AN ENGINEERING PROJECT REPORT

On

Optical Character Recognition

Submitted By:

Abhishek Joshi	**200302**
Prashant Phuyal	**200324**
Prashant Shrestha	**200325**
Saurav karki	**200334**

Submitted To:

The Department of Information and Communication Technology

In partial fulfillment of requirement for the degree of

Bachelor of Engineering in Information Technology

Cosmos College of Management and Technology

(Affiliated to Pokhara University)

Tutepani, Lalitpur, Nepal

Date of Submission: 5th July 2024

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Prof Dr. Ram Kumar Sharma (Principal), Dr. Babu Ram Dawadi (Vice Principal), Er. Ajay Adhikari (HOD) for their kind help and suggestion to complete this project. We would like to thank all our faculty teachers for always encouraging and motivating us to the highest peak and to provide us the opportunity to prepare the project.

We express our deep appreciation to Mr. Ranjan Raj Aryal **(**Supervisor) who always encouraged us and helped us at any time. We would also like to share our gratitude to Pokhara University for providing an opportunity to evaluate our programming skills through this project.

Lastly, we heartily thank all our classmates and seniors who encouraged us throughout the project and the course.

ABSTRACT

In the running world, there is growing demand for the software systems to recognize characters in computer system when information is scanned through paper documents as we know that we have number of newspapers and books which are in printed format related to different subjects. These days there is a huge demand in "storing the information available in these paper documents in to a computer storage disk and then later reusing this information by searching process". One simple way to store information in these paper documents in to computer system is to first scan the documents and then store them as IMAGES. But to reuse this information it is very difficult to read the individual contents and searching the contents form these documents line-by-line and word-by-word. The reason for this difficulty is the font characteristics of the characters in paper documents are different to font of the characters in computer system. As a result, computer is unable to recognize the characters while reading them. This concept of storing the contents of paper documents in computer storage place and then reading and searching the content is called document processing. Sometimes in this document processing we need to process the information that is related to languages other than the English in the world. For this document processing we need a software system called Optical Character Recognition System (OCR). This process is also called Document Image Analysis (DIA).

Table of Content

Table of Contents

LIST OF FIGURES

ABBREVIATION

OCR: Optical Character Recognition

API: Application Programming Interface

CHAPTER 1

1 Introduction

1.1 Background of the Project

Optical character recognition (OCR) is a system that converts input text into machine-encoded format. Today, OCR is helping not only in digitizing the handwritten medieval manuscripts, but also helps in converting the typewritten documents into digital form. This has made the retrieval of the required information easier as one doesn't have to go through the piles of documents and files to search the required information. Organizations are satisfying the needs of digital preservation of historic data, law documents, educational persistence etc.

An OCR system depends mainly, on the extraction of features and discrimination/classification of these features (based on patterns). Handwritten OCR have received increasing attention as a subfield of OCR. It is further categorized into offline system, and online system based on input data. The offline system is a static system in which input data is in the form of scanned images while in online systems nature of input is more dynamic and is based on the movement of pen tip having certain velocity, projection angle, position and locus point. Therefore, an online system is considered more complex and advance, as it resolves the overlapping problem of input data that is present in the offline system.

The purpose of this OCR system is to take handwritten English characters as input, process the character, train the neural network algorithm, to recognize the pattern and modify the character to a beautified version of the input. This work is restricted to English characters and numerals only. It is also helpful in recognizing special characters. It can be further developed to recognize the characters of different languages. One of the primaries means by which computers are endowed with humanlike abilities is through the use of a neural network. Neural networks are particularly useful for solving problems that cannot be expressed as a series of steps, such as recognizing patterns, classifying them into groups, series prediction and data mining.

1.2 Problem Statement

Problems in the manual attendance system that lead this project to be proposed are stated below:

- Currently manual data entry is slow and prone to human error and it leads to incorrect information, poor decision-making, and increased operational costs.
- Large amounts of paper documents are hard to manage and store which causes inefficiency, difficulty in locating documents, and high storage costs.
- Historical texts are often only in physical formats and hard to access, limiting research, public knowledge, and cultural preservation efforts.

1.3 Objective

The primary objective of this project is

- To develop a robust and efficient Handwritten OCR system capable of accurately recognizing and converting handwritten text into machine-encoded text.

1.4 Applications

This project can be implemented in following application areas:

- Converting handwritten notes into digital text for easy organization and searchability.
- Automating the recognition and processing of handwritten survey data collected in the field for research purposes.

CHAPTER 2

2 Literature Review

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene photo or from subtitle text superimposed on an image.

There have been many research papers published that use deep learning techniques for handwritten text recognition. A few of them are mentioned below. This paper is about OCR.A pipeline development needed to extract text from identity cards or to read resolution which are received from the users can be identified properly. The image whose threshold is below 50 will not be used for letter detection. For deciding the threshold, a lot of trials has been done. Two methods are used for text detection. Blur contour is based on two things i.e., faster RCNN and Detection. Contour based detection is particularly used for detecting the text and it is very helpful. If the proper thresholding and pre-processing of the image will be done then the image can also be detected. In the text detection phase RCNN is generally used. This approach has better time for detecting the text. An end-to-end pipeline is developed by combining several models for optical character recognition. This model uses a neural based approach and gives more accuracy than the model which uses image processing techniques. The disadvantage is that it is unable to detect fraud cases. the number plate of the vehicle. Some important pipelines are text recognition and detection. The preprocessing is needed for making the images more useful and decreasing the noise.

In 1974, Ray Kurzweil started the company Kurzweil Computer Products, Inc. and continued development of omni-font OCR, which could recognize text printed in virtually any font. Kurzweil is often credited with inventing omni-font OCR, but it was in use by companies, including CompuCom, in the late 1960s and 1970s. Kurzweil used the technology to create a reading machine for blind people to have a computer read text to them out loud. The device included a CCD-type flatbed scanner and a text-to-speech synthesizer. On January 13, 1976, the finished product was unveiled during a widely reported news conference headed by Kurzweil and the leaders of the National Federation of the Blind.

In the 2000s, OCR was made available online as a service (Web OCR), in a cloud computing environment, and in mobile applications like real-time translation of foreign-language signs on a smartphone. With the advent of smartphones and smart glasses, OCR can be used in internet connected mobile device applications that extract text captured using the device's camera. These devices that do not have built-in OCR functionality will typically use an OCR API to extract the text from the image file captured by the device. The OCR API returns the extracted text, along with information about the location of the detected text in the original image back to the device app for further processing or display.

CHAPTER 3

3 Requirement Analysis

Requirement analysis is a crucial phase in the software development lifecycle that involves gathering, analyzing, and documenting the needs and expectations of stakeholders for a new or modified product or system.

3.1 Feasibility Study

A feasibility study is a comprehensive analysis conducted to determine the viability, practicality, and profitability of a proposed project or system. It helps decision-makers understand whether to proceed with the project by examining various critical factors. It aims to determine whether the project is worth pursuing by evaluating various factors, including technical, economic, legal, operational, and scheduling aspects.

