OPTIPARSE | YOUR FINANCE BUDDY

```
  ___        _   _ ____
 / _ \ _ __ | |_(_)  _ \ __ _ _ __ ___  ___
| | | | '_ \| __| | |_) / _` | '__/ __|/ _ \
| |_| | |_) | |_| |  __/ (_| | |  \__ \  __/
 \___/| .__/ \__|_|_|   \__,_|_|  |___/\___|
      |_|
```

Developing an Android Application with Integrated OCR model for Financial Data Extraction and Transaction History Management

Table of Contents

- Overview
- Functionality and ML Models
  - Preprocessing and OCR for text extraction
  - Multi-prompt LLM for data structuring
  - Data Point Accuracy Evaluation
- User Interface and Experience
  - UI Design
  - User Experience and intuitiveness
  - Search and Filter Functionality
- Server Implementation
  - API Design
  - Backend Implementation
  - Dockerization
- Possible Improvements for Future

Explained via App Demonstration

Overview and Tech Stack

Android Application with Integrated OCR model for Financial Data Extraction and Transaction History Management

- Frontend developed in **Flutter** for cross-platform implementation
- **FastAPI** used for API development and integration of frontend with backend. Preferred over other frameworks as it’s fast and lightweight
- **SQLite** database used for information storage as querying of tabulated data is easier and faster. Moreover it seamlessly integrates with FastAPI
- **PaddlePaddle** and **Gemini LLM** used for Machine Learning Applications
- **Multi-Prompt aggregation** implemented in LLM querying in order to maximize confidence

Functionality and ML Models

Optical Character Recognition

Preprocessing

- **Hurdles**:
  - Lack of generalizability (Inconsistent Document Quality)
  - Complex Layouts
  - Unpredictable inputs due to varying camera qualities
  - Artifacts Introduced by Preprocessing

- **Iterations**:
  - Global Thresholding: ineffective for documents with uneven lighting or variable contrast.
  - Local Thresholding: ineffective with complex backgrounds, noise, non-text elements like logos or watermarks. Adds unwanted noise.
  - Histogram Normalization: Makes text harder to distinguish from the background, especially in documents with poor-quality scans or varying text density.

Optical Character Recognition

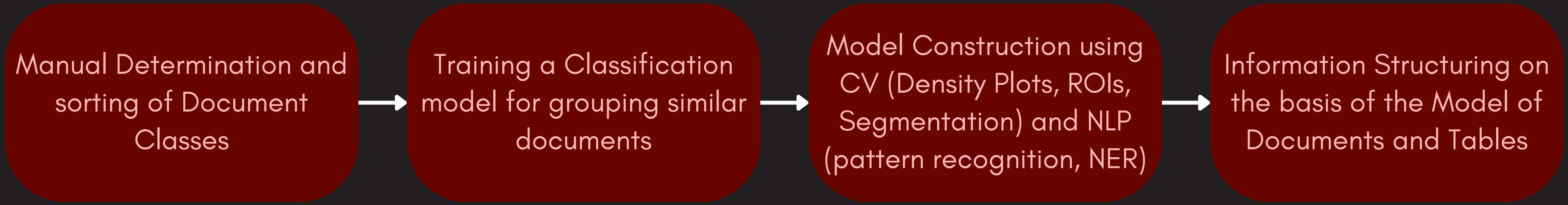
Model Selection

PaddleOCR	EasyOCR	TesseractOCR
Differentiable Binarization Net for text detection, which is more effective in detecting irregular text regions and complex layouts	CRNN model but lacks the flexibility and modularity in order to detect complex fonts, styles and structures	Uses connected component analysis along with LSTM, which is less accurate for complex images
Extremely fast due to lightweight backbones like MobileNet	Moderately Fast, but extremely slow for large and complex images	Extremely Slow

