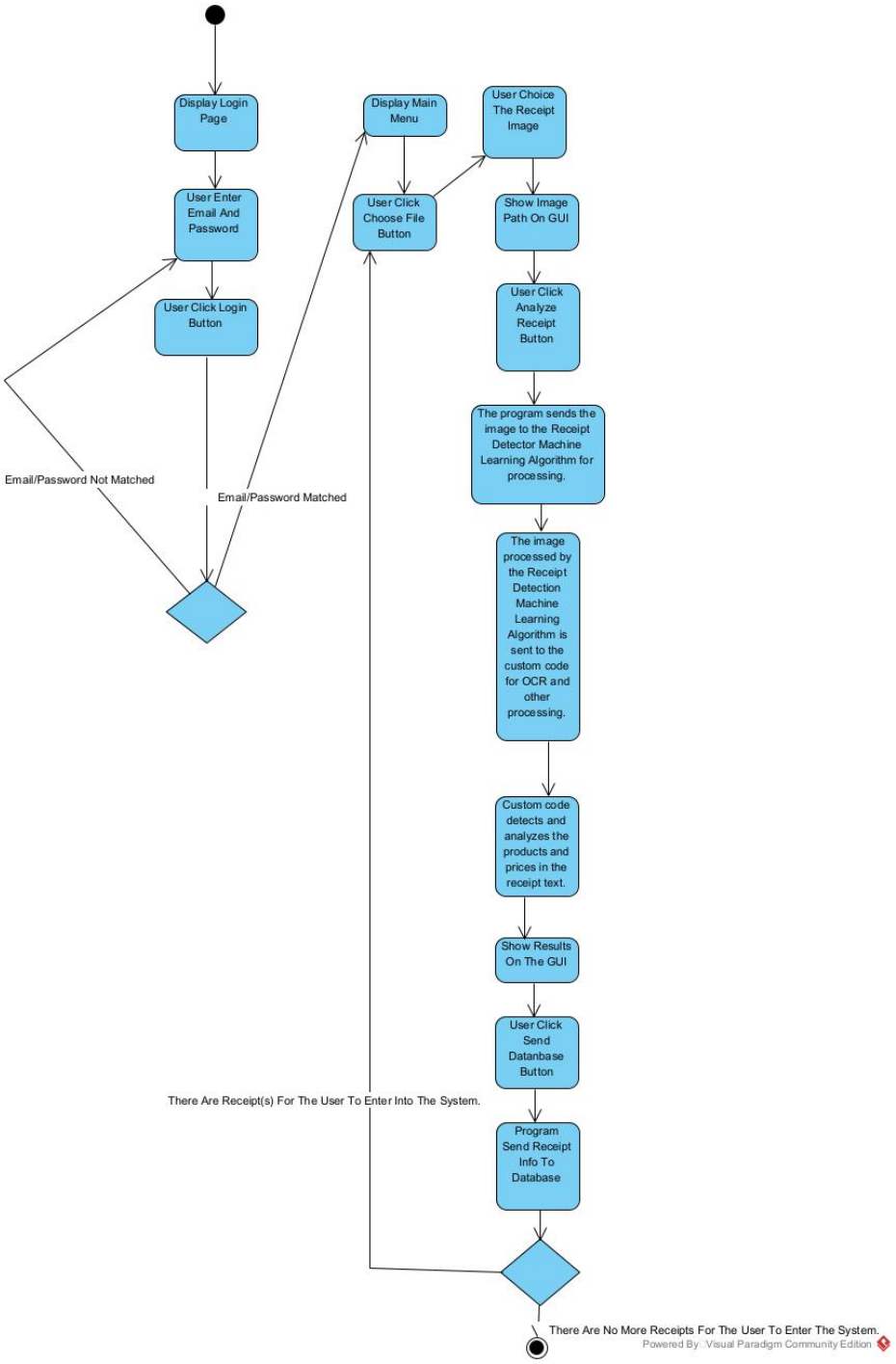

Figure 4.12. State Machine Diagram - Proposed Algorithm

4.5.6.3. Sequence Diagram. In this diagram, the user can clearly see what stages the information entered, from which sources and data flow.In addition, the working logic of the authentication server is explained in Sequance Diagram. You can see the stages of the program by following the arrows in the picture.