3.1.1 Technical Feasibility

Technical feasibility is a critical component of a feasibility study that assesses whether a project can be executed with the current technology, resources, and expertise available. It focuses on the technical aspects and requirements needed to complete the project successfully.

- Evaluate Technical Requirements: Determine the hardware, software, and technological infrastructure needed for the project.
- Assess Technical Skills: Ensure that the project team possesses the necessary technical skills and expertise.
- Identify Technical Challenges: Recognize potential technical problems or challenges that might arise during the project.
- Analyze Technical Risks: Evaluate the risks associated with the technical aspects of the project and plan for mitigation.

3.1.2 Operational Feasibility

Operational feasibility is an essential aspect of the feasibility study that assesses whether a proposed project can be successfully integrated into the organization's existing operations and processes. It evaluates the project's impact on current operations, identifies necessary changes, and ensures that the organization has the capability to support the project once it is implemented.

- **Assess Impact on Current Operations**: Determine how the project will affect existing workflows, processes, and systems.
- **Evaluate Resource Availability**: Ensure that the organization has the necessary resources (human, financial, and physical) to support the project.
- **Identify Operational Challenges**: Recognize potential operational issues or challenges that might arise during or after project implementation.
- **Ensure Smooth Integration**: Plan for the seamless integration of the new project into the existing operational framework.

3.1.3 Economic Feasibility

Economic feasibility, also known as financial feasibility, assesses whether a proposed project is financially viable and worthwhile. It involves analyzing the costs, benefits, financial risks, and returns associated with the project to determine if it makes economic sense to proceed. This aspect

of the feasibility study is crucial for decision-makers as it provides a clear picture of the project's financial implications.

- **Assess Costs**: Identify and estimate all costs associated with the project, including initial, operational, and maintenance costs.
- **Evaluate Benefits**: Determine the financial and non-financial benefits that the project will deliver.
- **Analyze Financial Risks**: Identify and evaluate potential financial risks and uncertainties.
- **Calculate Return on Investment (ROI)**: Estimate the ROI and other financial metrics to evaluate the project's profitability.
- **Support Decision-Making**: Provide a solid financial basis for deciding whether to proceed with, modify, or abandon the project.

3.1.4 Market Feasibility

Market feasibility is a critical aspect of a feasibility study that evaluates the potential market for a proposed project, product, or service. It aims to determine whether there is a demand for the offering, the size and characteristics of the target market, competitive landscape, and potential market share.

- **Assess Market Demand**: Determine if there is sufficient demand for the proposed product or service.
- **Identify Target Market**: Define the target audience, their needs, preferences, and buying behavior.
- **Analyze Market Size**: Estimate the size of the market and potential market share.
- **Evaluate Competition**: Analyze the competitive landscape, including key competitors and their strengths and weaknesses.
- **Identify Market Entry Barriers**: Recognize potential barriers to market entry and develop strategies to overcome them.

3.1.5 Legal and Compliance Feasibility

Legal and compliance feasibility assesses whether a proposed project, product, or service can be implemented in compliance with legal requirements, regulations, and standards. It focuses on ensuring that the project adheres to applicable laws and regulations throughout its lifecycle, from development to deployment and ongoing operation.

- **Identify Legal Requirements**: Determine the legal regulations and requirements that apply to the project, including local, national, and international laws.
- **Assess Compliance**: Evaluate whether the project can meet legal standards, industry regulations, and ethical guidelines.
- **Mitigate Legal Risks**: Identify potential legal risks and develop strategies to mitigate them.
- **Ensure Ethical Standards**: Ensure that the project aligns with ethical principles and corporate social responsibility (CSR) guidelines.
- **Secure Necessary Permissions**: Obtain required licenses, permits, certifications, and approvals for legal operation.

3.1.6 Risk Analysis

Risk analysis is a systematic process of identifying, assessing, and prioritizing risks associated with a project, decision, or action. It involves analyzing potential risks, estimating their likelihood and impact, and developing strategies to mitigate or manage them effectively. Risk analysis is crucial for decision-making as it helps organizations understand and prepare for uncertainties that could impact project objectives, timelines, and outcomes.

- **Identify Risks**: Recognize potential threats and opportunities that could affect the project.
- **Assess Likelihood and Impact**: Evaluate the probability of risks occurring and their potential consequences on project goals.
- **Prioritize Risks**: Rank risks based on their likelihood and impact to focus resources on the most significant threats.
- **Develop Risk Responses**: Define strategies to mitigate, avoid, transfer, or accept risks.
- **Monitor and Control Risks**: Continuously monitor identified risks and implement controls to manage them throughout the project lifecycle.

3.2 Functional Requirement

The functional requirements focus on essential capabilities that allow the system to perform basic text recognition tasks effectively. These requirements are typically geared towards simplicity, ease of use, and basic functionality.

- **Text Recognition**: Accurate identification and extraction of alphanumeric characters from images.
- **Language Support**: Basic support for common languages and character sets (e.g., English ASCII).
- **Output Formats**: Ability to generate plain text or basic PDF documents from recognized text.
- **User Interface**: Simple interface for uploading images and retrieving recognized text.
- **Accuracy**: Basic accuracy metrics and error handling for reliable text extraction.
- **Performance**: Efficient processing with minimal latency and resource requirements.
- **Integration**: Basic APIs or interfaces for straightforward integration with other applications.
- **Security**: Basic measures to ensure data security and privacy during processing.

3.3 System Requirement

- A computer with atleast 8 GB of RAM.
- IDE**:** Microsoft Visual Studio
- Uninterrupted internet access for importing various external libraries neeeded during development.
- OS: Windows 10
- Flutter: front-end
- Python: back-end

CHAPTER 4

4 METHODOLOGIES

Developing an OCR (Optical Character Recognition) project involves a systematic methodology to ensure comprehensive planning, efficient implementation, and successful deployment of the system. The methodology begins with clearly defining the project scope and objectives, which include identifying the specific goals such as improving document digitization processes or automating data extraction from images. Requirements gathering plays a crucial role in this phase, involving consultations with stakeholders to understand their needs and expectations. This process helps in documenting detailed functional requirements such as text recognition accuracy, language support, and desired output formats.

Once the requirements are established, the next step is technology selection, where appropriate OCR engines, programming languages, and libraries are chosen based on the project's specific needs and considerations like accuracy, scalability, and integration capabilities. The design phase follows, focusing on creating the system architecture and user interface. This includes designing modules for image preprocessing, text recognition, document layout analysis, and output generation, alongside developing a user-friendly interface for seamless interaction and management of OCR results.

Implementation involves the actual development and integration of chosen technologies and components. Image preprocessing algorithms are implemented to enhance image quality, reduce noise, and correct skewing, all crucial for improving OCR accuracy. Text recognition algorithms and OCR libraries are integrated to accurately identify and extract characters from images, considering factors like different fonts, languages, and document layouts. Output modules are developed to generate searchable PDFs, plain text files, or other specified formats as required.

Testing and evaluation are critical stages to ensure the OCR system's reliability and performance. This includes rigorous unit testing of individual components, integration testing to verify seamless operation across modules, and performance testing to assess speed, accuracy, and resource usage under various conditions. Deployment strategies are planned to roll out the OCR system into production environments, ensuring scalability and integration with existing systems or workflows.

