TED UNIVERSITY

2023 SPRING

CMPE 492 Senior Project II

Final Report

Name of the Project: Receipt Analyzer

The URL of the webpage: https://bzelihagunay.wixsite.com/receipt-analyzer

Team Members' Name: Burçak Zeliha Günay, Ahmet Berat Akdoğan, Başar Aslan, Hüseyin Hikmet Fındık

Section: 01

Supervisor's Name: Aslı Gençtav

Jury Members' Name: Venera Adanova, Gökçe Nur Yılmaz

Date of Submission: 05/06/2023

Table of Contents

1. Final Architecture and Design of the System

1.1 System Overview

The Receipt Analyzer system is an Android-based mobile application designed to analyze market receipts and extract text information from them. The system utilizes Firebase Realtime Database for data storage and synchronization. The Receipt Analyzer system adopts a client-server architecture. This architecture provides a structured and modular approach to development, separating the concerns of data processing, user interface, and data storage.

1.2 Client-Side Architecture

The client-side architecture of the Receipt Analyzer system consists of three main interfaces:

1.2.1 Home Page UI: The Home Page UI is the initial screen that users encounter upon opening the mobile application. It displays a simple welcoming message and a capture button. This interface is implemented using the **activity_main.xml** file.

1.2.2 Crop Page UI: After the user presses the capture button, they are presented with the Crop Page UI. This interface allows users to select an image from the gallery or capture an image using the device's camera. It also includes a crop button for extracting the necessary part of the receipt image. The Crop Page UI is implemented using the **activity_main.xml** file.

1.2.3 Helper UI: Once the image is successfully cropped and scanned, the Helper UI is displayed. This interface shows the detected text on the screen and provides two helper buttons. One button allows users to repeat the capture process, while the other button enables the transmission and writing of data to the Firebase Realtime Database. The Helper UI is implemented using the **activity_main.xml** file.

1.3 Server-Side Architecture

The server-side architecture of the Receipt Analyzer system consists of the following components:

1.3.1 Backend Structure: The core backend functionality is implemented in the **MainActivity.java** class. This class handles tasks such as photo taking, cropping, and extracting desired text from the receipt image. It establishes a connection to the Firebase Realtime Database and saves the extracted data. The **MainActivity.java** class interacts with various UI elements defined in the XML files.

1.3.2 Storage: The **Users.java** class serves as a helper class for constructing the scanned texts line by line with child nodes in the Realtime Database. It contains attributes and methods for storing and retrieving text information.

1.3.3 Firebase Realtime Database: The Firebase Realtime Database is used as the storage mechanism for the Receipt Analyzer system. It provides a cloud-hosted, NoSQL database for storing and synchronizing the detected text data. The data synchronization occurs in real-time, ensuring that the mobile application remains up to date with the latest information.

1.4 Integration and Communication

The client-side interfaces communicate with the server-side components through appropriate method calls and event handling. The **MainActivity.java** class utilizes Firebase SDK and dependencies to establish a connection with the Firebase Realtime Database. The extracted text information is written and saved in the database using the appropriate methods provided by Firebase.

1.5 System Workflow

The workflow of the Receipt Analyzer system is as follows:

- Upon launching the application, users are presented with the Home Page UI.
- After pressing the capture button, the Crop Page UI allows users to select or capture an image of the receipt.
- The captured image is then cropped using the selected area.
- The cropped image is processed to extract the text from the receipt using OCR techniques.
- The extracted text is displayed in the Helper UI, along with helper buttons.
- Users can repeat the capture process or transmit the extracted data to the Firebase Realtime Database by clicking the respective buttons.

Overall, the Receipt Analyzer system provides a user-friendly interface for capturing, cropping, and extracting text from market receipts. The Firebase Realtime Database ensures efficient data storage and synchronization for a seamless user experience.

2. Final Status of Receipt Analyzer Project

The Receipt Analyzer project is a comprehensive and innovative solution designed to automate the process of receipt analysis. This final report provides a detailed overview of the project, highlighting its successful completion and the key achievements in terms of software development, functionality, and performance. The Receipt Analyzer project has successfully addressed the challenges associated with receipt processing, data extraction, and analysis, offering users an efficient and user-friendly tool for managing and analyzing receipts.

The Receipt Analyzer project aimed to develop a mobile application that utilizes advanced computer vision and natural language processing techniques to extract relevant information from receipts. The primary objective was to provide users with a convenient and efficient way to organize, categorize, and analyze their receipt data.