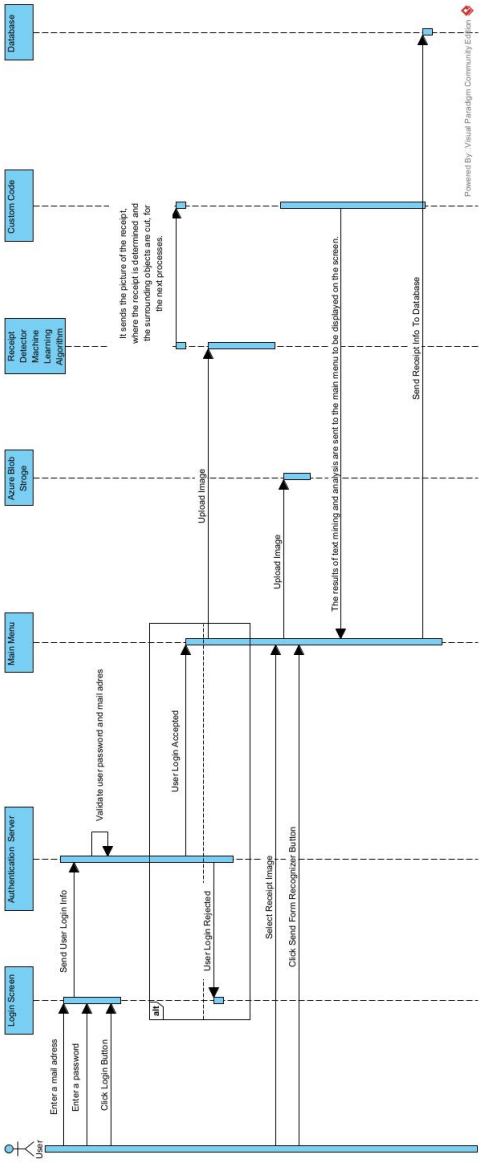

Figure 4.13. Enter Caption

4.5.6.4. EER Diagram. **EER Diagram in both methods (Form Recognizer and Proposed Algorithm) are the same.**

There are 4 tables in the MySQL diagram.In the Users table, each user has a unique id.

- The username and password values in this table are not mandatory. The reason it was designed this way is to provide flexibility for small businesses to use this service without an extra authentication server.
- In the Receipt table, the receipts have a unique ID, the date of purchase, the total tax amount and the total amount are kept in this table.
- The receipt and accuracy values from Azure Form Recognizer are stored in the Item Table. The reason for storing accuracy values is that it gives an option to start an investigation starting from the lowest accuracy rate in possible suspicious situations.
- Each item has its own unique id. Since there can be one or more products in a receipt, the Item has Receipt table allows us to establish a logic between the receipt and the products. You can examine the relationship between the tables on the EER diagram.

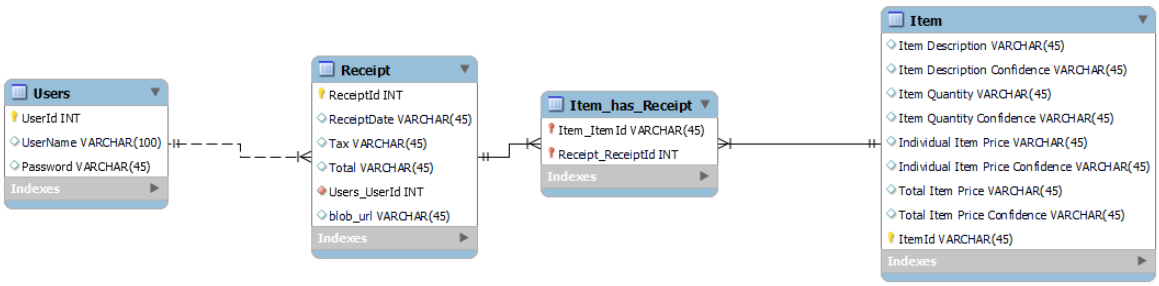

Figure 4.14. EER Diagram

4.6. Implementation

Companies can grow or shrink over time. In this case, it is very critical that the applications used by companies are scalable. We use cloud infrastructure to provide this feature and pinpoint the needs of workplaces.

Microsoft Azures was chosen as the cloud service provider in the project. The reasons for this are as follows.

- Highly Scalable Structure
- Lower costs
- Data backup architecture

4.6.1. Azure Database for MySQL

In our project, we will first create an Azure Database for MySQL servers -Flexible server resource. To create this resource, follow the steps below in order.

(i) Sign in to the Azure Portal: Go to the Azure Portal (portal.azure.com) and sign in using your Azure account.

(ii) Create a new resource: Click on the ”Create a resource” button on the Azure Portal dashboard.

(iii) Search for ”Azure Database for MySQL”: In the search bar, type ”Azure Database for MySQL” and select it from the list of available resources.

(iv) Click on ”Create”: On the Azure Database for MySQL overview page, click on the ”Create” button to start the creation process.

(v) Configure the basic settings:

(vi) Subscription: Select the Azure subscription you want to use for the MySQL server.

(vii) Resource group: Choose an existing resource group or create a new one to contain the MySQL server.

(viii) Instance name: Provide a unique name for your MySQL server.

(ix) Region: Select the Azure region where you want to deploy the MySQL server.

(x) Pricing tier: Choose the pricing tier based on your requirements (e.g., Basic, General Purpose, Memory Optimized).

