A computer screen shot of a computer

Description automatically generated

Smart Receipt Management and Extraction

Interim Report

M00843707

Damien Maujean

12 JANUARY 2024

Supervise by Nivede Issur

Table of Contents

[Table of figure 3](#_Toc156037973)

[Introduction 4](#_Toc156037974)

[Background 4](#_Toc156037975)

[Deliverable 4](#_Toc156037976)

[Server-side 4](#_Toc156037977)

[Mobile Application 4](#_Toc156037978)

[Literature Review 4](#_Toc156037979)

[Receipt Extractor 4](#_Toc156037980)

[Paper 1 - Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms. 4](#_Toc156037981)

[Paper 2 - Information Extraction from Scanned Invoices using Machine Learning, OCR and Spatial Feature Mapping Techniques 6](#_Toc156037982)

[Paper 3 - Computer Vision for Document Image Analysis and Text Extraction 6](#_Toc156037983)

[Web App 7](#_Toc156037984)

[Paper - React Native vs Flutter, cross-platform mobile application frameworks. 7](#_Toc156037985)

[Summary 8](#_Toc156037986)

[1rst step. 9](#_Toc156037987)

[Model Training code 9](#_Toc156037988)

[Gantt Chart 10](#_Toc156037989)

[Requirements Specification 11](#_Toc156037990)

[Functional requirements 11](#_Toc156037991)

[Database Requirement 11](#_Toc156037992)

[Analysis and design 12](#_Toc156037993)

[System structure 12](#_Toc156037994)

[Receipt Extraction Sequence Diagram 13](#_Toc156037995)

[Database structure 13](#_Toc156037996)

[Implementation and Testing 14](#_Toc156037997)

[Implementation 14](#_Toc156037998)

[Testing 14](#_Toc156037999)

[Ethics 14](#_Toc156038000)

[Reference 15](#_Toc156038001)

[Bibliography 15](#_Toc156038002)

# 

### Table of figure

[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 5](#_Toc156038851)

[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. 6](#_Toc156038852)

[Table 3 - Advantage and Limitation of technologies used in "Computer Vision for Document Image Analysis and Text Extraction" written by Benchekrou. 6](#_Toc156038853)

[Table 4 - Advantage and limitation of Flutter and React Native from "React Native vs Flutter, cross-platform mobile application frameworks" written by Wu. 7](#_Toc156038854)

[Figure 1 - Yolo Training Jupiter Notebook 9](#_Toc156038860)

[Figure 2 - Project Gantt Chart 10](#_Toc156038861)

[Figure 3 - Global Project Design 12](#_Toc156038862)

[Figure 4 - Receipt extraction feature Sequence Diagram 13](#_Toc156038863)

[Figure 5 - Database Table structure 13](#_Toc156038864)

[Figure 6 - Ethic form 14](#_Toc156038865)

# Introduction

## Background

Managing the accounting of a company is a long complex work but therefore “plays a significant role in the effective management process (Alabdullah, 2019), and one of the tasks is to log all the differences expenses for varied reasons. Therefore, the only way is to manually extract and store all this information in their receipt paper, which represents repetitive work and a big loss of time and energy which could be placed into more important tasks. Moreover, it represents also an important amount of paper which need to be store and represent a big quantity of paper that could be digitalize. A few solutions exist but nothing corresponding to the Mauritian market either professionally or personally but are limited and expensive. To solve this problem and accessible user-friendly receipt extractor powered by state-of-the-art technologies.

## Deliverable

### Server-side

The first deliverable is the server-side program containing the Receipt extraction model which will extract the different part of the receipt to be extracted by an Optical Character Recognition (OCR). Moreover, the server-side will manage the storage of the extracted data and manage the communication with the client-side application.

### Mobile Application

The second deliverable is a mobile application which will represent the gateway of the user to the different feature of the Receipt Extraction. It will serve to communicate with the server and send the picture of the receipt to the model extractor, retrieve the old receipt data, generate the different analytics and allow the user to correct the extracted data before storing into the database.