2.1 Key Features and Functionality

The Receipt Analyzer project offers a range of features and functionalities, including:

- Market selection options to cater to different grocery store brands.
- Image capture or gallery selection for receipt input.
- Image processing and cropping to extract relevant receipt information.
- OCR (Optical Character Recognition) for accurate text extraction.
- Real-time text display and feedback through the HelperUI.
- Integration with Firebase Realtime Database for secure and synchronized data storage.
- Visualization of price changes over time through the TimePriceGraphUI.

2.2 Achievements and Impact

The completion of the Receipt Analyzer project has resulted in several significant achievements. The Receipt Analyzer project has achieved the following milestones:

- Image Processing and Text Extraction: Accurate and efficient extraction of text from receipt images using advanced image processing and OCR techniques.
- Data Storage and Synchronization: The integration with the Firebase Realtime Database enables seamless storage and real-time synchronization of extracted text, price, and date information across multiple devices.
- Chart Visualization: The chart allows users to visualize price and date trends from analyzed data in an intuitive line chart format. Provision of valuable insights and analytics through the TimePriceGraphUI.
- User Interface: The user interface of the Receipt Analyzer application is designed to be user-friendly, facilitating easy receipt capture and analysis.

2.3 Challenges and Future Prospects

The Receipt Analyzer project has immense potential for further enhancements and future development.

- Image Quality and Variation: Ensuring consistent and accurate text extraction from diverse receipt formats and varying image qualities required continuous improvement and fine-tuning of the image processing algorithms.
- OCR Accuracy: The accuracy of optical character recognition (OCR) can be influenced by factors like font styles, unusual characters, or handwritten text on receipts. Future enhancements could focus on improving OCR accuracy.
- Enhanced Analysis Features: The project could be extended to include more advanced analysis features, such as categorization of expenses, extraction of additional metadata, and integration with external accounting systems.
- Market Expansion and Receipt Syntax Learning: In the future, the Receipt Analyzer project aims to increase the number of markets that can be selected on the homepage. The program is designed to learn and adapt to the receipt syntax of different markets. By incorporating support for various market receipts, the system can provide a more comprehensive analysis of purchases made across different markets.
- Consolidated Price-Date Charts: With the ability to process receipts from different markets, the Receipt Analyzer project envisions creating consolidated price-date charts. By extracting information from receipts with varying syntaxes, the system can identify the same products purchased from different markets and present them on the same chart. This feature enables users to visualize and compare prices and date trends for identical products across different markets, providing a more comprehensive overview of their spending patterns.

3. Impact of Engineering Solutions in Receipt Analyzer Project: A Global, Economic, Environmental, and Societal Perspective

The Receipt Analyzer project aimed to develop advanced engineering solutions to automate receipt analysis. This part highlights the impact of the engineering solutions developed in the Receipt Analyzer project from a global, economic, environmental, and societal standpoint. The Receipt Analyzer project, with its innovative software architecture and functionalities, has brought significant benefits to various stakeholders and has the potential to positively influence a global scale.

3.1 Global Impact

- Accessibility: The mobile application enables users worldwide to easily manage and analyze their receipts, contributing to a more organized financial management process.
- Scalability: The modular software architecture allows for easy scalability, accommodating an increasing number of users and supporting diverse market brands globally.

3.2 Economic Impact

- Time and Cost Savings: By automating the receipt analysis process, users can save valuable time and effort, enabling them to focus on more productive activities.
- Financial Management: The comprehensive analysis provided by the system assists users in optimizing their spending, budgeting, and financial decision-making, potentially leading to improved financial stability.

3.3 Environmental Impact:

- Paper Reduction: By digitizing receipts and promoting electronic record-keeping, the project reduces the need for paper-based receipts, thus minimizing deforestation and the carbon footprint associated with paper production and disposal.
- Energy Efficiency: By utilizing mobile devices and cloud-based storage, the project reduces the energy consumption associated with traditional receipt management methods.

3.4 Societal Impact

- Financial Empowerment: The system empowers individuals to take control of their finances, promoting financial literacy and independence.
- Transparency: By providing users with a comprehensive overview of their spending habits and price fluctuations, the system promotes transparency in consumer transactions.
- Data Insights: Aggregated and anonymized data from the project can be analyzed to identify broader market trends and consumer behavior, potentially benefiting policymakers and businesses.