(xi) Configure server settings:

- Administrator username: Specify the username for the MySQL server's administrator account.
- Password: Set a strong password for the administrator account.
- Confirm password: Re-enter the password to confirm it.
- Configure networking:
- Connectivity method: Choose the desired connectivity method, such as "Public endpoint" or "Private endpoint."
- Public endpoint: If you choose this option, configure firewall rules to allow access to the MySQL server from specific IP addresses or ranges.
- Private endpoint: If you choose this option, configure a virtual network and a private endpoint to access the MySQL server privately.
- Additional settings:
- Version: Select the version of MySQL you want to use.
- Storage: Configure the storage options for the MySQL server, such as the storage type and capacity.
- Backup: Enable or disable automated backups and configure backup retention settings.
- Advanced: Optionally, configure advanced settings like Azure AD authentication, auditing, or data encryption.

(xii) Review and create: Review the summary of your MySQL server configuration and click on the "Create" button to start the deployment process.

(xiii) Wait for deployment: Azure will start creating the Azure Database for MySQL server based on your settings. The deployment may take a few minutes to complete.

4.6.2. Azure Blob Storage

This application saves every uploaded photo to blob stroge. The main reason for this is that it provides convenience to us in possible system errors, in situations where the application results in low performance and in cases where it must be legally stored. You can create this storage by following the steps below.

(i) Sign in to the Azure Portal: Go to the Azure Portal (portal.azure.com) and sign in using your Azure account.

(ii) Create a new resource: Click on the ”Create a resource” button on the Azure Portal dashboard.

(iii) Search for ”Blob Storage”: In the search bar, type ”Blob Storage” and select it from the list of available resources.

(iv) Click on ”Create”: On the Blob Storage overview page, click on the ”Create” button to start the creation process.

(v) Configure the basic settings:

- Subscription: Select the Azure subscription you want to use for the Blob Storage.
- Resource group: Choose an existing resource group or create a new one to contain the Blob Storage.
- Storage account name: Provide a unique name for your Blob Storage account.
- Region: Select the Azure region where you want to deploy the Blob Storage.

(vi) Configure advanced settings (optional):

- Account kind: Choose between ”StorageV2 (general purpose v2)” or ”BlobStorage (blob storage only)”.
- Performance: Select the desired performance tier (Standard or Premium).
- Replication: Choose the replication option for your Blob Storage (Locally-redundant storage, Zone-redundant storage, Geo-redundant storage, etc.).
- Access tier: Select the access tier for your Blob Storage (Hot, Cool, or Archive).
- Encryption: Configure encryption settings for your Blob Storage (enabled by default).

(vii) Review and create: Review the summary of your Blob Storage configuration and click on the ”Create” button to start the deployment process.

(viii) Wait for deployment: Azure will begin creating the Blob Storage resource based on your settings. The deployment may take a few minutes to complete.

(ix) Access your Blob Storage account: Once the Blob Storage resource deployment is complete, navigate to the Azure Portal dashboard.

(x) Locate your Blob Storage resource: Search for your Blob Storage resource using the search bar or locate it within your resource groups.

(xi) Open Blob service settings: Within your Blob Storage resource, locate and click on the ”Containers” or ”Blob service” section.

(xii) Create a container: Within the Blob service settings, click on the ”Container” or ”Create container” button to create a new container.

(xiii) Configure container settings:

- Name: Provide a unique name for your container. The name must be lowercase and can include letters, numbers, and hyphens.
- Public access level: Choose the desired public access level for the container (Private, Blob, Container or Anonymous).

(xiv) Advanced settings (optional): Depending on your requirements, you can configure advanced settings such as metadata, access policies or CORS (Cross-Origin Resource Sharing) rules.

(xv) Create the container: After configuring the container settings, click on the ”Create” or ”OK” button to create the container.

(xvi) Access and manage the container: Once the container is created, you can access and manage it through the Azure Portal or by using Azure Storage SDKs, APIs, or tools like Azure Storage Explorer.

4.6.3. Azure Form Recognizer

The application uses the Azure Form Recognizer resource to read and make sense of the texts on the slips. It is a service that works with api logic on the basis of source. You can create this resource by following the steps below.

(i) Sign in to the Azure Portal: Go to the Azure Portal (portal.azure.com) and sign in using your Azure account.

(ii) Create a new resource: Click on the ”Create a resource” button on the Azure Portal dashboard.

(iii) Search for ”Form Recognizer”: In the search bar, type ”Form Recognizer” and select it from the list of available resources.

(iv) Click on ”Create”: On the Form Recognizer overview page, click on the ”Create” button to start the creation process.

(v) Configure the basic settings:

- Subscription: Select the Azure subscription you want to use for the Form Recognizer.
- Resource group: Choose an existing resource group or create a new one to contain the Form Recognizer.
- Region: Select the Azure region where you want to deploy the Form Recognizer resource.

(vi) Configure account details:

- Account type: Choose either the ”S1” or ”S2” pricing tier based on your billing options.
- Account name: Provide a unique name for your Form Recognizer resource.