# Literature Review

## Receipt Extractor

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

This study made by Joel Odd and Emil Theologou research and develop on “investigated if it was feasible to use machine learning tools on OCR extracted text data to classify receipts and extract specific data points” (Odd and Theologou, 2018).

# 

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

Has showed in the table 1, it is showed that two elements compose the receipt extractor, the Machine learning model and the Optical Character Recognition (OCR) and that for these elements multiple technologies were tested. For the OCR, we have the Azure Computer Vision APIU and Google Drive Rest API which are efficient but are third party dependent and may require data processing, on the other hand tesseract OCR offer local deployment advantage and result more quickly but not has efficient. For the machine learning model two technologies were tested, the Custom N-Gram Model which is dedicated to the purpose with efficient classifying and share the same weakness has the Scikit-learn which is the generalization problem due to a small dataset.

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

For the research paper made by W.B. Darsha which solve the fooling problem “Extracting information from scanned invoices (images) is a challenging task”(Darsha, 2023) using an model and a OCR.

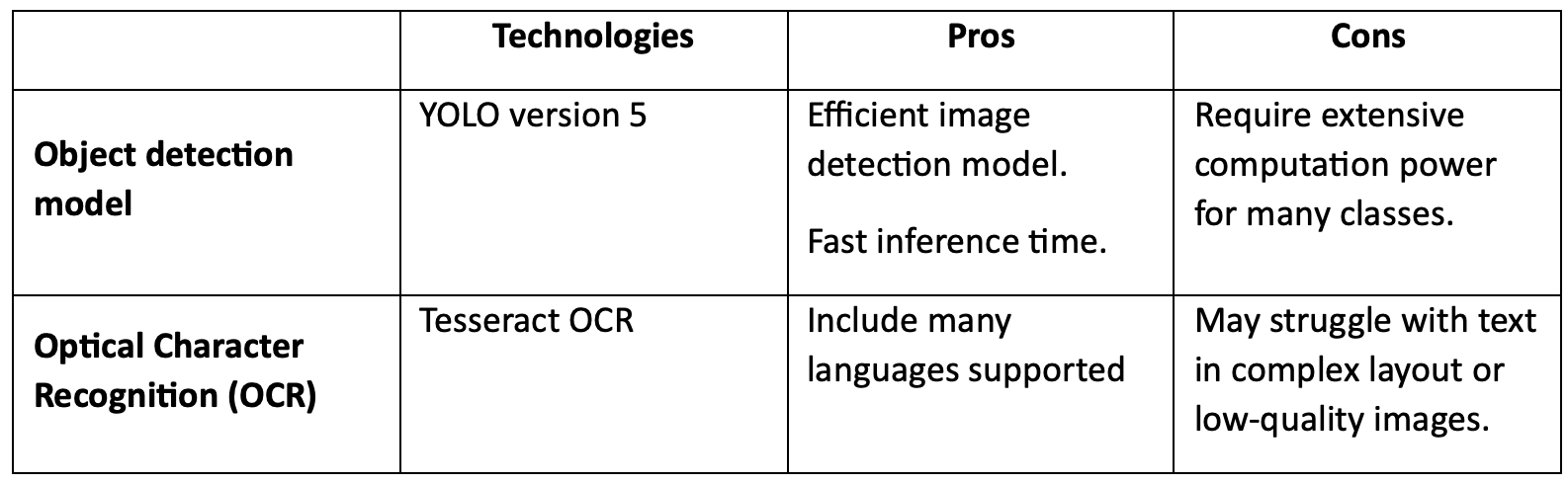


Table 2 - Advantage and Limitation of technologies used in "Information Extraction from Scanned Invoices using Machine Learning, OCR and Spatial Feature Mapping Techniques" written by Darsha.

Has showed in Table 2, this paper used the YOLO object detection (You Only look Once) to identify the different aspect of the receipt follow by the usage of the Tesseract OCR mentioned in the earlier section of the Literature Review. The YOLO model can be easily call and train through the import of Ultralitics python package.