4. Contemporary Issues in Receipt Analyzer Project

The Receipt Analyzer project has encountered several contemporary problems that require attention and resolution. These issues primarily revolve around the accurate extraction and organization of data when dealing with receipts containing multiple products. Let's delve into these problems in detail:

4.1 Challenges in Receiving Multiple Products

One of the significant issues faced by the Receipt Analyzer project arises when dealing with receipts that contain more than one product. While OCR successfully reads all the products as text, difficulties arise when attempting to separate and tag these products into distinct categories such as date, price, and product name within the database. This results in complications during the data structuring phase, hindering accurate analysis, and reporting of individual products on the receipt.

4.2 Incomplete Date Representation in Graphs

Another problem manifests when generating graphical representations for receipts with multiple data. When a receipt includes only one product, the system accurately extracts and presents the price data on the graph. However, when the same product has previous date and price data, it encounters difficulties in correctly incorporating different dates on the X-axis of the graph. This limitation undermines the visual representation of product-specific purchasing trends over time, potentially hindering users' ability to analyze and compare historical data effectively.

4.3 Occasional OCR Text Extraction Errors

The Receipt Analyzer project occasionally encounters OCR text extraction errors, albeit rarely. These errors manifest in various ways, such as the addition of unnecessary spaces, failure to detect existing spaces, or misinterpretation of certain characters, leading to partial misunderstandings of the product names. While the OCR technology used in the project generally performs well, these sporadic errors contribute to data inaccuracies and can impact the overall reliability of the system.

Resolving these contemporary problems requires focused attention and targeted solutions. Here are some potential approaches to address these issues:

1. Enhanced Data Parsing and Tagging Algorithms: The project can benefit from the development of more advanced algorithms and techniques for parsing and tagging data from receipts with multiple products. By leveraging pattern recognition methods, the system can accurately identify and categorize individual products' relevant information, such as dates, prices, and product names, facilitating comprehensive data organization.

2. Improved Charting Mechanisms: To tackle the challenges in charting multiple product purchases, enhancements to the charting functionality should be implemented. The system can be updated to handle and visualize multiple price values associated with the same product effectively. This may involve developing algorithms to aggregate and present price variations over time clearly and concisely, ensuring accurate representation on the charts.

3. OCR Error Correction and Validation: To mitigate OCR text extraction errors, the project can incorporate error correction and validation mechanisms. These mechanisms can analyze the extracted text, compare it with reference data, and apply corrective measures such as spell-checking, data normalization, and context-based validation. Additionally, implementing user feedback loops to report and rectify OCR errors can contribute to continuous improvement and accuracy enhancement.

By addressing these contemporary problems, the Receipt Analyzer project can further enhance its accuracy, reliability, and usability. Resolving challenges related to tagging multiple products, improving charting functionalities, and minimizing OCR text extraction errors will contribute to a more robust and dependable system.

5. New Tools and Technologies Used

- Android Studio IDE was used as the software development environment and Java programming language was used as the supported language.
- To detect the texts on the market receipt to the system, OCR technology was used. In this way, the data is obtained by the Optical Character Recognition system.
- As a Firebase solution, Realtime Database was used to store the extracted data and reuse it in the application to plot the graph. Since this database does not actually contain SQL, a live database with a key-value system has been experienced without using any queries.
- A graph with x and y coordinates corresponding to the date and price values obtained by OCR is presented to the user. In this context, the MPAndroidChart library is used.

6. Use of Library Resources and Internet Resources

For the library resources, in order to plot the graph at the end of the implementation process, it benefited from some helpful GitHub links to add dependencies also in order to make the connection of the Firebase to the Android Studio project, helpful libraries are added and updated during the implementation of the project. Various Java classes such as MediaStore, CropImage, Bitmap, FirebaseDatabase, DatabaseReference, LineData, Bindings are used to get the image from the gallery of the mobile application, perform OCR operation, connection of database to the application, and plotting graph to display on the UI. **For the internet resources**, the official documentation of the Firebase Realtime Database is used to understand the main concept of the working mechanism of the database. By using this document, sending, and retrieving the data successfully accomplished without overwriting the data and data loss. The crop and OCR mechanism, as well as the basic UI of the mobile application are implemented with the help of YouTube videos and online resources such as GeeksForGeeks, StackOverflow, etc.

7. Test Results

Referring to the Test Plan Report, **Test Case 1**, the process from starting the program from scratch to the end of the crop process, is successfully executed and passed without any bugs or errors.