(vii) Review and create: Review the summary of your Form Recognizer configuration and click on the ”Create” button to start the deployment process.

5. TEST AND RESULTS

In this section, we see the test results of the success of Azure Form Recognizer, the success of proposed algorithm, and the success of the Receipt Detector Machine Learning Algorithm.

5.1. Test Systems

During the test, devices at 4 different power levels were used. These :

System 1 (Low Tier)

- Processor: i3-2100
- RAM: 4GB
- Storage: 128GB SSD
- Graphics: Integrated graphics (e.g., Intel HD Graphics)
- Operating System: Windows 10

System 2 (Low - Mid Tier)

- Processor: Intel Core i5-3570
- RAM: 8GB
- Storage: 256GB SSD
- Graphics: Integrated graphics
- Operating System: Windows 10

System 3 (Mid Tier) :

- Processor: Intel Core i7-10700KF
- RAM: 16GB
- Storage: 1TB SSD
- Graphics: Nvidia GeForce GTX 1660
- Operating System: Windows 10

System 4 (High Tier)

- Processor: Intel Core I9-12900Kf
- RAM: 32GB
- Storage: 2TB SSD
- Graphics: Nvidia GeForce RTX 4090
- Operating System: Windows 10

Currently, the most used Windows operating system is Windows 10. For this reason, all tests were performed on Windows 10 Version 22H2 Build: 19045.3031.

5.2. Azure Form Recognizer Test Results

The application usually does not work in the laboratory (ideal conditions) in real-life scenarios. Therefore, it is very important to simulate different environments.A total of 200 receipts were tested under different conditions. These photos and Form Recognizer data were reviewed by 3 different people and the average success value was calculated. Average success rate is %99.5. You can see the distribution of success rate in the table.

Table 5.1. Receipt Analysis System With OCR Success Rate

Rating		
Conditions	**Number of Receipts**	**Success Rate**
Ideal Condition	50	99.7
Low light Condition	50	99.5
High Light Condition	50	99.5
Low Resolution	50	99.3
Total Users	200	99.5

5.3. Proposed Algorithm Test Results

The application usually does not run in the lab (ideal conditions) in real life scenarios. Therefore, it is very important to simulate different environments. A total of 200 chips were tested under different conditions. These photos were checked by 3 people. Even under ideal conditions, it is seen that the success rate is below 50% and the average is 37.75%. This situation is far below the sufficient level for an application that can be used in financial systems.

Table 5.2. Receipt Analysis System With Proposed Algorithm Success Rate

Rating		
Conditions	**Number of Receipts**	**Success Rate**
Ideal Condition	50	45,2
Low light Condition	50	37.8
High Light Condition	50	34.6
Low Resolution	50	33,4
Total Users	200	37.75

5.4. Receipt Detector Machine Learning Algorithm Test Results

In this section, we will examine the success rates of the Receipt Detection Machine Learning Algorithm. Below each figure is written what this figure means.

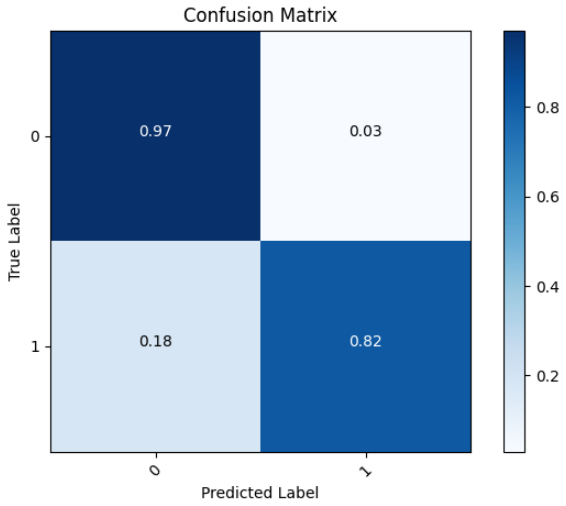

Figure 5.1. Confision Matrix

Confusion Matrix: The Confusion Matrix is a specific table layout that provides a visualization of the performance of an algorithm, typically a supervised learning one. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). It's mainly used for binary or multi-class classification problems, providing a spectrum of performance metrics like precision, recall, F1 score, and support.

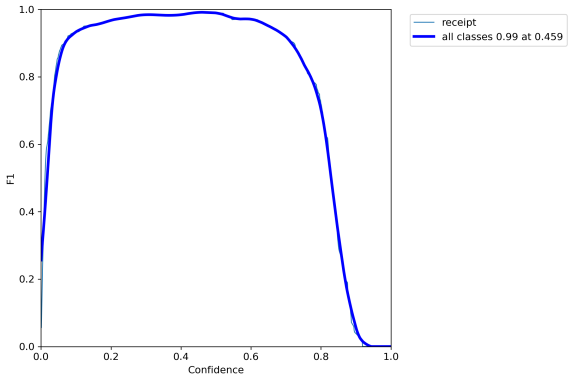

Figure 5.2. F1 Curve Value

F1 Curve Value: The F1 score is the harmonic mean of precision and recall. While precision is the number of correct positive results divided by the number of all positive results, and recall is the number of correct positive results divided by the number of positive results that should have been returned, the F1 score tries to find the balance between precision and recall. A higher F1 score means that you have low false positives and low false negatives, so you're correctly identifying real threats and you are not disturbed by false alarms.