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

This study has for purpose to “investigates in depth a major component used in Document Image Processing known as Optical Character Recognition (OCR).”(Benchekroun, 2022), it hasn’t the same goal but share the same step of analysis and text extraction with a study using synthesis data..

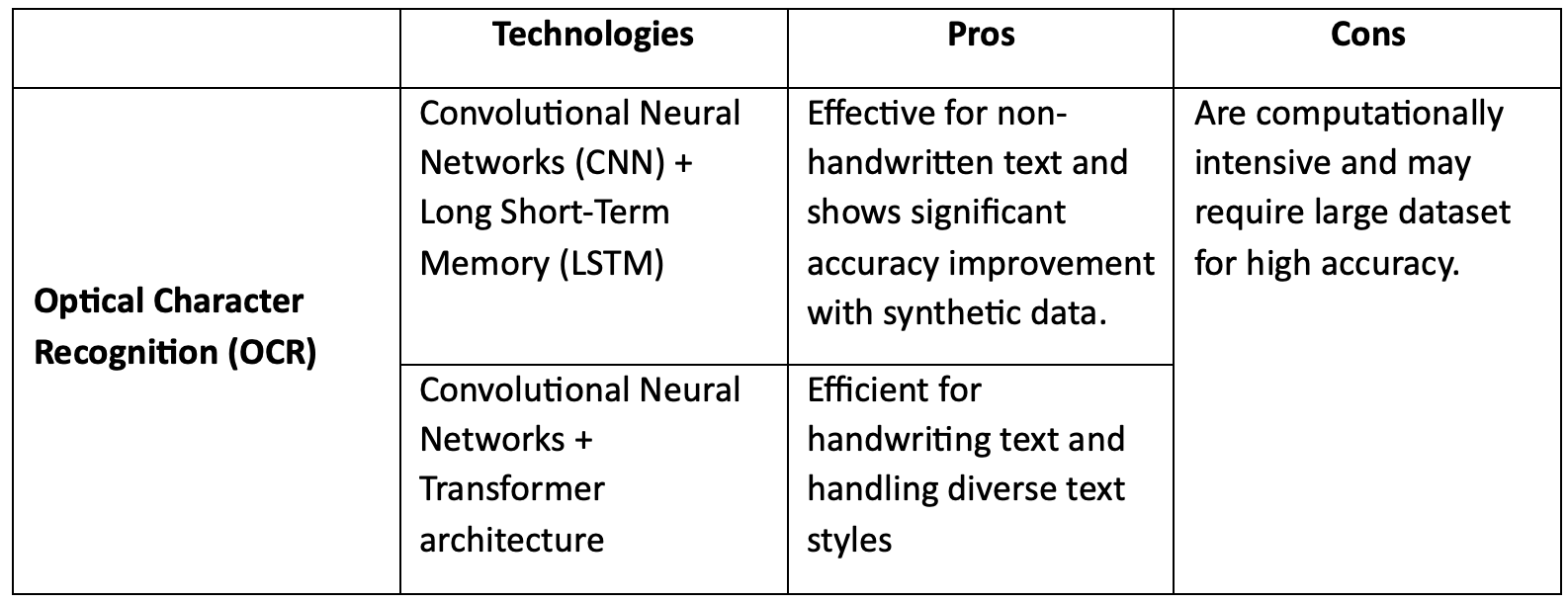


Table 3 - Advantage and Limitation of technologies used in "Computer Vision for Document Image Analysis and Text Extraction" written by Benchekrou.

The study experiments on the usage have showed in table 3 of two OCR within the use synthetic data which improve of 24% bringing the accuracy on non-handwritten to 97%, giving also a solution for handwriting with CNN+Transformer architecture which in our project is not relevant since the main data will be printed. Furthermore, the usage of synthetic data since promising to increase the training dataset and allow to the model a better globalization of the extraction.