Training and documentation are essential components post-deployment, providing user training sessions and comprehensive documentation on system usage, maintenance procedures, and troubleshooting guides. Continuous maintenance and support involve monitoring system performance, addressing bugs or issues, and implementing updates to enhance functionality and security. Evaluation through user feedback and performance assessments helps in identifying areas for improvement and future enhancements, ensuring the OCR system continues to meet evolving business needs effectively.

4.1 System Design Life Cycle

The System Design Life Cycle (SDLC) is a methodical approach to developing and maintaining systems that ensures systematic planning, design, implementation, and support. It begins with defining the system's goals and requirements in the Conceptual Design phase, followed by detailed planning and architecture in the System Requirements Analysis and System Architectural Design phases, respectively. Implementation involves coding and integrating components, followed by rigorous testing to validate functionality in the Testing phase. Deployment and Integration then transition the system into operation, with ongoing Maintenance to support, optimize, and update the system throughout its lifecycle. This structured approach ensures the system meets user needs, quality standards, and evolves effectively over time.

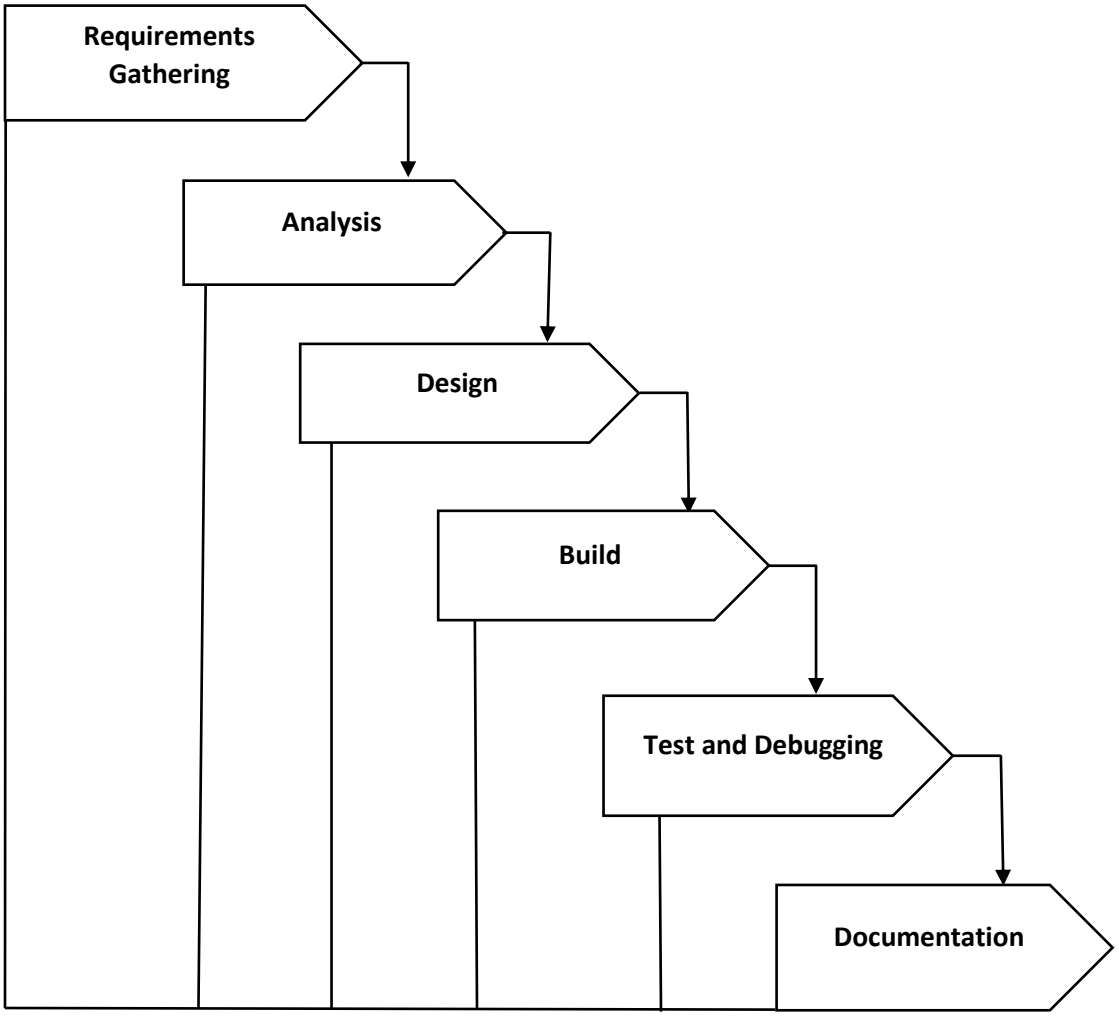


Figure 1: Waterfall Model

4.2 Use Case Diagram

A use case diagram is a visual representation that depicts how users interact with a system and its various functionalities. It consists of actors (users or external systems) and use cases (actions or services provided by the system). Actors are represented as stick figures, and use cases are shown as ovals. Lines between actors and use cases illustrate the interactions or relationships, indicating which actors are involved in each use case. Use case diagrams are valuable for understanding system requirements, defining boundaries, and visualizing the flow of interactions between users and the system's functionalities in a clear and concise manner. They serve as a blueprint for designing systems and guiding development efforts by focusing on user goals and system behaviors.