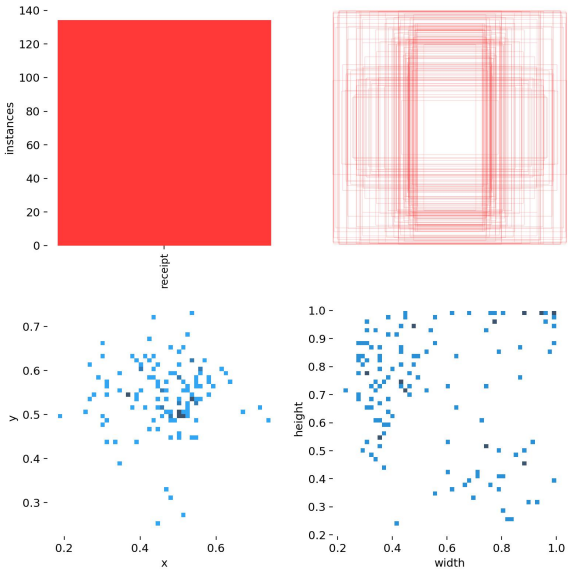

Figure 5.3. Labels Value

Labels Value: Labels are the pre-determined categories data are split into in supervised learning algorithms. In the case of training a machine learning model, each example in the training dataset includes one or more features and a label. The model learns to predict the labels from the features.

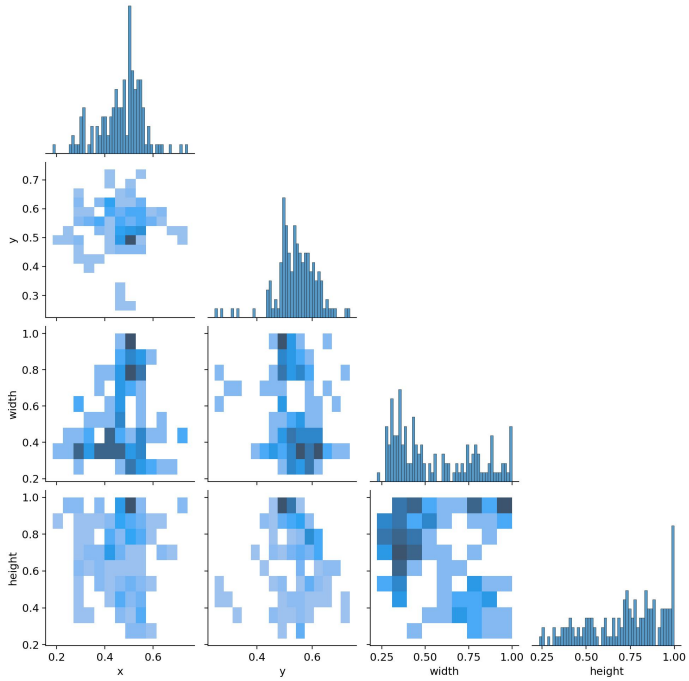

Figure 5.4. Labels Correlogram Value

Labels Correlogram Value: Correlogram is a graphic representation of the correlation matrix, showing the correlation coefficients for different variables. In the case of labels, this can be useful in understanding the relationships and correlation between different labels, which can help with feature selection, model selection, and interpretation of model outputs.

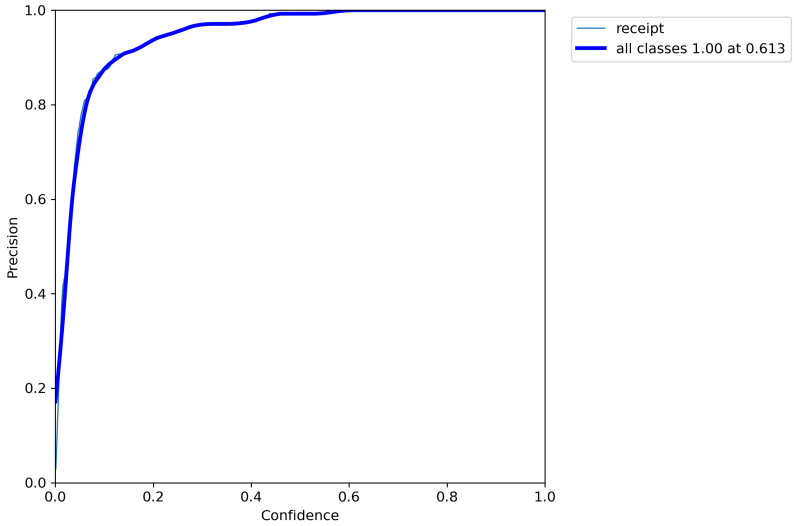

Figure 5.5. P Curve Value

P Curve Value: Precision-Recall is a useful measure of success of prediction when the classes are very imbalanced. The precision-recall curve shows the tradeoff between precision and recall for different threshold.

