Table of Contents

[Abstract 2](#_Toc163649900)

[Introduction 2](#_Toc163649901)

[Problem statement 2](#_Toc163649902)

[Background 2](#_Toc163649903)

[Aims 2](#_Toc163649904)

[Objectives 3](#_Toc163649905)

[Literature review 3](#_Toc163649906)

[Receipt Extractor 3](#_Toc163649907)

[Paper 1 - Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms. 3](#_Toc163649908)

[Paper 2 - Information Extraction from Scanned Invoices using Machine Learning, OCR, and Spatial Feature Mapping Techniques 5](#_Toc163649909)

[Paper 3 - Computer Vision for Document Image Analysis and Text Extraction 7](#_Toc163649910)

[Mobile Application 8](#_Toc163649911)

[Paper 4 - React Native vs Flutter, cross-platform mobile application frameworks. 8](#_Toc163649912)

[Conclusion 10](#_Toc163649913)

[Requirements specification 10](#_Toc163649914)

[Receipt Section Detection 10](#_Toc163649915)

[Text Extraction 10](#_Toc163649916)

[Format and Classify text extraction. 11](#_Toc163649917)

[User Interface 11](#_Toc163649918)

[Design 11](#_Toc163649919)

[Server 12](#_Toc163649920)

[Receipt extraction feature. 12](#_Toc163649921)

[User Correction Feature. 13](#_Toc163649922)

[Server Communication 13](#_Toc163649923)

[Database 13](#_Toc163649924)

[Application 14](#_Toc163649925)

[Receipt upload for receipt extraction. 14](#_Toc163649926)

[List Receipt Historical 14](#_Toc163649927)

[Analytics 14](#_Toc163649928)

[Implementation 15](#_Toc163649929)

[Server 15](#_Toc163649930)

[Docker Usage 15](#_Toc163649931)

[Makefile 15](#_Toc163649932)

[Receipt Section Detection Model (YOLO) 15](#_Toc163649933)

[Classification/Formatting 15](#_Toc163649934)

[Mobile Application 15](#_Toc163649935)

[Testing 15](#_Toc163649936)

[Result & Analyzing 15](#_Toc163649937)

[Conclusion 16](#_Toc163649938)

# Abstract

# Introduction

## Problem statement

This project of ‘smart receipt management and Extraction’ aims to enhance the efficiency of company accounting processing, which “plays a significant role in the effective management process” (Alabdullah, 2019). A big part of this accounting process consists of recording varied expenses and, for numerous reasons, has always been a manual process. This repetitive process requires a significant amount of time and energy. Therefore, this project aims to develop a solution to this problem: How to facilitate the receipt logging process?

## Background

As previously mentioned above in the problem statement, the purpose of this project is to facilitate the recoding process of receipts. The problems that arise from this manual receipt logging task, as said before, requires individuals to log the different receipts one by one manually, which significantly uses a lot of time and energy. But, by automating this process, enabling this task of receipt logging to be digitalized, it could considerably assist the users in this task, not just by reducing the amount of energy used to complete the process, but also valuable time. Concurrently, there are solutions already in existence across the market, such has “Recipator AI” (AI, no date) or “Veryfi” (*Transform Documents into Actionable Data in Seconds using Veryfi OCR API*, no date) but these applications are limited in functionalities and are not adapted to the Mauritian market, lacking an easy-to-use system for individuals. Moreover, the digitalization of this process of receipt handling would improve the ecological issue of receipt storage and production, thereby reducing the ecologic and environmental impact of this task, as it currently produces a high amount of paper waste and enlarges business’ carbon footprint. In summary, this project aims to tackle the specific task of handling and managing receipts through a user-friendly system using the latest state-of-the-art technologies.

## Aims

The project “Smart Receipt Management and Extraction” system is designed to streamline and simplify the repetitive and time-consuming task of handling, organizing and informatize receipts, whether physical or digital. Therefore, the primary purpose of this project is to serve a comprehensive receipt extraction and management system, offering the user a set of features allowing for a user-friendly user-interface experience.

## Objectives

Literature Review

Gather and analyze similar projects and research literature to examine their technologies used, compare their advantages and disadvantages, and to appropriately apply the state-of-the-art technologies from these papers to the different elements of the current system.

Receipt extraction model

Develop a receipt extraction model which will be used to efficiently extract the data from multiple variations of receipts, offering the user the wanted receipt extraction automation to facilitate the receipt logging process.