Information processing via LLMs

Algorithm Determination | Initial Approach

Development Pipeline



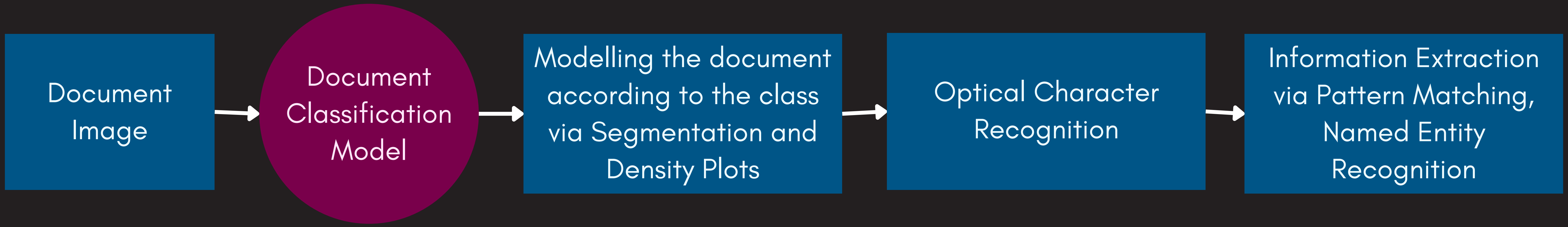
Manual Determination and sorting of Document Classes

Training a Classification model for grouping similar documents

Model Construction using CV (Density Plots, ROIs, Segmentation) and NLP (pattern recognition, NER)

Information Structuring on the basis of the Model of Documents and Tables

Inference Pipeline



Document Image

Document Classification Model

Modelling the document according to the class via Segmentation and Density Plots

Optical Character Recognition

Information Extraction via Pattern Matching, Named Entity Recognition

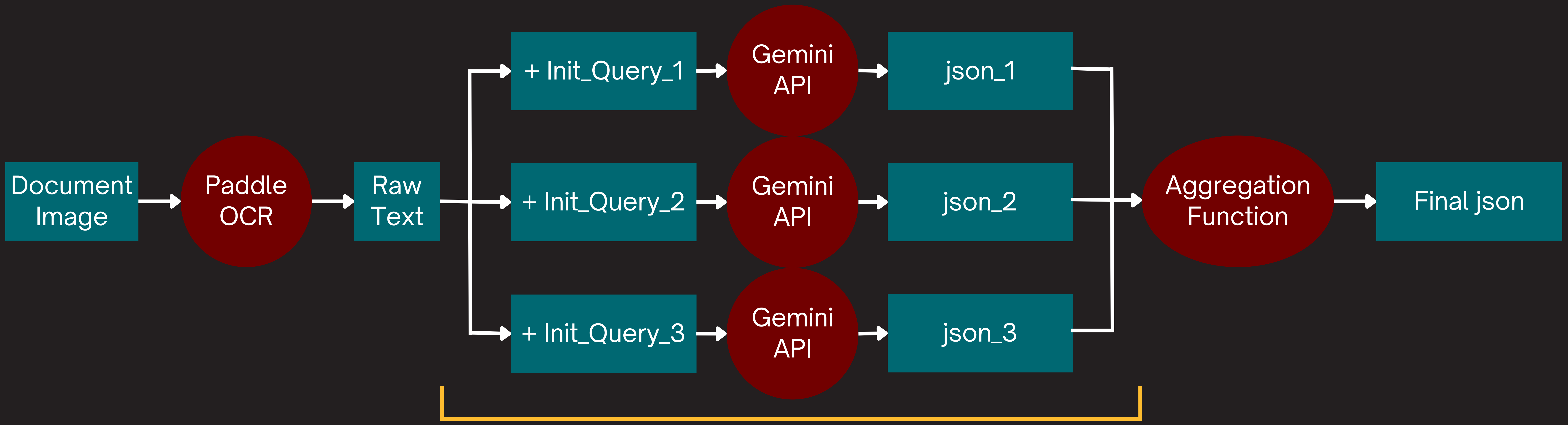
Information processing via LLMs

Algorithm Determination | Hurdles and Drawbacks

- No Set Document Class, or generalized format of documents
- Even if sub-models such as Tables, Bill Headers, Summaries etc are common structures, their individual formats differ widely
- Primitive Techniques like NER, Density Plots etc are not generalizable and robust, especially for complex financial documents
- **Amplification of error** due to propagation across multiple steps of the pipeline