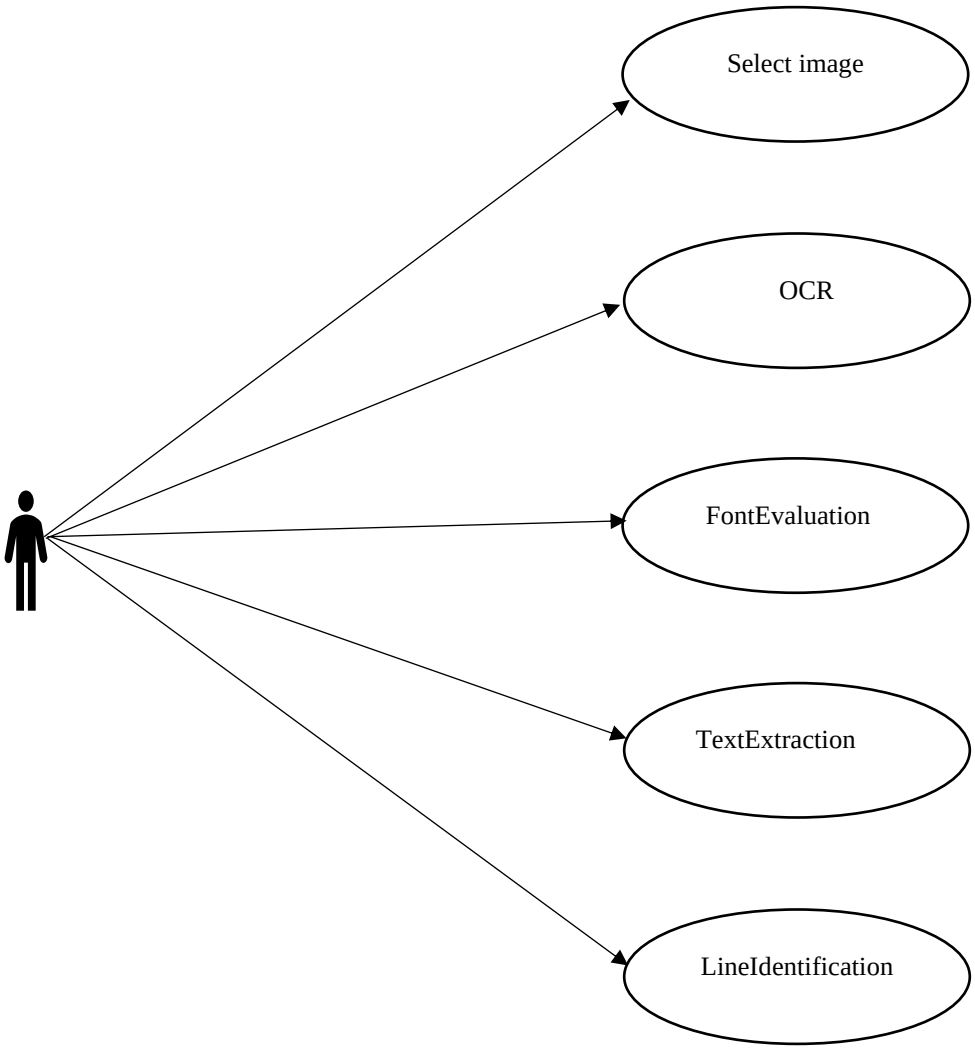


Figure 3: Use Case Diagram

4.3 ER Diagram

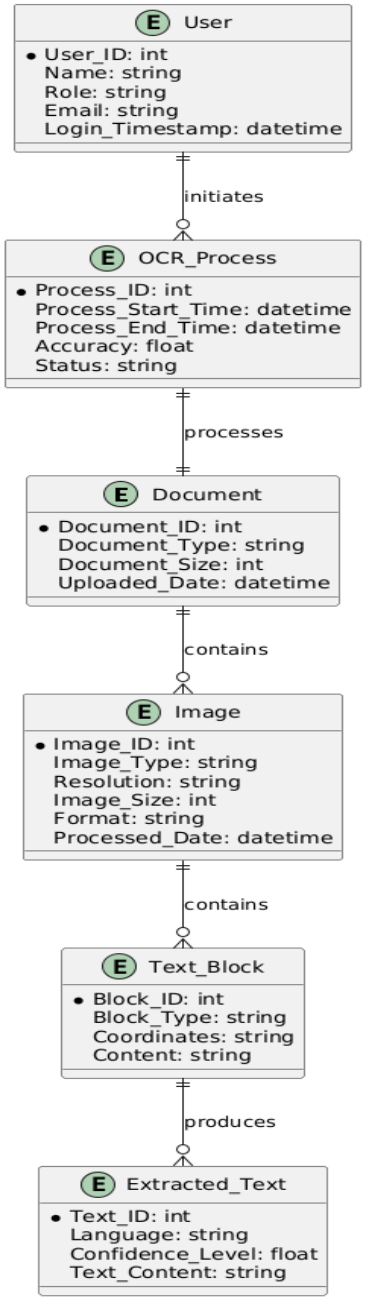