## Web App

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

This research paper done by Wenhao Wu confront two mobile programing language, react native and flutter to “execute a comprehensive study on React Native and Flutter”(Wu, 2018)

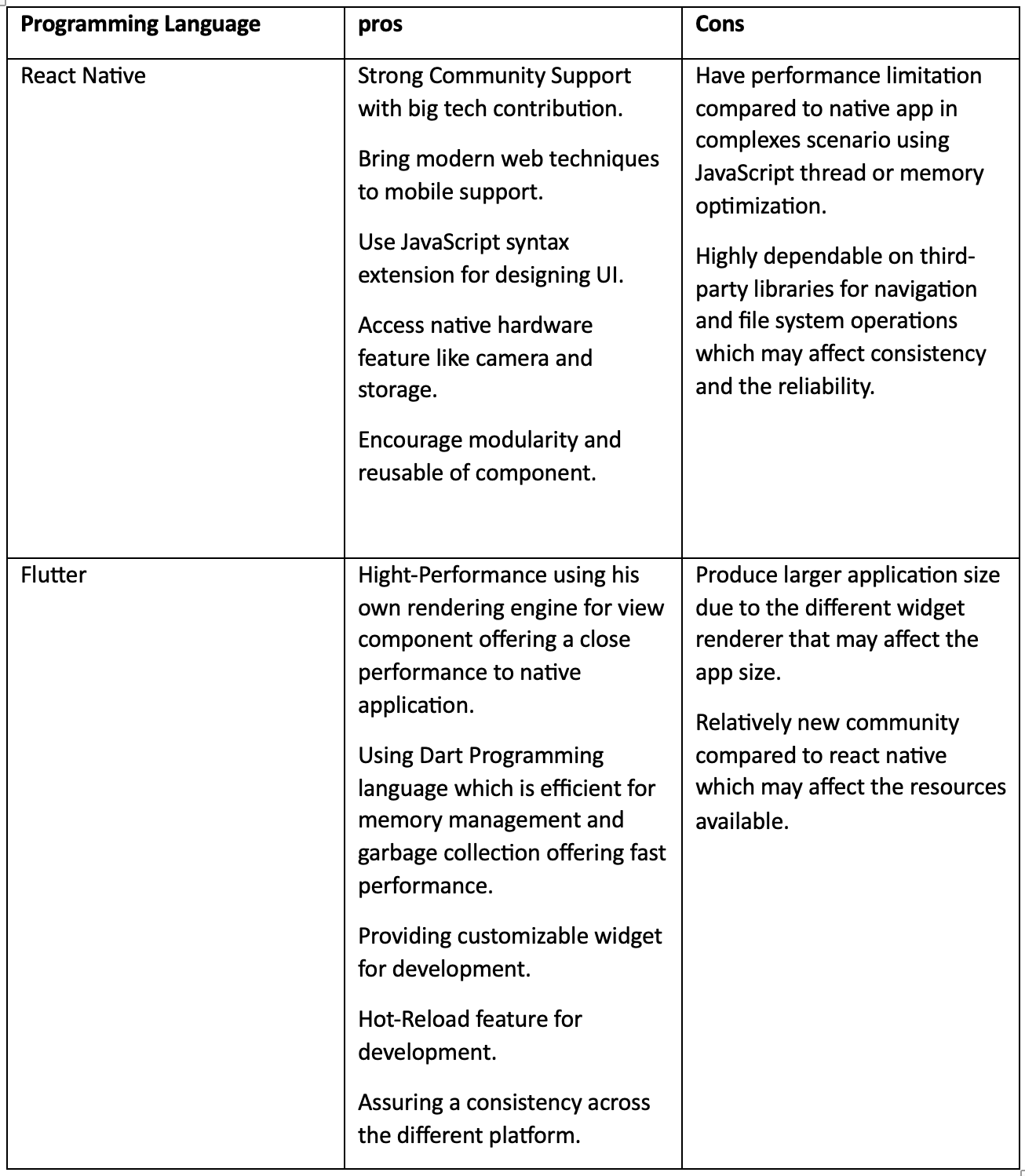


Table 4 - Advantage and limitation of Flutter and React Native from "React Native vs Flutter, cross-platform mobile application frameworks" written by Wu.