Information processing via LLMs

Gemini 1.5 Flash with multi-prompt output aggregation



Parallel Processing to Reduce time

Data Point Accuracy Evaluation

Data Point	Correct Inferences	Total Inferences	Accuracy %
Merchant Name	57	64	0.890625
Merchant ID/ code	44	64	0.6875
Address of merchant	58	64	0.90625
Phone number of merchant	44	64	0.6875
email id of merchant	6	64	0.09375
FAX of merchant	13	64	0.203125
Invoice/Bill/Receipt number	57	64	0.890625
GST Registration Number	52	64	0.8125
GST %	41	64	0.640625
Identification Number	12	64	0.1875

OPTIPARSE | YOUR FINANCE BUDDY

Data Point	Correct Inferences	Total Inferences	Accuracy %
Date	58	64	0.90625
Month	58	64	0.90625
Year	58	64	0.90625
Time	51	64	0.796875
Class of financial document	64	64	1
Type of item purchased	47	64	0.734375
Total amount	61	64	0.953125
Cashier name	29	64	0.453125
Customer name	5	64	0.078125
Cust ID	9	64	0.140625
Number of items	57	64	0.890625
TOTAL ACCURACY			13.765625

Score Evaluation

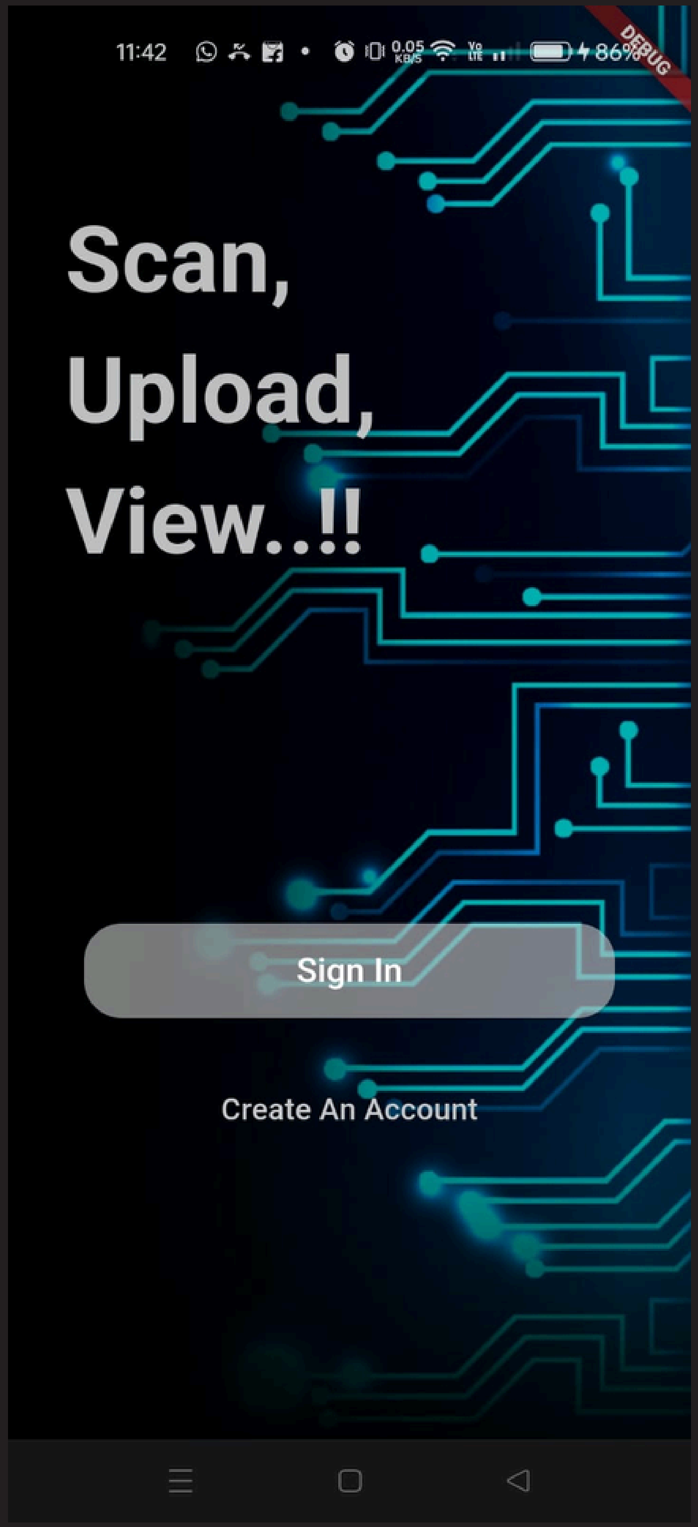
$$\text{Score} = \frac{\text{Sum of accuracy of all data points}}{(1+\text{Avg inference time in sec})*10}$$