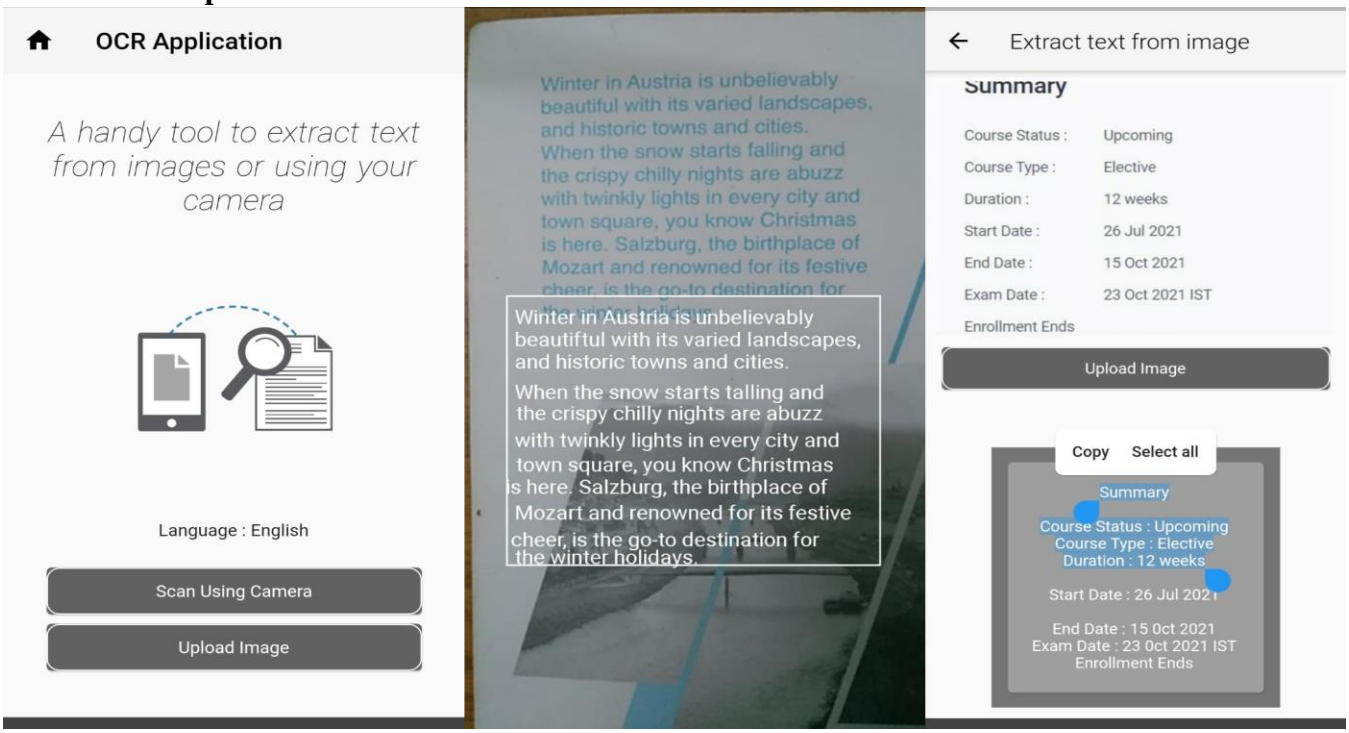
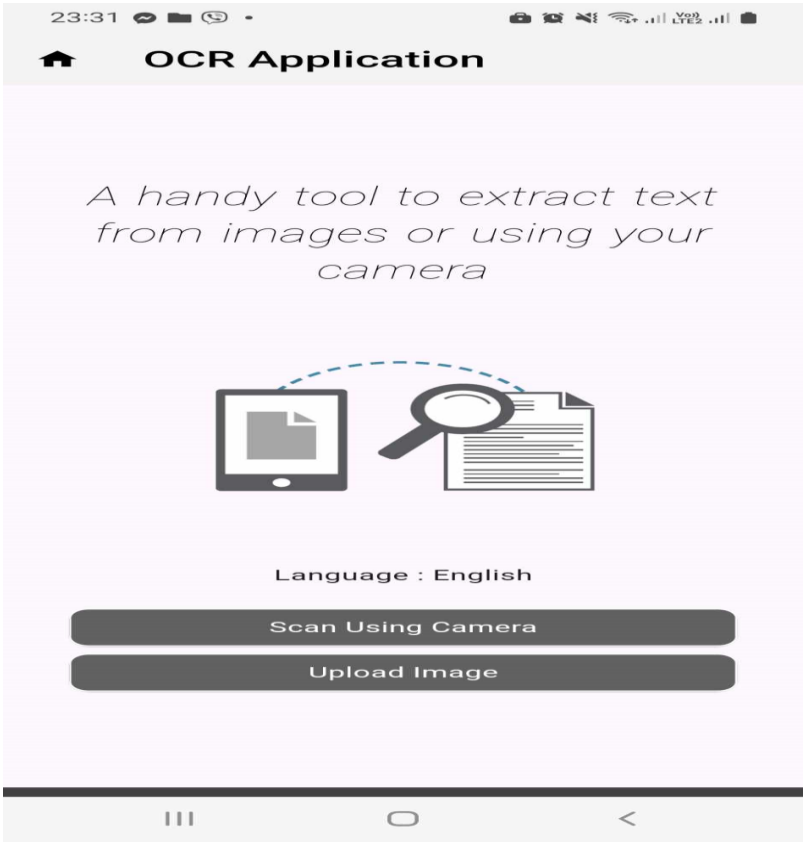
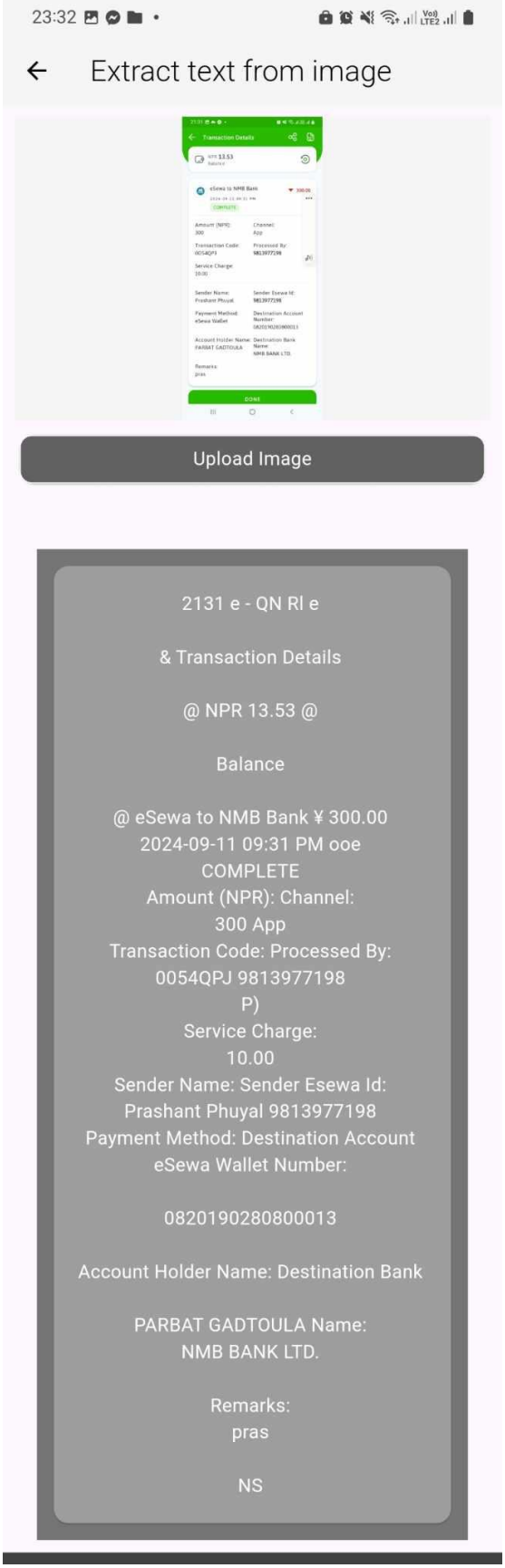