From the table 4 we can see the advantage and limitation of React Native and Flutter programing language, both cross-platform and with their own strength in different domain. React Native having a big community and using modern web technique while Flutter having some high rendering performance with consistent UI across platform offer a good alternative for mobile application development.

## Summary

Has we saw in the previous sections of the Literature Review, the project can be separate in two parts, the receipt extraction model and the mobile application having both multiple technologies usable. For the receipt extraction model which can also be separate into two technology such has the Object detection Model followed with Optical Character Recognition technologies. From the different research, the YOLO object detection model version 8 seems the most suitable option with his fast efference time and accessibilities with python language combine with Tesseract OCR also fully compatible to python and his efficiency to extract printed text suite perfectly my requirement for receipt extraction being both state-of-the-art technologies in their domain. As for the mobile application development, Flutter programming language is more appropriate compared to React Native with is high performance in rendering and his consistent UI across platform.

# 1rst step.

## Model Training code

|  |
| --- |
|  |
| Figure 1 - Yolo Training Jupiter Notebook |

### Gantt Chart



Figure 2 - Project Gantt Chart

The Gantt Chart showed in *figure 1* is the same one provided during the proposal and is still up to date to the current project development. Therefore, no modification is to be done and the rest of the development will be done following this chart.

## Requirements Specification

### Functional requirements

* **Mobile Application**: Development of a user-friendly mobile application using flutter acting has a gateway to the different feature for the user.
* **Receipt recognition model**: Implementation of a Yolo algorithm to identify the different part from the receipts.
* **Data Extraction**: Utilization of Tesseract has OCR to extract the data from the different parts from receipts.
* **User correction**: Allowing the user through the application to correct the data before stored into database.
* **Server-Application Communication**: Implement API communication between the mobile application and the server.
* **Receipt Management**: feature to centralize all the different user receipt data using Postgres.
* **Purchase Analytics**: Provide analytics about user purchase habit.
* **Data Export**: Capability to export the receipt data in CSV or Excel format.
* **Send Receipt to Extraction**: Allow the user to send to the server-side one to many receipt int different picture format or pdf format.

### Database Requirement

* **Store User information:** Storing the user information such has user id, email and password.
* **Store Raw email:** Storing raw receipt image data with a receipt id and an upload datetime.
* **Storing Processed data:** Storing all the raw extracted and parse data from the servers processing with the execution datetime and the receipt id.

## Analysis and design

### System structure



Figure 3 - Global Project Design

Base partially on the research showed in the Literature review section, we can see has showed in *figure 2* the project structure is composed of a mobile app coded in flutter who will manage some key feature listed on the figure. It’ll communicate through API call to the server who will itself communicate with the Postgres database accessible through the local environment, the usage of the local environment between the server and the database allows a secure and fast connection between them. The server will serve to manage the database query, generate the analytics and contain the receipt extraction model, the model which will be trained with google Collaboratory which provide powerful GPU which will reduce the training time.