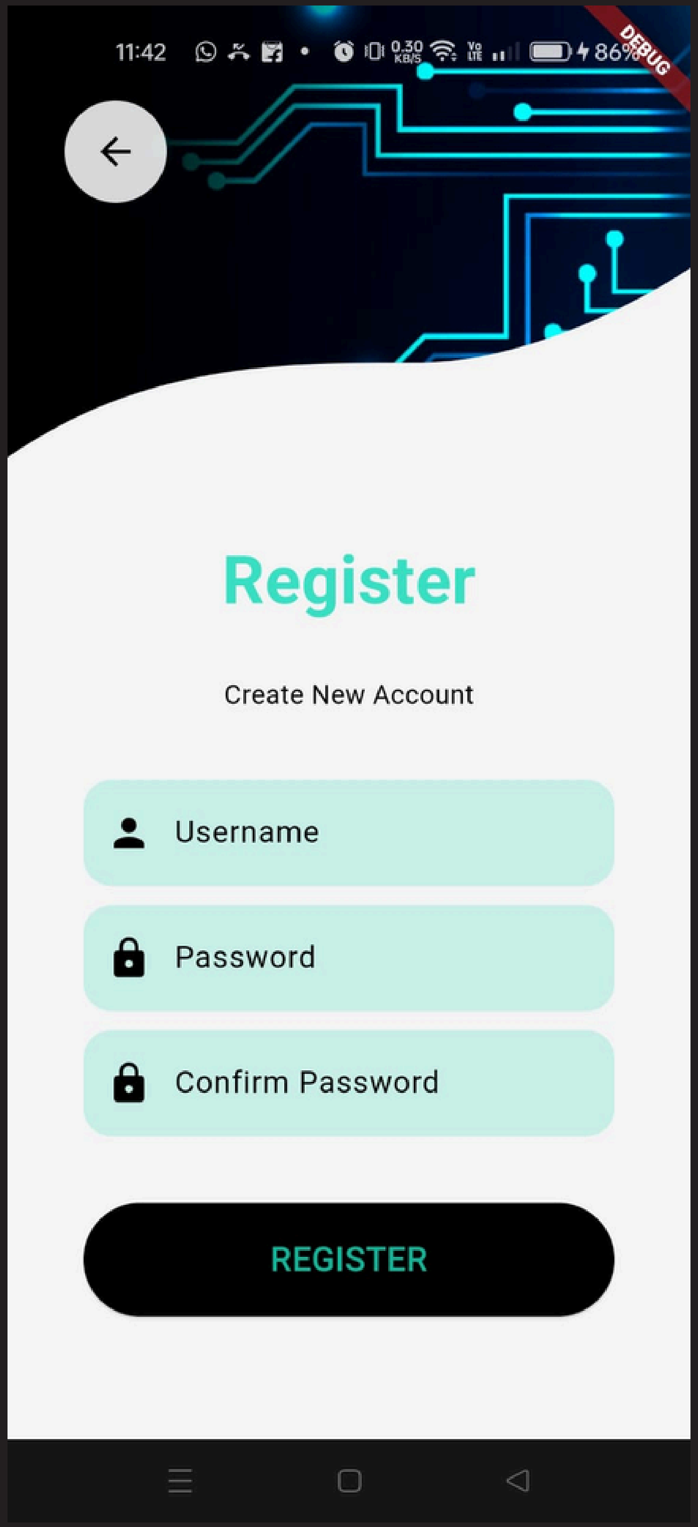
$$= \frac{13.7656}{(1+0.2989)*10}$$