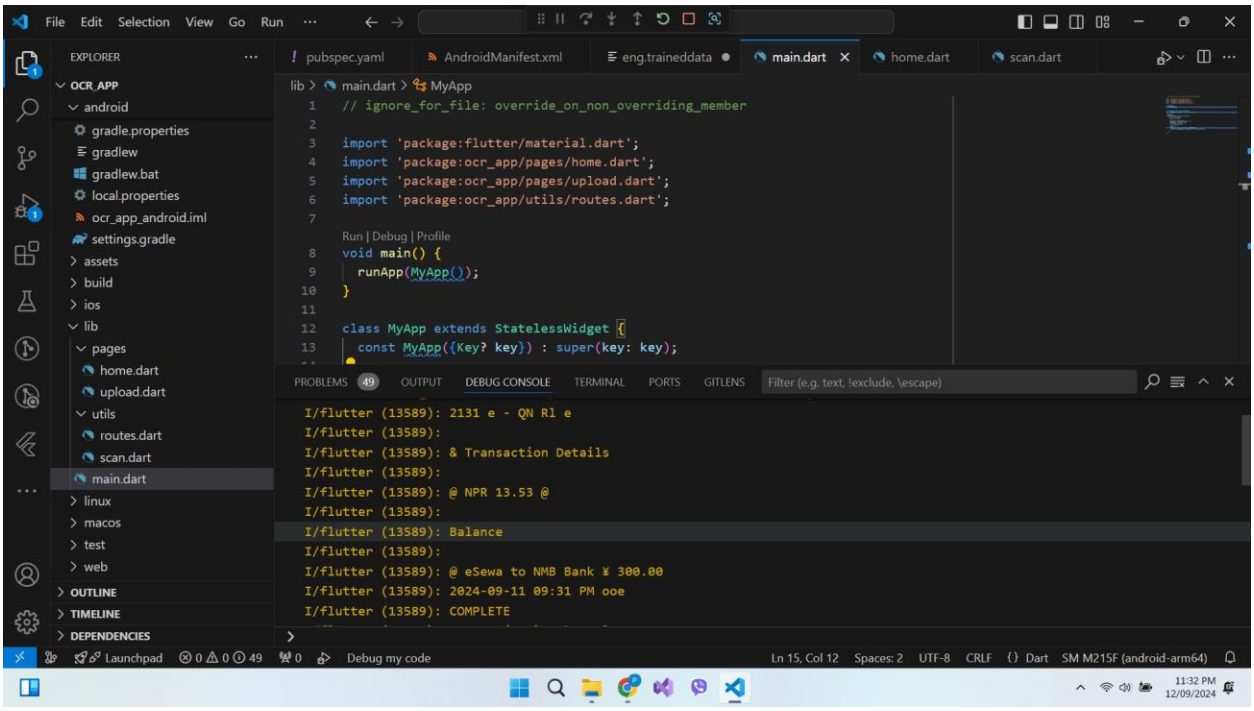
Figure 4:ER Diagram

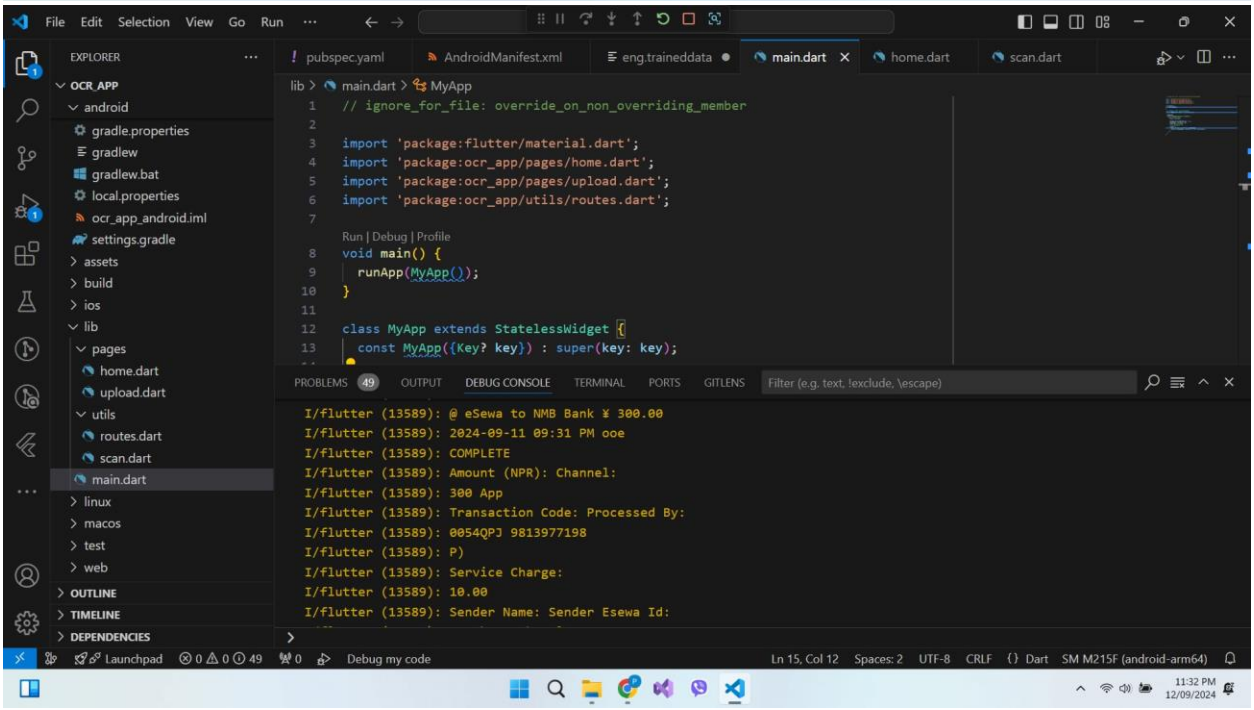
4.4 Work Completed

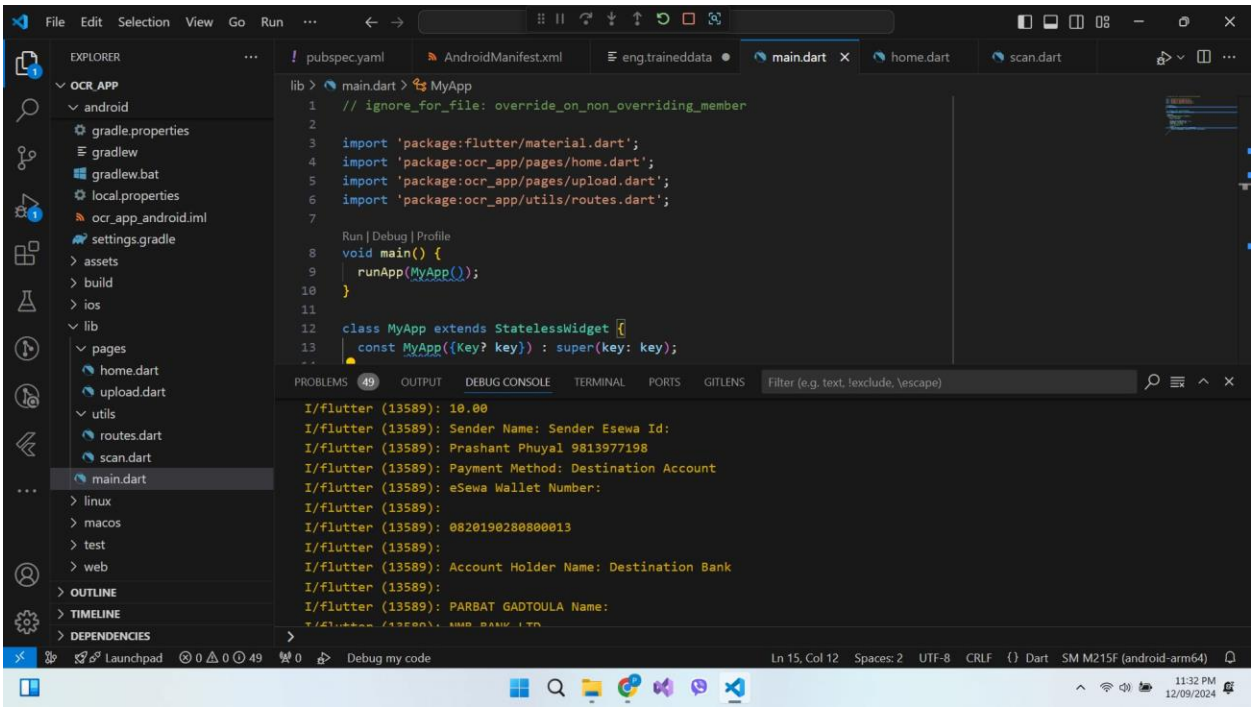


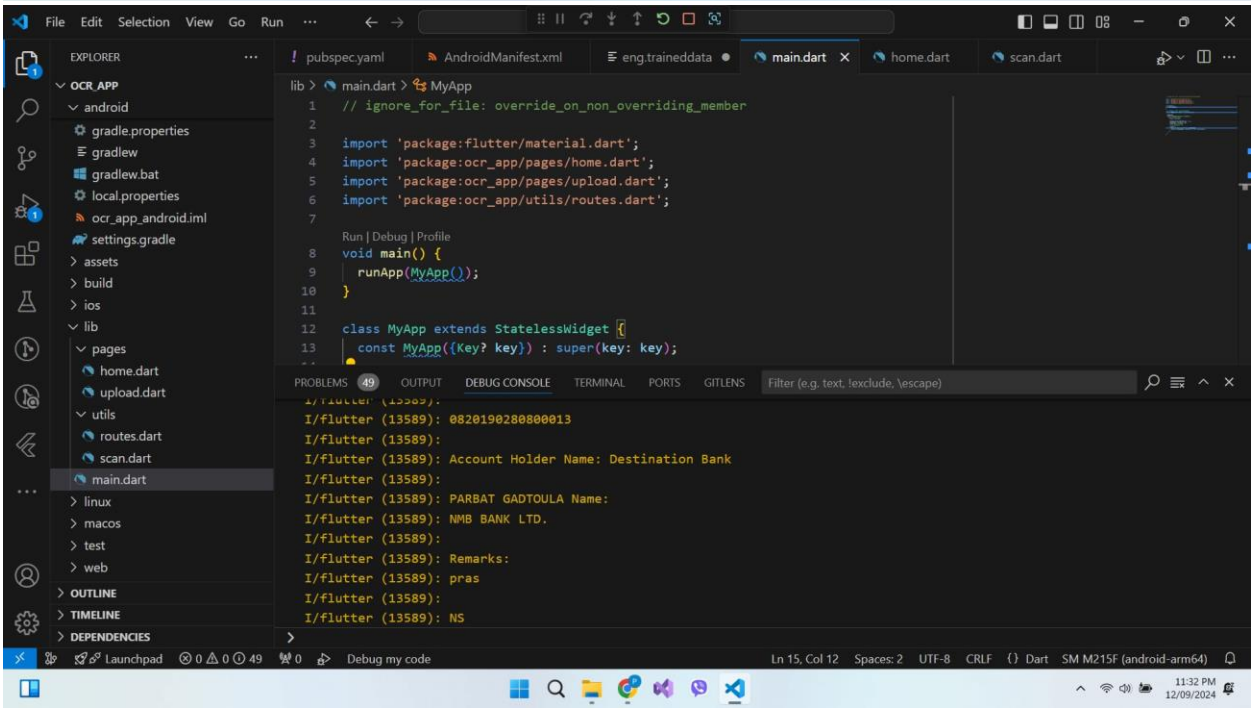


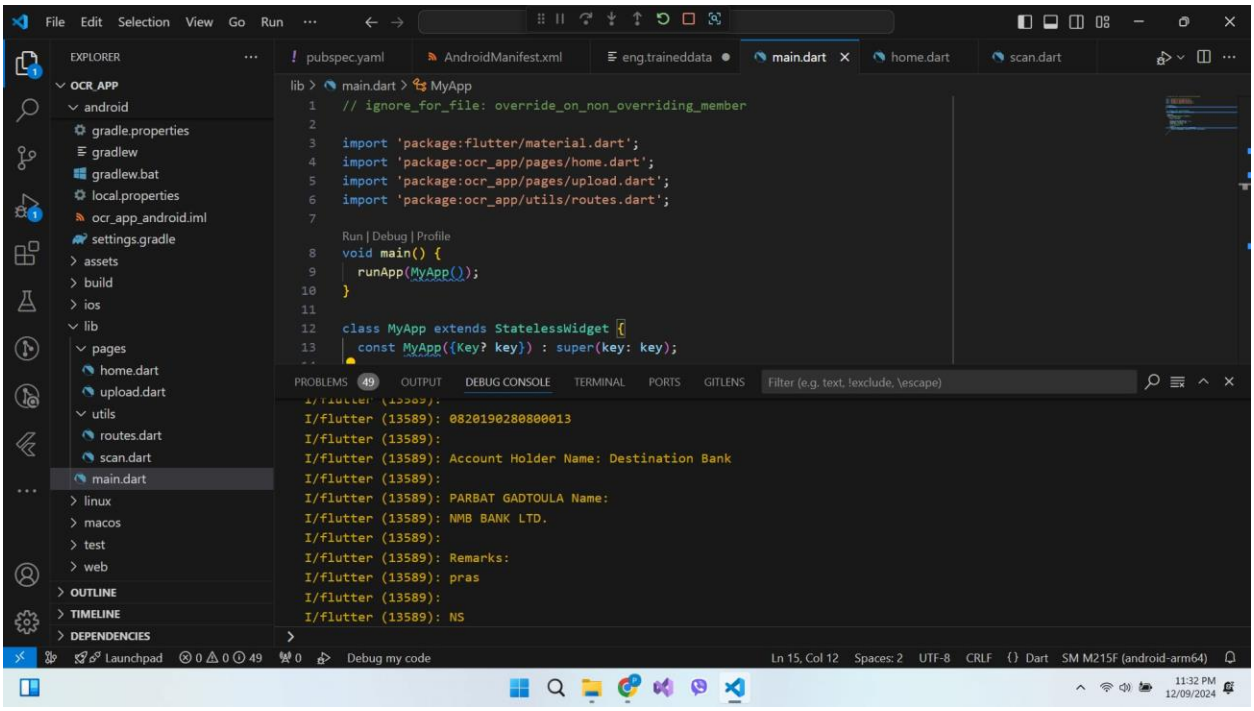

23:32
Extract text from image
Upload Image
2131 e - QN Rl e
& Transaction Details
@ NPR 13.53 @
Balance
@ eSewa to NMB Bank ¥ 300.00
2024-09-11 09:31 PM ooe
COMPLETE
Amount (NPR): Channel:
300 App
Transaction Code: Processed By:
0054QPJ 9813977198
P)
Service Charge:
10.00
Sender Name: Sender Esewa Id:
Prashant Phuyal 9813977198
Payment Method: Destination Account
eSewa Wallet Number:
0820190280800013
Account Holder Name: Destination Bank
PARBAT GADTOULA Name:
NMB BANK LTD.
Remarks:
pras
NS


File Edit Selection View Go Run
EXPLORER
OCR_APP
android
gradle.properties
gradlew
gradlew.bat
local.properties
ocr_app_android.iml
settings.gradle
assets
build
ios
lib
pages
home.dart
upload.dart
utils
routes.dart
scan.dart
main.dart
linux
macos
test
web
OUTLINE
TIMELINE
DEPENDENCIES
pubspec.yaml
AndroidManifest.xml
eng.traineddata
main.dart
home.dart
scan.dart
lib > main.dart > MyApp
// ignore_for_file: override_on_non_overriding_member
import 'package:flutter/material.dart';
import 'package:ocr_app/pages/home.dart';
import 'package:ocr_app/pages/upload.dart';
import 'package:ocr_app/utils/routes.dart';
Run | Debug | Profile
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
const MyApp({Key? key}) : super(key: key);
PROBLEMS 49
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Filter (e.g. text, !exclude, \escape)
I/flutter (13589): 2131 e - QN Rl e
I/flutter (13589):
I/flutter (13589): & Transaction Details
I/flutter (13589):
I/flutter (13589): @ NPR 13.53 @
I/flutter (13589):
I/flutter (13589): Balance
I/flutter (13589):
I/flutter (13589): @ eSewa to NMB Bank ¥ 300.00
I/flutter (13589): 2024-09-11 09:31 PM ooe
I/flutter (13589): COMPLETE
Launchpad
Debug my code
Ln 15, Col 12
Spaces: 2
UTF-8
CRLF
Dart
SM M215F (android-arm64)
11:32 PM
12/09/2024