Receipt dataset

Gather a large quantity of receipt for the dataset to have a precise, accurate and efficient model for extraction of the user’s receipts as effectively possible, and for this dataset to be used for the training and testing purposes of this receipt prediction model.

Mobile Application

Develop a Mobile application of front-end of the project, to be used by the users as a gateway for the system, which will be partially in a server.

Local Server

Develop a local server which will be used to host the extraction model and the database. It will contain all the “heavy” process need by the system.

System Test

Create tests for the system and its different components, such as the mobile application, the server, the communication between them and the different features which will be provided to the user.

# Literature review

## Receipt Extractor

### Paper 1 - Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms.

This paper from Joel Odd and Emil Theologou (2018) is a study that “investigated if it was feasible to use machine learning tools on OCR [Optical Character Recognition] extracted text data to classify receipts and extract specific data points”. This process firstly extracts the receipt data through different Optical Character Recognition (OCR), then classifies through a model into different categories. They have tested different technologies for the OCR extraction and the model prediction, all listed below in *Table 1*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Technologies** | **Pros** | **Cons** |
| **Optical Character Recognition (OCR)** | Azure Computer Vision API | * Good accuracy in text recognition. | * The security is uncontrollable due to third party dependance. * May require significant data preprocessing. |
| Google Cloud Vision API | * Good performance and time response. | * The security is uncontrollable due to third party dependance. * Managing variations in receipt formats can be complex |
| Tesseract OCR | * Deployable locally * Not dependent of third party. * Strong community | * Varying receipt formats may affect accuracy. * Accuracy depends on image quality. * Slow process for large amount of data. |
| **Machine Learning Models** | Linear Support Vector Classification (LinearSVC) | * 94% accuracy achieved. * Efficient with large text dataset | * Not has efficient for non-linear data relationship. * Require optimal parameters tuning, otherwise reduction in performance. |
| Multilayer Perceptron Classifier (MLPClassifier) | * Can capture complex relationship (non-linear). * Flexibility with multiple parameters. | * May be computationally intensive as the model become complex. * Risk of overfitting if the parameters are not well set. |
| Naive Bayes Classifier | * Easy implementation to handle large dataset. * May be use has baseline for classification problem. | * Can make naïve assumption which will lead to a reduction in accuracy for complex data. * Training dataset balance strongly influence the accuracy. |

Table 1 - Advantage and Limitation of technology used in "Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms" written by Odd and Theologou

From the OCR, three principal technologies are tested “Azure Computer Vision API” provided by Microsoft, “Google Drive REST API” provide by Google, and “Tesseract OCR”. The “Azure Computer Vision API” such has “Google Drive REST API” are efficient and powerful OCR tools but are third party dependent and all their processing power are deported in their own servers. This deported strategy allows powerful OCR but creates delay using API to upload and download the data and create a dependence to their services and could be costly. On the other hand, “Tesseract OCR” is an open-source OCR which can be deployed locally and, therefore, have a quicker response time compared to “API” OCR. In the context of our project, the user will have the ability to correct any error from the extraction, therefore a quick response time is essential for a streamline user-experience and is preferable over a slice reduction of extraction accuracy.

After extracting the text from the receipt, the output is categorized through models. Different models were used, such as the Linear Support Vector Classification, the Multi-layer Perceptron classifier, and the Naive Bayes Classifier, which all have their individual advantages and disadvantages, which are shown in *Table 1*. Their model is based on a text extract before using the model strategy, and therefore, is not used for image recognition.

### Paper 2 - Information Extraction from Scanned Invoices using Machine Learning, OCR, and Spatial Feature Mapping Techniques

The second research paper “Information Extraction from Scanned Invoices using Machine Learning, OCR and Spatial Feature Mapping Techniques” is a project focusing one extracting information from scanned invoice using different technologies for different step (Darsha, 2023). In the context of our receipt extractor project, we will focus on the receipt detection/classification and the text extraction step only.

For the Optical Character Recognition, like in the first paper in the section *Paper 1 - Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms.*, similar technologies were used, such as Tesseract OCR and Google Cloud Vision API. Since we have already discussed the advantages and disadvantages of these technologies, there is no need to re-analyze them. There it shows that these technologies are the state of the art in their domain and prove their efficiency.