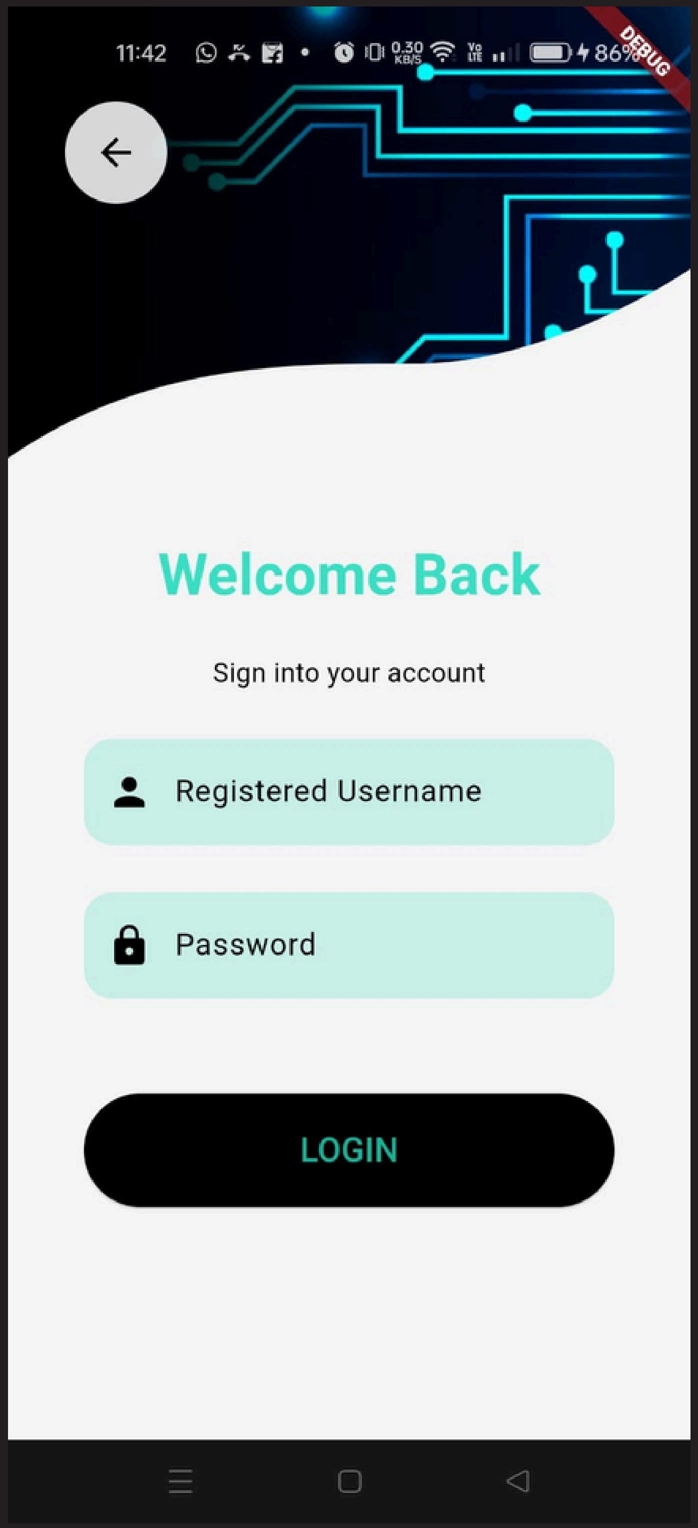
$$= \frac{13.7656}{(1+0.2989)*10} = \boxed{1.05978905228}$$