### Receipt Extraction Sequence Diagram

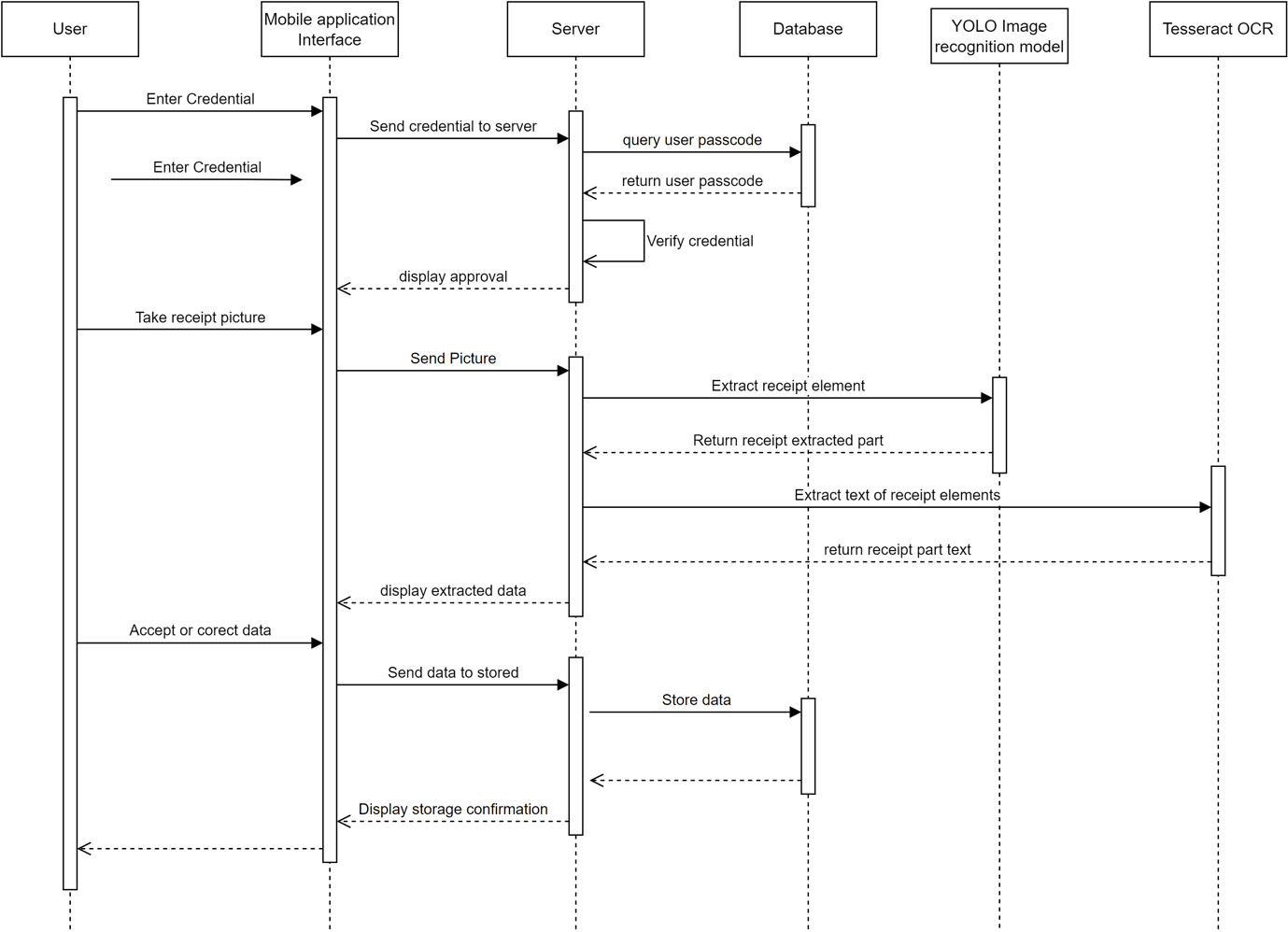


Figure 4 - Receipt extraction feature Sequence Diagram

The *figure 3* show the different interaction of the different agents such has the user with the mobile application, the mobile application with the server and finally the server with the database, YOLO model and finally the OCR.