As for the text detection and classification, it uses the YOLO (You Only Look Once) model, which is a state-of-the-art object detection application provided by Ultralitics in python, that is shown in *Table 2*. The YOLO model is well known for its rapid speed in prediction, while having a low background mistake, which is an important characteristic for a receipt extraction application wherein the picture would be taken quickly from a mobile phone. However, it also has its own limitations such as the trade-off between speed and accuracy, which could cause problems in case of too low accuracy. Also, the processing could be resource-intensive for training and prediction.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Technologies** | **Pros** | **Cons** |
| **Object detection model** | YOLO version 5 | * Good accuracy in detecting and classifying multiple objects. * Fast inference time and global speed. * Reduction of background error by processing all picture | * Require extensive computation power for many classes. * Potential trade-off between speed and accuracy. * Less densely pack data may reduce accuracy. |
| **Optical Character Recognition (OCR)** | Tesseract OCR | See *Table 1* | |
| Google Cloud Vision API | See *Table 1* | |
| Convolutional Neural Networks (CNNs) | * Good at extracting hierarchical feature from images. * Versatile for a wide range of image recognition. | * High computational resources for training. * Risk of overfitting training data. * Need a lot of tuning and optimization for optimal performance. |

Table 2 - Advantage and Limitation of technologies used in "Information Extraction from Scanned Invoices using Machine Learning, OCR and Spatial Feature Mapping Techniques" written by Darsha.

### Paper 3 - Computer Vision for Document Image Analysis and Text Extraction

The third paper “Computer Vision for Document Image Analysis and Text Extraction” is a research article aiming to improve Optical Character Recognition (OCR) systems, particularly for image processing (Benchekroun, 2022). It explores multiple technologies, such has Convolutional Neural Network (CNN) + Long Short-Term Memory (LSTM) Network for feature extraction from images, Deep CNN + Transformer/seq2seq Network to handle sequential data, Generative Adversarial Networks (GANs) to generate synthetic training data, and Morphological Operations for preprocessing training data with different operation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Technologies** | **Pros** | **Cons** |
| **Optical Character Recognition (OCR)** | Convolutional Neural Networks (CNN) + Long Short-Term Memory (LSTM) | * Effective for non-handwritten text. * Shows significant accuracy improvement with synthetic data. | * Are computationally intensive and may require large dataset for high accuracy. * Struggle with complex patterns in data |
| Deep CNN + Transformer/seq2seq Network | * Capable of parallel processing, offering faster execution. * Can extract complex features from images. | * Require significant computation resources. * Difficulty to optimize due to his complexity. |
| **Synthetic Data Generation** | Generative Adversarial Networks (GANs) | * Increase training data. * Help model learn feature that may not be present in real-world scenario. | * Generated data might not always represent real-world scenarios. * Can reduce accuracy in case of generated data not accurate to the scenario. |

Table 3- Advantage and Limitation of technologies used in "Computer Vision for Document Image Analysis and Text Extraction" written by Benchekrou.

The advantages and disadvantages of these technologies can be found in *Table 3*. Furthermore, the usage of GANs to create synthetic data is an interesting feature to improve the training dataset and, therefore, increases the accuracy if an OCR algorithm is developed from scratch, such as the different ones used in this third research paper. Developing an OCR from scratch without using an already made system, such has Tesseract or Google Cloud Vision API, could allow for a more precise extraction especially for Mauritian receipts, thereby it will increase the development time, but it will be limited by its training data, which in out context, is limited and, therefore, would not be recommended compared to the powerful pretrained system.

## Mobile Application

### Paper 4 - React Native vs Flutter, cross-platform mobile application frameworks.

The last research paper “React Native vs Flutter, cross-platform mobile application frameworks” focuses on comparing two frameworks to develop cross-platform mobile application: React Native and Flutter (Wu, 2018). These two frameworks are prominent factors in mobile development, and their advantages and disadvantages are available in *Table 4*. React Native, which was developed by Facebook, is using JavaScript and React, which are famous programing languages. However, for complex development, it will require a specific programming language per platform. On the other hand, Flutter is developed by Google using Dart, which is not as widely adopted as JavaScript, but uses a single code base for both iOS and Android. However, this single code base creates a larger app size compared to React Native app.