User Interface and Experience

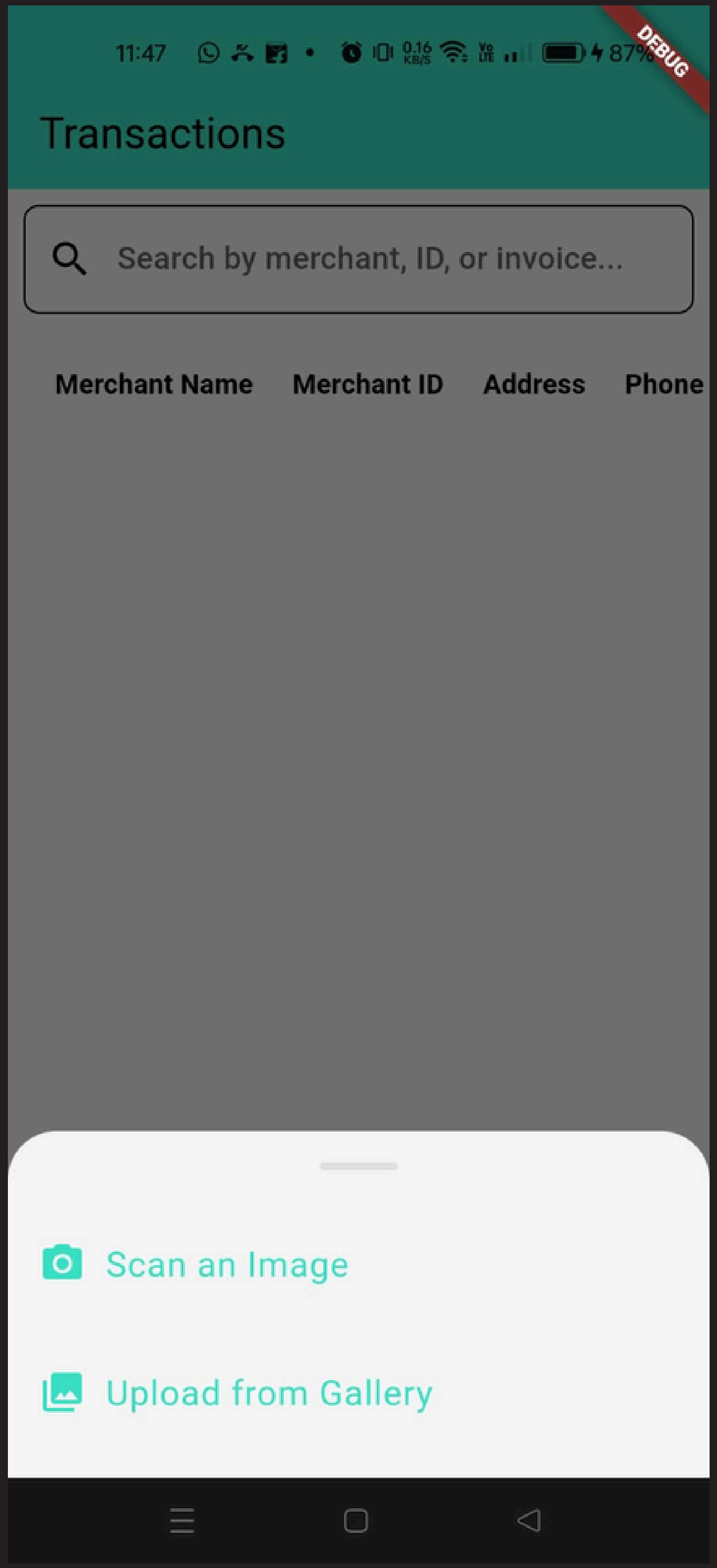
OPTIPARSE | YOUR FINANCE BUDDY

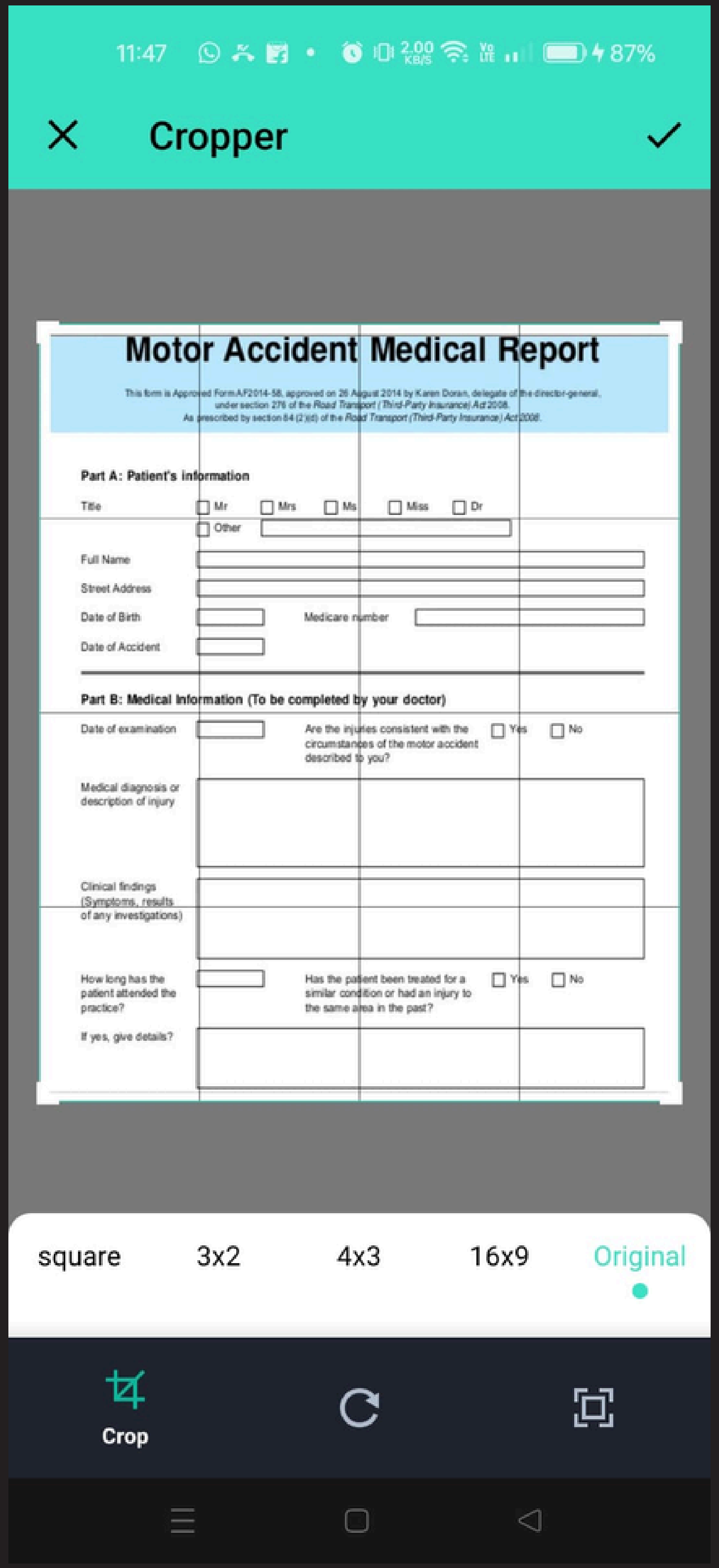






OPTIPARSE | YOUR FINANCE BUDDY





Motor Accident Medical Report

This form is Approved Form AF2014-58, approved on 26 August 2014 by Karen Doran, delegate of the director-general, under section 276 of the *Road Transport (Third-Party Insurance) Act 2008*.
As prescribed by section 84 (2)(d) of the *Road Transport (Third-Party Insurance) Act 2008*.

Part A: Patient's information

Title □ Mr □ Mrs □ Ms □ Miss □ Dr
□ Other

Full Name

Street Address

Date of Birth

Medicare number

Date of Accident

Part B: Medical Information (To be completed by your doctor)

Date of examination

Are the injuries consistent with the circumstances of the motor accident described to you? □ Yes □ No