In the **Test Case 2**, the control process and test of whether the data after the crop, display to the phone screen without data loss and whether the data is successfully saved to the database with a single button or not, is tested. It has been observed that the test is concluded successfully when the desired type of crop operation is done in a certain order of date, product name and price as well as there is no data loss in the detected parts of the receipt. That means that if the crop operation is not performed perfectly, there have been cases where the test failed. For instance, in the below screenshot of the mobile application, since crop rectangle is not started from the date part of the receipt and not ended with the product name, it is observed that the detected text is not accurate enough.

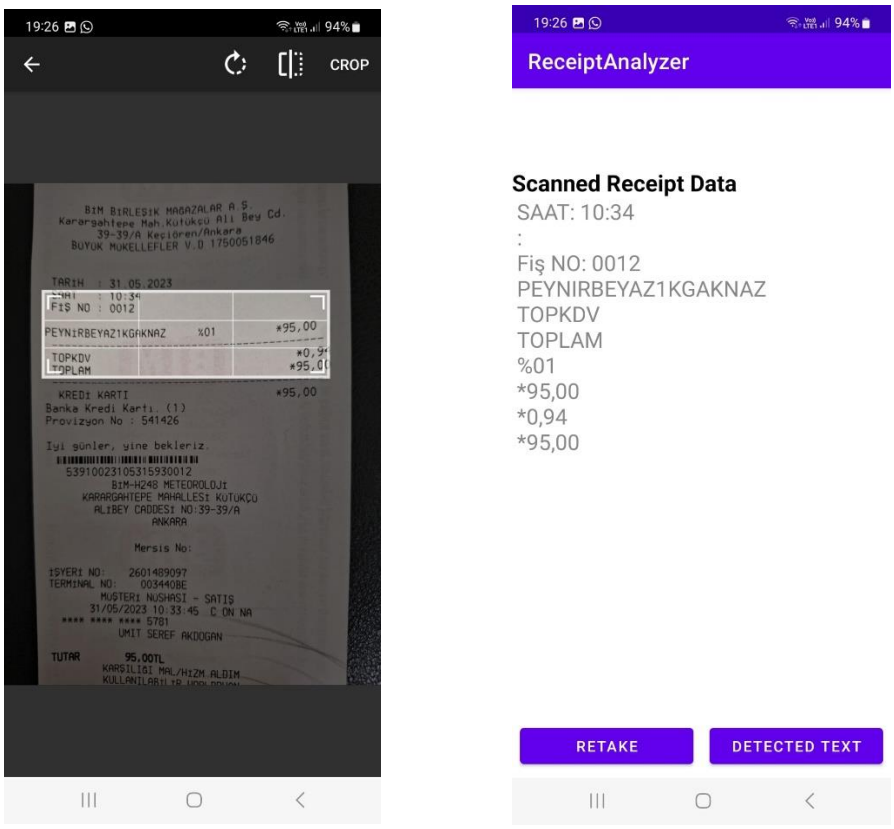

Figure 1. Failed Test Scenario.