In the context of the receipt extractor application, the emphasis will be on the execution speed and stability on the mobile app. Therefore, Flutter, as shown in *Table 4*, is fast with different features for development, such has the hot reload with the portability from Android to iOS without code modification.

|  |  |  |
| --- | --- | --- |
| **Programming Language** | **pros** | **Cons** |
| **React Native** | * Strong Community Support with big tech contribution. * Bring modern web techniques to mobile support. * Use JavaScript syntax extension for designing UI. * Access native hardware feature like camera and storage. * Encourage modularity and reusable of component. | * Have performance limitation compared to native app in complexes scenario using JavaScript thread or memory optimization. * Highly dependable on third-party libraries for navigation and file system operations which may affect consistency and the reliability. |
| **Flutter** | * Hight-Performance using his own rendering engine for view component offering a close performance to native application. * Using Dart Programming language which is efficient for memory management and garbage collection offering fast performance. * Providing customizable widget for development. * Hot-Reload feature for development. * Assuring a consistency across the different platform. | * Produce larger application size due to the different widget renderer that may affect the app size. * Relatively new community compared to react native which may affect the resources available. |

Table 4 - Advantage and limitation of Flutter and React Native from "React Native vs Flutter, cross-platform mobile application frameworks" written by Wu.

## Conclusion

# Requirements specification

[ tell the feature, why these features related with literature review].

* Detect Receipt Part
  + Inspired by paper 2.
  + Help for classification.
* Extract receipt data.
  + Key feature in all the papers
* Classify Receipt
  + Important for analytics
  + Use of an LLM
  + Refer to paper 1 and 2.
* User interface
  + Give access system to users.
  + No viable solution out of the paper.
* Database management
  + No referring in current paper.
  + Storing of historic for user -> may provide analytics.

## Receipt Section Detection

As referenced while analyzing *Paper 2 - Information Extraction from Scanned Invoices using Machine Learning, OCR, and Spatial Feature Mapping Techniques ,* it presented technologies of detection of the different sections of a receipt before applying the text extraction (Darsha, 2023). These strategies will be kept by detecting the key element of the receipt to increase the formatting and classification of the receipt data by developing an image recognition model to identify the precise areas representing some elements of the receipt, such has the total, items list, time, and shop information.

## Text Extraction

A key feature of the receipt extraction will be the character recognition which will allow the digitalization of the data from the paper to the receipt management system. As per the inside from the work from Odd and Theologou (Odd and Theologou, 2018)(*Paper 1*), Benchekrou (Benchekroun, 2022)(*Paper 3*) and Darsha (Darsha, 2023)(*Paper 2*) showing Optical Character Recognition (OCR) technologies, such has Tesseract OCR and Google Cloud Vision API, along with advanced machine learning models. Presented in all three research papers, it shows the importance of the text extraction as a vital feature as it impacts the precision of the global system.

## Format and Classify text extraction.

After extracting the raw data from the receipt, a formatting and classification showed in *Paper 1 - Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms* and *Paper 2 - Information Extraction from Scanned Invoices using Machine Learning, OCR, and Spatial Feature Mapping Techniques* underscore the effectiveness of the machine learning algorithm for text classification using models such has LinearSVC and MLPClassifier. This approach facilitates the classification of the receipt and the formatting of the output to enable the uniformities of the data structure for a possible database storage and analytics.

## User Interface

Even though the papers criticize the receipt extraction, it showcases multiple technologies for the processing, but none of them provide user-interface for common users to employ these technologies, which without this will restrict the usage of the software to specialized set of users. Therefore, a User Interface (UI) is required to increase the spectrum of possible users. As showed in *Paper 4 - React Native vs Flutter, cross-platform mobile application frameworks ,* comparing Flutter and React Native mobile developing framework (Wu, 2018). These frameworks demonstrated the portability of the UI development through multiple devise common use of the system.

# Design

[ design of these feature (globally) with the technologies I think I’ll use (note at this point I haven’t start working in the artefact, so purely theorical]

After discussing the existing work in *Literature review* and discussing the different requirements in *Requirements specification*, we can now discuss the design of the system application. At that stage of development, it can be separated into two distinct parts, which will be working together simultaneously. The server side, which will manage all the computationally exhausting processing, such as the “receipt extraction feature” which will reduce the execution time and provide a better user experience while communicating with the database. Also, everything will be built in a docker container to offer an easy deployment on any device. And on the other hand, the mobile application will be used as a gateway for the user to operate the system and communicate with the server. In the next following sections, a list of the features will be integrated.

## Server

* + Receipt section detection model.
  + Receipt extraction feature (section detection + OCR + Classification + user).
  + Prediction user correction feature.
  + Server Communication.
  + Database.
  + Database management.