File Edit Selection View Go Run
EXPLORER
OCR_APP
android
gradle.properties
gradlew
gradlew.bat
local.properties
ocr_app_android.iml
settings.gradle
assets
build
ios
lib
pages
home.dart
upload.dart
utils
routes.dart
scan.dart
main.dart
linux
macos
test
web
OUTLINE
TIMELINE
DEPENDENCIES
pubspec.yaml
AndroidManifest.xml
eng.traineddata
main.dart
home.dart
scan.dart
lib > main.dart > MyApp
// ignore_for_file: override_on_non_overriding_member
import 'package:flutter/material.dart';
import 'package:ocr_app/pages/home.dart';
import 'package:ocr_app/pages/upload.dart';
import 'package:ocr_app/utils/routes.dart';
Run | Debug | Profile
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
const MyApp({Key? key}) : super(key: key);
PROBLEMS 49
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Filter (e.g. text, !exclude, \escape)
I/flutter (13589): @ eSewa to NMB Bank ¥ 300.00
I/flutter (13589): 2024-09-11 09:31 PM ooe
I/flutter (13589): COMPLETE
I/flutter (13589): Amount (NPR): Channel:
I/flutter (13589): 300 App
I/flutter (13589): Transaction Code: Processed By:
I/flutter (13589): 0054QPJ 9813977198
I/flutter (13589): P)
I/flutter (13589): Service Charge:
I/flutter (13589): 10.00
I/flutter (13589): Sender Name: Sender Esewa Id:
Launchpad
Debug my code
Ln 15, Col 12
Spaces: 2
UTF-8
CRLF
Dart
SM M215F (android-arm64)
11:32 PM
12/09/2024


File Edit Selection View Go Run
EXPLORER
OCR_APP
android
gradle.properties
gradlew
gradlew.bat
local.properties
ocr_app_android.iml
settings.gradle
assets
build
ios
lib
pages
home.dart
upload.dart
utils
routes.dart
scan.dart
main.dart
linux
macos
test
web
OUTLINE
TIMELINE
DEPENDENCIES
pubspec.yaml
AndroidManifest.xml
eng.traineddata
main.dart
home.dart
scan.dart
lib > main.dart > MyApp
1 // ignore_for_file: override_on_non_overriding_member
2
3 import 'package:flutter/material.dart';
4 import 'package:ocr_app/pages/home.dart';
5 import 'package:ocr_app/pages/upload.dart';
6 import 'package:ocr_app/utils/routes.dart';
7
Run | Debug | Profile
8 void main() {
9 runApp(MyApp());
10 }
11
12 class MyApp extends StatelessWidget {
13 const MyApp({Key? key}) : super(key: key);
PROBLEMS 49
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Filter (e.g. text, !exclude, \escape)
I/flutter (13589): 10.00
I/flutter (13589): Sender Name: Sender Esewa Id:
I/flutter (13589): Prashant Phuyal 9813977198
I/flutter (13589): Payment Method: Destination Account
I/flutter (13589): eSewa Wallet Number:
I/flutter (13589):
I/flutter (13589): 0820190280800013
I/flutter (13589):
I/flutter (13589): Account Holder Name: Destination Bank
I/flutter (13589):
I/flutter (13589): PARBAT GADTOULA Name:
Launchpad
Debug my code
Ln 15, Col 12
Spaces: 2
UTF-8
CRLF
Dart
SM M215F (android-arm64)
11:32 PM
12/09/2024


File Edit Selection View Go Run
EXPLORER
OCR_APP
android
gradle.properties
gradlew
gradlew.bat
local.properties
ocr_app_android.iml
settings.gradle
assets
build
ios
lib
pages
home.dart
upload.dart
utils
routes.dart
scan.dart
main.dart
linux
macos
test
web
OUTLINE
TIMELINE
DEPENDENCIES
pubspec.yaml
AndroidManifest.xml
eng.traineddata
main.dart
home.dart
scan.dart
lib > main.dart > MyApp
1 // ignore_for_file: override_on_non_overriding_member
2
3 import 'package:flutter/material.dart';
4 import 'package:ocr_app/pages/home.dart';
5 import 'package:ocr_app/pages/upload.dart';
6 import 'package:ocr_app/utils/routes.dart';
7
Run | Debug | Profile
8 void main() {
9 runApp(MyApp());
10 }
11
12 class MyApp extends StatelessWidget {
13 const MyApp({Key? key}) : super(key: key);
PROBLEMS 49
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Filter (e.g. text, !exclude, \escape)
I/flutter (13589): 0820190280800013
I/flutter (13589):
I/flutter (13589): Account Holder Name: Destination Bank
I/flutter (13589):
I/flutter (13589): PARBAT GADTOULA Name:
I/flutter (13589): NMB BANK LTD.
I/flutter (13589):
I/flutter (13589): Remarks:
I/flutter (13589): pras
I/flutter (13589):
I/flutter (13589): NS
Launchpad
Debug my code
Ln 15, Col 12
Spaces: 2
UTF-8
CRLF
Dart
SM M215F (android-arm64)
11:32 PM
12/09/2024


File Edit Selection View Go Run
EXPLORER
OCR_APP
android
gradle.properties
gradlew
gradlew.bat
local.properties
ocr_app_android.iml
settings.gradle
assets
build
ios
lib
pages
home.dart
upload.dart
utils
routes.dart
scan.dart
main.dart
linux
macos
test
web
OUTLINE
TIMELINE
DEPENDENCIES
pubspec.yaml
AndroidManifest.xml
eng.traineddata
main.dart
home.dart
scan.dart
lib > main.dart > MyApp
// ignore_for_file: override_on_non_overriding_member
import 'package:flutter/material.dart';
import 'package:ocr_app/pages/home.dart';
import 'package:ocr_app/pages/upload.dart';
import 'package:ocr_app/utils/routes.dart';
Run | Debug | Profile
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
const MyApp({Key? key}) : super(key: key);
PROBLEMS 49
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Filter (e.g. text, !exclude, \escape)
I/flutter (13589): 0820190280800013
I/flutter (13589):
I/flutter (13589): Account Holder Name: Destination Bank
I/flutter (13589):
I/flutter (13589): PARBAT GADTOULA Name:
I/flutter (13589): NMB BANK LTD.
I/flutter (13589):
I/flutter (13589): Remarks:
I/flutter (13589): pras
I/flutter (13589):
I/flutter (13589): NS
Launchpad
Debug my code
Ln 15, Col 12
Spaces: 2
UTF-8
CRLF
Dart
SM M215F (android-arm64)
11:32 PM
12/09/2024

CHAPTER 5

5 Time Schedule

Figure 2: Gantt Chart

CHAPTER 6

6 Bibliography

- **https://en.wikipedia.org/wiki/Optical_character_recognition#:~:text=Optical%20character%20recognition%20or%20optical,billboards%20in%20a%20landscape%20photo)**
- **https://www.scribd.com/document/136702462/uml-fo-ocr**
- **https://basilchackomathew.medium.com/best-ocr-tools-in-python-4f16a9b6b116**