### Database structure

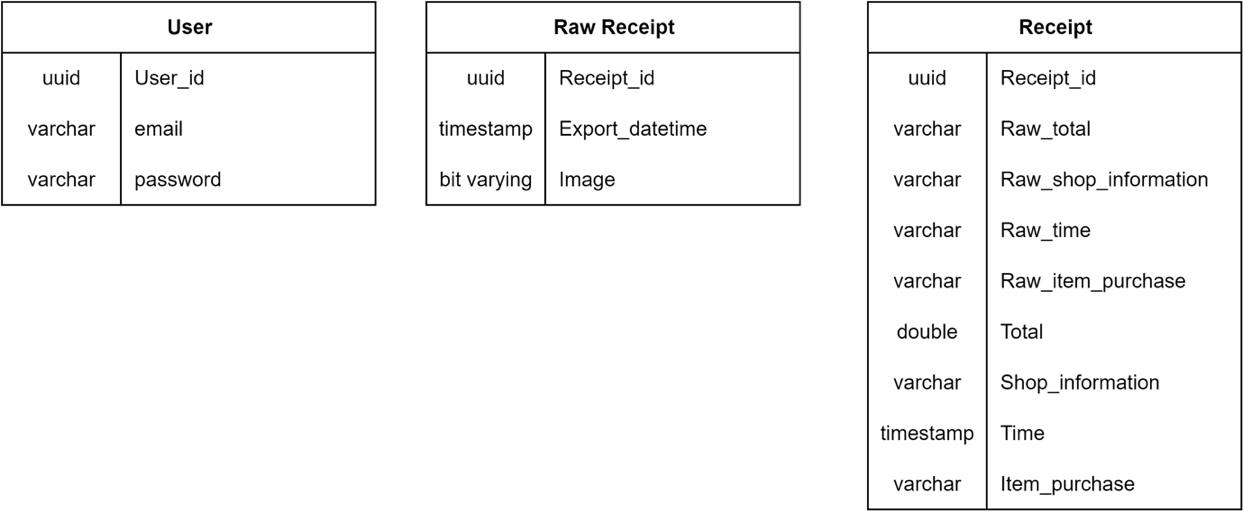


Figure 5 - Database Table structure

## Implementation and Testing

### Implementation

For this project, has explained in the precedents section of this report and more precisely in System Structure section, a mobile application using Flutter will be implemented. On the other side, in the server which will be mostly code in python will contain the receipt recognition component compose of a YOLO algorithm version 8 callable through Ultralitics followed by the efficient text extraction Tesseract. The project architecture is design with a clear separation of concern, where the back-end process receipt data and the front end which handling user interaction.

### Testing

Testing is a critical part in project development. Therefore, a Test-Driven Development (TDD) will be applied to ensure a robust and error-free implementation and development. The unit tests will be written for each key module of the project, followed by integration tests to ensure a global stability. Furthermore, some performance measurement will be done firstly on the receipt recognition to ensure good result and high-speed processing, and secondly measuring the application to server respond time to ensure good user experience.

## Ethics

|  |  |
| --- | --- |
| Figure 6 - Ethic form |  |

# Reference

Alabdullah, T. (2019) ‘Management Accounting and Service Companies’ Performance: Research in Emerging Economies’, *Australasian Accounting, Business and Finance Journal*, 13(4), pp. 100–118. Available at: https://doi.org/10.14453/aabfj.v13i4.8.

Benchekroun, O. (2022) ‘Computer Vision for Document Image Analysis and Text Extraction’.

Darsha, W.B. (2023) ‘Information Extraction From Scanned Invoices using Machine Learning, OCR and Spatial Feature Mapping Techniques’.

Odd, J. and Theologou, E. (2018) ‘Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms’.

Wu, W. (2018) ‘React Native vs Flutter, cross-platform mobile application frameworks’.

# Bibliography

*Azure AI Vision with OCR and AI | Microsoft Azure* (no date). Available at: <https://azure.microsoft.com/en-us/products/ai-services/ai-vision> (Accessed: 12 January 2024).

*Flutter documentation* (no date). Available at: <https://docs.flutter.dev/> (Accessed: 12 January 2024).

*Google Colaboratory* (no date). Available at: <https://colab.research.google.com/> (Accessed: 12 January 2024).

*Présentation de l’API Google Drive | Google Drive* (no date) *Google for Developers*. Available at: <https://developers.google.com/drive/api/guides/about-sdk?hl=fr> (Accessed: 12 January 2024).

*React Native · Learn once, write anywhere* (no date). Available at: <https://reactnative.dev/> (Accessed: 12 January 2024).

*scikit-learn: machine learning in Python — scikit-learn 1.3.2 documentation* (no date). Available at: <https://scikit-learn.org/stable/index.html> (Accessed: 12 January 2024).

*Tesseract User Manual* (no date) *tessdoc*. Available at: <https://tesseract-ocr.github.io/tessdoc/Home.html> (Accessed: 12 January 2024).

Ultralytics (no date) *Train*. Available at: <https://docs.ultralytics.com/modes/train> (Accessed: 12 January 2024).