### Receipt extraction feature.

As previously mentioned, the receipt extraction is the key feature of the system which has global purpose to extract from a receipt image into a specific format while predicting the type of receipt. The fields which will be extracted are the shop information, the list of items, the total, and the date of the transaction, which are to be formatted into a json format to be stored in a database. The structure of this feature is composed of three major parts; the receipt section image recognition model, followed by the text extraction of this section, and finally, the formatting and classification of the receipt extracted data.

The receipt section detection model is the first step of the receipt extraction feature. The model will compose of a YOLO v8 model to predict the different receipt sections, which are explained in the preceding paragraph trained by Mauritian receipt to better predict the section of the receipt. To describe in more detail, the prediction will take input as a picture of 640 per 640 pixel and return for each class (chop information, item list, …) their coordinate on the picture.

The second step of this feature is the extraction of the text display of these class extracts through OCR, for each detected class, a sub-image composed of coordinates of the class prediction, and the extracted text. As for the choice of the OCR, based on the different OCR technologies showed in the *Literature review*, the OCR used would be Tesseract OCR due to its high efficiency and capacity to be deployed locally without using a third-party service.

Finally, the classification and formatting of the of the output of the previous function output. This task will be performed by a Custom Multi-Layer Perceptron (MLP) to, firstly classify the receipt into categories (groceries, restaurant, cosmetics, electronics, etc.), then format all the information into a specific format (see *Figure 1 - Receipt Extraction Feature output format*) in json for an easy manipulation afterward.



Figure 1 - Receipt Extraction Feature output format.

### User Correction Feature.

While the data will be automatically extracted, the user will have the opportunity to “review” the extracted data, through the flutter application interface, to ensure the data is correct, which will be stored in the database. The corrected data will be set as “reviewed” and will be accessible by the analytics feature to ensure the accuracy of the data. The structure of the feature can be seen in the sequence diagram.

[ sequence diagram to be input]

### Server Communication

As for the communication between the server and the application, the HTTP (Hypertext Transfer Protocol) will be integrally handled by a FastAPI application host into the server. The primary function of the FastAPI app is to execute the requested server feature and relay back the result to the mobile app. Each endpoint will be related to a feature of the server and therefore, forming the server’s operation core.

### Database

The server architecture will include a database system which will maintain user and receipt’s related data. Using Docker technology, the server will use a PostgreSQL database initializing alongside the server’s image. During the server’s image building, the PostgreSQL image will be initialized, and if in absence of pre-existing data, the database schema will be initialized through the execution of an SQL scripts. The usage of docker will ensure a seamless integration of the system.

The database schema is composed of three primary tables, outline as follows:

* Users table: A table for user related information.
* Raw Receipt Table: A table to store the unprocessed data before any data manipulation.
* Receipts Tables: A table which will contain all data related to the receipt extraction.

Each table use a specific purpose and all tables’ keys can be found in the database schema showed in [insert reference to database schema]

## Application

* + Upload receipt picture for receipt extraction.
  + List receipt historical.
  + Analytics generation
  + Application communication with server.

### Receipt upload for receipt extraction.

On the user side, the key feature of the application is the upload of the receipt of the extraction of its data to be stored in the database. The first step is the loading of the receipt, either through the uploading from the phone gallery, or a live capture through the camera. It is then followed by the upload to the server through http post request, where the receipt extraction will be done. The result will be stored as “pending” status in the database before being sent back to the mobile application. With the result of the extraction, the user will have the option to review and modify the result to correct potential error before being sent back to the server to update the receipt data stored in the database and set is status as “reviewed”.

### List Receipt Historical

The application will allow the user to review their historical entries in a form of a list where key information about the receipt will be display. It will query the “reviewed” status receipt from the user before formatting the data for display.

### Analytics

The application will offer the user the possibilities to show analytics through their data stored in the database. When this page has loaded, the current user’s data will be requested to the server, which will query it from the database back to the mobile application. The data will be processed directly from the application during the building of the widget to create different analytics for the user. The first graph provided will be a line graph showing the sum of purchase/s per day within a range of one month, with the capacity to display for older month/s. It will allow the user to keep track of the money spent during the month. A second graph provided will be generated as a pie graph that shows the sum of money spent per receipt categories (groceries, restaurant, etc.) per month, also with the ability to see data of older month/s. It will allow the user to view the amount for each category and allow it to manage a potential budget.