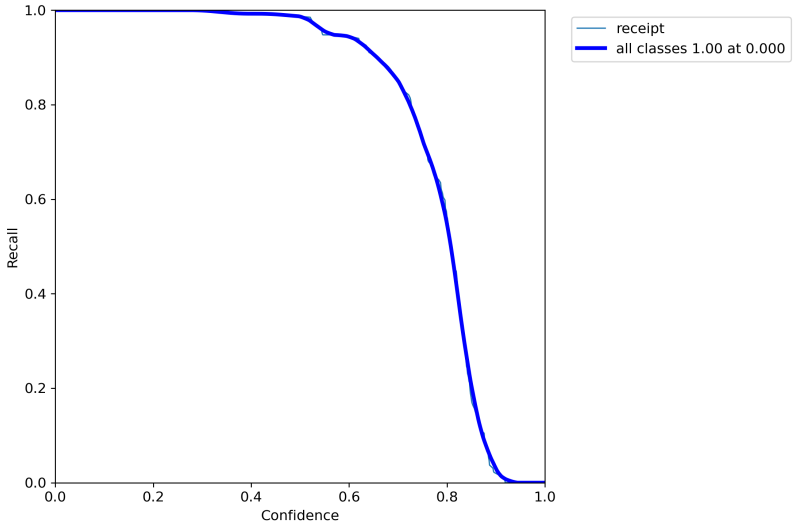

Figure 5.6. R Curve

R Curve: This could refer to a variety of concepts, but often pertains to the Receiver Operating Characteristic (ROC) curve. ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings, making it a popular tool for binary classification problems.

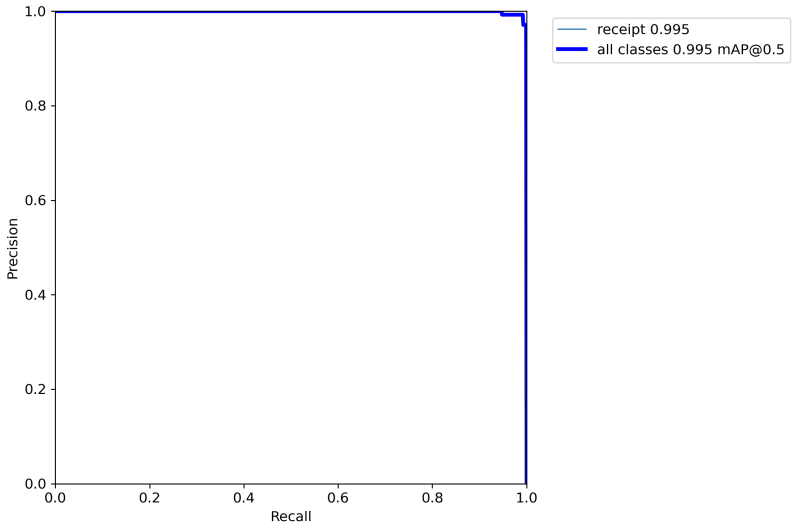

Figure 5.7. PR Curve

PR Curve: Precision-Recall Curve (PR Curve) is a plot of Precision (y-axis) and Recall (x-axis) for different thresholds, much like the ROC curve. A high area under the curve represents both high recall and high precision, indicating a good quality of predicted positives and a small number of false positives.