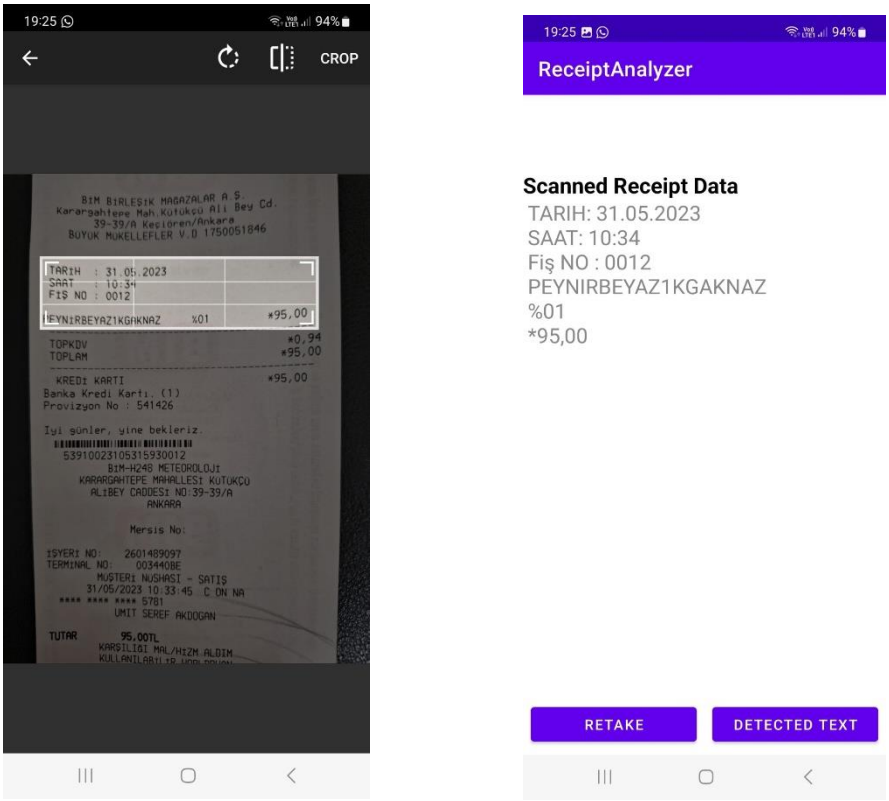

Figure 2. Passed Test Scenario.

On the **Test Case 3,** charting and plotting the obtained data. Checking the correctness of the elements on the X-axis and Y-axis of the graph. In of the scenarios, tests are observed as failed, in some of the test cases, it is tested as passed. For instance, it is observed that when the price and date data is entered manually to the graph before the crop operation, all data were observed in a time price graph flawlessly, however, when the crop operation is implanted repeatedly, the data which is coming from the receipt is lost and last data is overwriting that data entered previously.

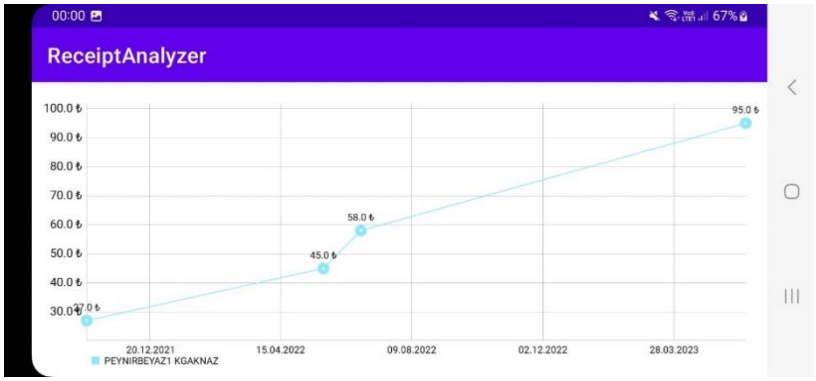

Figure 3. First Test Scenario of plotting graph.

In this scenario, overwriting the cropped data problem is solved, which means user can be able to take the receipt photo and crop the image repeatedly. In this way, it is tested as passed scenario for the scanned receipt data is plotted on the graph dynamically. However, this time while the y-axis (price) is dynamically growing without any error, the x-axis (date) overwrites as a last detected data without showing the previous date data. It was determined as a failed test that the date part of the graph in the x-coordinate was formatted incorrectly.

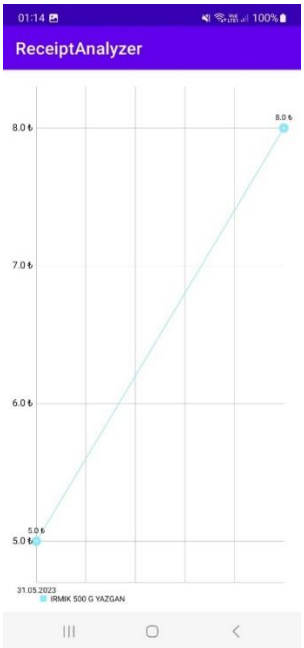

Figure 4. Second Test Scenario of plotting graph.

8. References

Firebase ML Kit - Recognize Text in Android:

Firebase. (n.d.). Recognize Text in Android. Retrieved June 5, 2023, from https://firebase.google.com/docs/ml-kit/android/recognize-text?hl=en#2.-extract-text-from-blocks-of-recognized-text

MPAndroidChart GitHub Repository:

PhilJay. (n.d.). MPAndroidChart. Retrieved June 5, 2023, from https://github.com/PhilJay/MPAndroidChart

Firebase Realtime Database - Getting Started on Android:

Firebase. (n.d.). Getting Started on Android. Retrieved June 5, 2023, from https://firebase.google.com/docs/database/android/start?hl=en

Android Image Cropper GitHub Repository:

ArthurHub. (n.d.). Android-Image-Cropper. Retrieved June 5, 2023, from https://github.com/ArthurHub/Android-Image-Cropper

Google ML Kit Vision - Text Recognition (Version 2):

Google. (n.d.). Text Recognition. Retrieved June 5, 2023, from https://developers.google.com/ml-kit/vision/text-recognition/v2?hl=tr