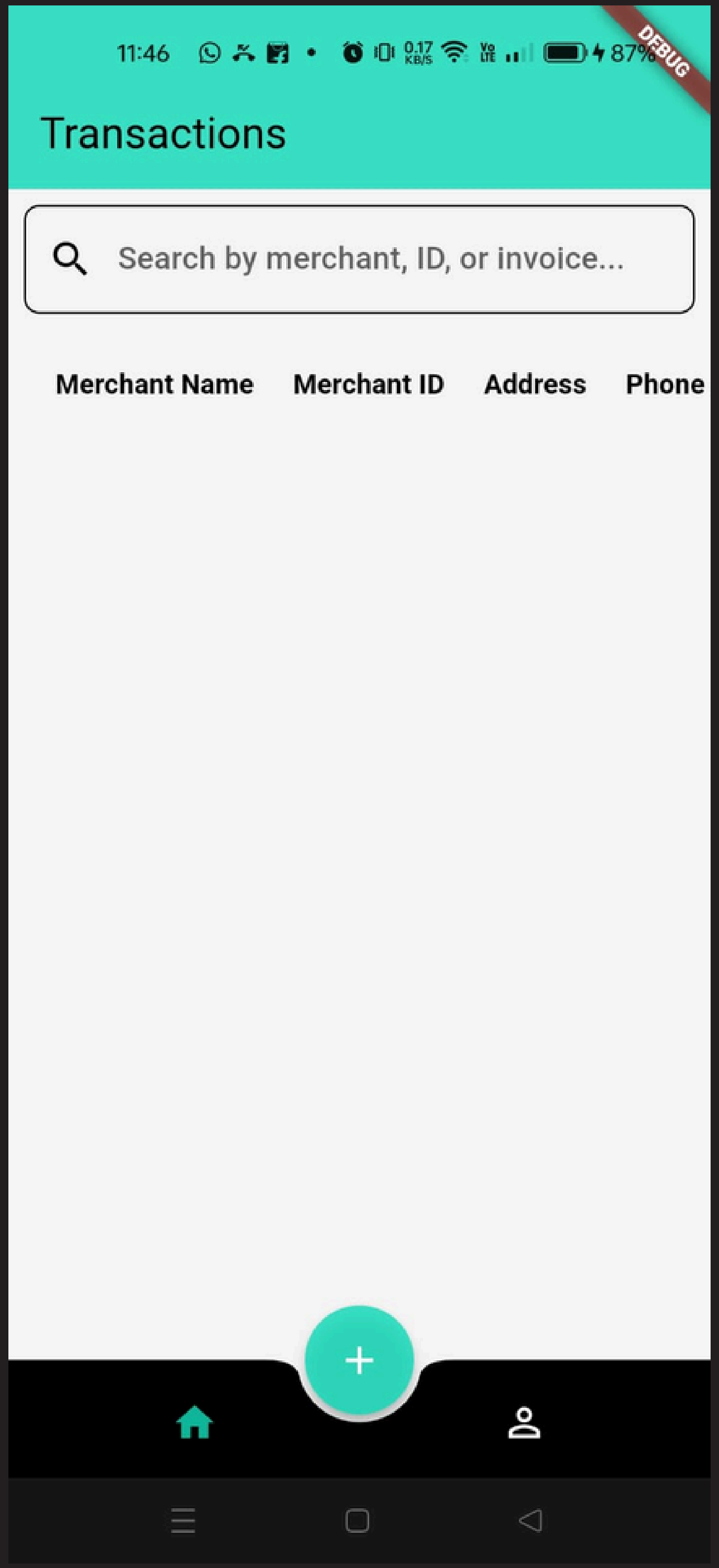
Medical diagnosis or description of injury

Clinical findings (Symptoms, results of any investigations)

How long has the patient attended the practice?

Has the patient been treated for a similar condition or had an injury to the same area in the past? □ Yes □ No

If yes, give details?



Server Implementation

Implementing the Backend

FastAPI | SQLite3

- We chose FastAPI for implementing the server API’s due to it being lightweight and robust.
- We processed information on the backend in Python, and used SQLite3 for storing data on the server.
- API Routes:
    - /signup
        - Creates a user by accepting a username and password, and stores the hash in the database for authentication.
    - /token
        - Implements OAuth2 based authentication system, generating a JWT token with an expiry, which is needed for every query apart from signup and login, to authenticate the user.

Implementing the Backend

FastAPI | SQLite3

... continued

- API Routes:
  - /update_details
    - Allows the user to update his information, such as account number, email address etc.
  - /query
    - API which allows image upload, and processes the image via the backend pipeline, fetches the extracted transaction data and stores it on the backend database, so that user can fetch the data and view at any time.

Implementing the Backend

FastAPI | SQLite3

... continued

- API Routes:
  - /transaction_data
    - Decodes the token to fetch the user’s details and return the transaction data, optionally taking arguments for the number of transactions to return and the offset.

Dockerization

We setup Dockerization using a simple Docker file to setup the database, install the requirements and launch the Web App, to create a container for deployment

Possible Improvements for Future

1. Implementation of multi modality in data structuring along with multi query aggregation order to get diverse range of outputs.
2. Improvement of aggregation function to include bias for more accurate models and queries.
3. Implementing Diverse range of filters and search in UI (along with the basic filtering in status quo) in order to make UX better
4. Integration with bank accounts in order to keep comprehensive and complete history tracking