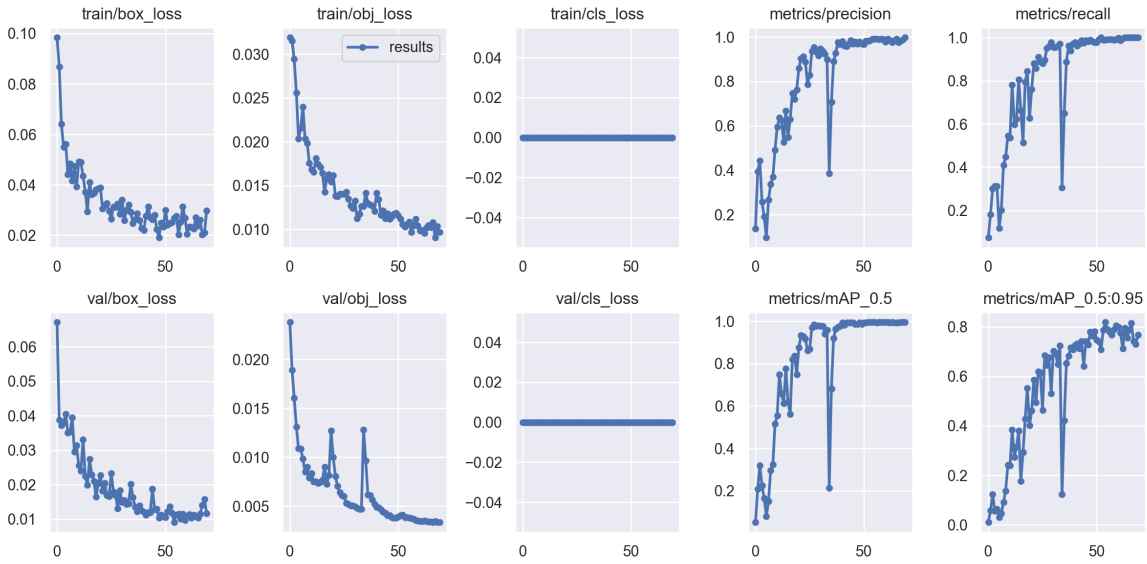

Figure 5.8. Other Informations Receipt Detection ML results

train/box_loss: The loss function used during training to measure the discrepancy between predicted bounding box coordinates and ground truth box coordinates.

train/obj_loss: The loss function used during training to measure the discrepancy between predicted objectness scores and ground truth objectness labels.

train/cls_loss: The loss function used during training to measure the discrepancy between predicted class probabilities and ground truth class labels.

metrics/precision: Precision in machine learning is the fraction of relevant instances among the retrieved instances. It answers the question ”What proportion of positive identifications was actually correct?” A model with high precision indicates fewer false positives.

metrics/recall:Recall is the fraction of the total amount of relevant instances that were actually retrieved. It answers the question ”What proportion of actual positives was identified correctly?” A model with high recall indicates fewer false negatives.

val/box_loss: The loss function used during validation to measure the discrepancy between predicted bounding box coordinates and ground truth box coordinates.

val/obj_loss: The loss function used during validation to measure the discrepancy between predicted objectness scores and ground truth objectness labels.

val/cls_loss: The loss function used during validation to measure the discrepancy between predicted class probabilities and ground truth class labels.

metrics/mAP_0.5: This is the Mean Average Precision at an Intersection over Union (IoU) threshold of 0.5. IoU is a measure of the overlap between two bounding boxes. mAP_0.5 is used as a metric for object detection models, where a higher value indicates better model performance.

metrics/mAP_0.5:0.95: This refers to the mean Average Precision over a range of IoU thresholds from 0.5 to 0.95 (with step size 0.05). It averages the model's performance at various levels of tolerance for bounding box overlap, providing a more robust measure of its performance across a range of scenarios. This is often used in evaluating models for tasks like object detection in computer vision.

6. CONCLUSION

The application works with high accuracy and can be easily integrated into different systems thanks to the simplicity of its architecture or can be customized thanks to its simple structure. These customizations are not limited to just the logo. This system can be easily integrated into 3rd party software such as Teams or software for the company.

Thanks to this program, the following transactions in accounting processes are reduced:

- Human errors
- Delays
- Unnecessary costs

This situation increases the satisfaction of the employees and makes the company financially profitable.

6.1. Future Work

Ready-made connectors and API-based version of the program can be designed. The GUI can be beautified as needed. Apart from these, the following technical features can be added:

- Customer Relationship Management (CRM) Integration
- Enterprise Resource Planning (ERP) Integration
- Business Intelligence (BI) Integration
- Collaboration Tools Integration
- Handwritten receipt/invoice support
- Voice command feature for visually impaired employees
- Invoice support
- A mechanism that allows us to categorize the products on the receipt
- MacOs Support
- Linux Support
- Android Support
- IOS Support
- Improved Accuracy
- Better Error Management
- Compatibility with Various Data Formats
- Compatibility with More Image Formats
- Language Support
- Performance Optimization
- Graphical User Interface (GUI) Development
- Enhanced Automation
- Improved Data Security

- Data Privacy
- Scalability Improvements
- Cross-platform Support
- Machine Learning Integration(Before the send to Form Recognizer check image is contained receipt)
- Real-time Processing
- Voice Recognition Features
- 2 Factor Authentication
- Biometric Authentication
- Multi-Factor Authentication
- IoT Integration
- Real-Time Analytics
- Reporting Features
- Augmented Reality Integration
- AI-Powered Predictive Analytics
- Advanced Searching Algorithms
- Advanced Filtering
- Enhanced User Experience
- Role-Based Access Control
- Logging and Monitoring Features
- Email Notifications
- SMS Notifications
- Push Notifications
- Chatbot Integration
- Workflow Automation

- User Customization Features
- Dark Mode
- Offline Support
- Multithreading
- GPU Acceleration
- Personalized Recommendations
- User Feedback Mechanism
- Multilingual Support
- Integration with Voice Assistants
- Encryption Techniques

Bibliography

[1] G. Kim, T. Hong, M. Yim, *et al.*, "Ocr-free document understanding transformer," *arXiv preprint arXiv:2111.15664*, 2021.

[2] S. Park, S. Shin, B. Lee, *et al.*, "Cord: A consolidated receipt dataset for post-ocr parsing," in *Workshop on Document Intelligence at NeurIPS 2019*, 2019.

[3] H. Zhang, E. Whittaker, and I. Kitagishi, "Extending trocr for text localization-free ocr of full-page scanned receipt images," *arXiv preprint arXiv:2212.05525*, 2022.

[4] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, IEEE, vol. 2, 2007, pp. 629–633.

[5] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[6] *Accountant salary in united states*, `https://www.indeed.com/career/accountant/salaries`, Accessed: 2023-06-11.

APPENDIX A: PSEUDO CODE

(i) Open Login page

(ii) Get login information from user

(iii) If true, open the main menu

(iv) Get an image to process from user

(v) Upload this image to the blob file

(vi) Send blob url to azure form recognizer

(vii) Parse the incoming reply

(viii) Show the parsed reply to the user

(ix) If the user presses the send to database button, send the information to the relevant database.

(x) Keep main menu open for next receipt image

APPENDIX B: USER GUIDE

This user guide explains how users should use the application. The image of each stage is available just below.

Step 1 Login by entering your e-mail address and password on the login screen

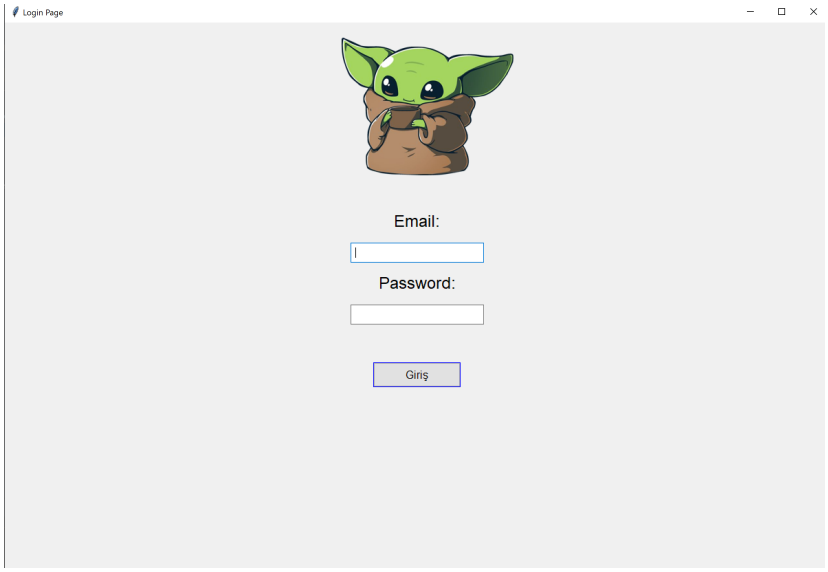

Figure B.1. User Manual Step 1

Step 2 Select the receipt photo you want to process by clicking the Choose file button.

Figure B.2. User Manual Step 2

Step 3 Send the receipt image you selected to the Form Recognizer for processing by clicking the Send to Form Recognizer button. At the end of the process, you will be able to see the lines on the receipt on the screen.

Figure B.3. User Manual Step 3

Step 4 Click the Save to Database button. This button sends the information on the receipt to be saved to the database.

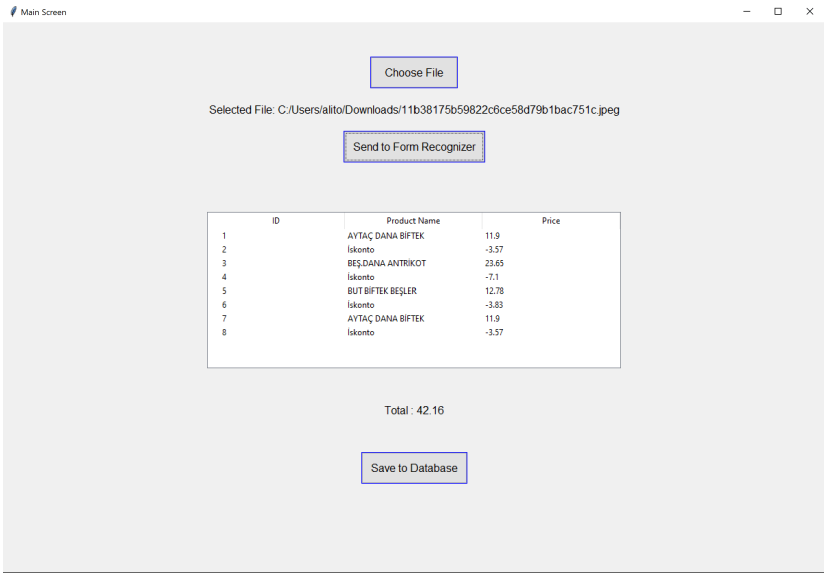

Figure B.4. User Manual Step 4