EXPENSE MANAGER: AN EXPENSE TRACKING APPLICATION USING IMAGE PROCESSING

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering in Information Technology

By

NUPUR SAWARKAR (18101B0030)

PRANAY YENAGANDULA (18101B0032)

DEVANG SHETYE (18101B0068)

Under the Guidance of

Prof. SHRUTI AGRAWAL

Department of Information Technology

Vidyalankar Institute of Technology

Wadala(E), Mumbai-400437

University of Mumbai

2021-22

CERTIFICATE OF APPROVAL

This is to certify that the project entitled

"EXPENSE MANAGER: AN EXPENSE TRACKING APPLICATION"

is a bonafide work of

NUPUR SAWARKAR (18101B0030)
PRANAY YENAGANDULA (18101B0032)
DEVANG SHETYE (18101B0068)

submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the

degree of

Undergraduate in **"INFORMATION TECHNOLOGY"**.

Guide	Head of Department	Principal
(Prof. Shruti Agrawal)	(Prof. Vipul Dalal)	(Dr. S.A. Patekar)

Project Report Approval for B. E.

This project report entitled ***EXPENSE MANAGER: AN EXPENSE TRACKING APPLICATION*** by

1. ***NUPUR SAWARKAR (18101B0030)***
2. ***PRANAY YENAGANDULA (18101B0032)***
3. ***DEVANG SHETYE (18101B0068)***

is approved for the degree of ***Bachelor of Engineering in Information Technology.***

Examiners

1._____-

2._____-

Date:

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of student	Roll No.	Signature
1. Nupur Sawarkar	18101B0030	
2. Pranay Yenagandula	18101B0032	
3. Devang Shetye	18101B0068	

Date:

ACKNOWLEDGEMNT

We are honored to present "**Expense Manager: An Expense Tracking Application**" as our B.E Final Year Project.We are using this opportunity to express our profound gratitude to our principal "**Dr. Sunil Patekar**" for providing us with proper facilities and support.

We express our deepest gratitude towards our HOD "**Prof. Vipul Dalal**" for his valuable and timely advice during the various phases in our project. We would like to thank our project guide **"Prof. Shruti Agrawal"** for support, patience and faith in our capabilities and for giving us flexibility in terms of working and reporting schedules. Finally, we would like to thank everyone who have helped us directly or indirectly in our project.

We would also like to thank our staff members and lab assistant for permitting us to use computer in the lab as and when required. We thank our college for providing us with excellent facilities that helped us to complete and present this project.

Nupur Sawarkar
Pranay Yenagandula
Devang Shetye

ABSTRACT

In life cycle of human after birth the requirement of materials and belongings is clear, so as to fulfil our needs and desire, we buy goods. The rule of earth is that you simply must have money in order to shop for desired good. So, in this way the method of earning and spending goes on in our life.

Hence, in today's busy and expensive life, we are in an exceedingly great rush to create money. But at the end of the month, we broke off. As we are unknowingly spending on little and unwanted things. So, we've intercommunicated with the concept to trace our earnings. Our project will help everyone who are getting to know their expenses and save from it. The user is going to be given the ability to set a monthly limit and if the user crosses that limit our app will notify the user about the identical. The user can give receipts as an input, using AI our app will split it into different categories. Here user can define their own categories for expense type like food, clothing, rent and bills and therefore the user may also set limits for a selected category. User are going to be able to see chart of expenses by transaction date or by category. This project isn't indented for a specific user or age bracket but anyone and everybody who wants to trace their expense can use this app.

So, the general idea of this Project is to help people view and study their overall expenditure pattern by developing a mobile application to analyse all the purchases made by the user by simply scanning all the receipts.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

INTRODUCTION

In order to trace their expenses, People generally use traditional paper system to keep the record of their income and expenditures. This sort of traditional system is burdensome and takes longer. So, there must be a management system which must help us to manage our daily earnings and expenses easily, and also helps us to analyze records efficiently. So, we figured out the simplest way to eliminate the normal system with digital, portable, easier and straightforward way to record these data in precisely few clicks with our Android application called “Expense Manager”.

The idea of this Project is to assist people view and study their overall expenditure analysis with the assistance of a mobile app to analyze all the purchases made by the user by simply scanning all the bills and receipts. The User will just have to click a picture of a receipt or choose one from the Gallery and track their payments. Using AI and machine learning, the app is going to be ready to group items category-wise, for example- food, clothes, stationery, etc. The User can monitor his/her expenses and analyze them categorically using graphs and tables, enabling them to higher understand their expense patterns and help them spend intelligently. Using this app, the users also can set monthly limits over specific categories helping them to avoid over-spend on those items. The app will prompt users after they overspend or make repeat purchases. Our main aim is to create our users capable of achieving Personal Life goals by giving them the flexibility to – monitor their expenses; analyze the buying trends; and assess their budget for future transactions. And hence, the required outcome from this app is to provide a higher sense of expenditure to our users.

1.1 Problem Statement

As we know, a lot of people continuously struggle to keep up with their regular expenses and most of the times fail to control or manage their budget. This is often caused because of lack of structural maintenance of their expenses and budget which leads to over expenditure. General survey suggests that 67% of individuals are unaware about where their money goes, 80% are constantly worried and stressed about their finances and most of them spend and save in haphazard manner. Therefore, there is a need for an organized platform to help understand users about their spending's and ensure a better expense pattern in future.

1.2 Scope

The system aims to provide a complete budget management solution where you will get an in detailed understanding of your income and how much you are spending; at the end of the month, you will exactly know where you are spending more and where you need to cut down your expenses.

You will be able to allocate money to different categories, set limit on each category and understand what is your monthly spending pattern and how much do you exactly spend on each category.

The Following is a complete breakdown of the Project into major modules with different Applications.

Modules based on Mobile Application-

1. Mobile App UI
2. Mobile App Backend Implementation

Modules based on Image Processing and Machine Learning-

1. Training the Model based on set of invoices
2. Image to Text conversion
3. Data Set Creation
4. Categorization of Products based on Data Set

Other Features-

1.Suggestion and recommendation system using machine learning.

2.Uploading bills or receipts and extracting text and sorting into different categories.

3.Statistcs and Analytics

Not included in Module-

1.Reading SMS and UPI transactions.

2.Spliting expenses among family and friends.

1.3 Aim and Objective

The Aim and Objective of the project is to entirely manage and track the daily expenses of our Consumers; And to understand their needs and provide them a platform capable of monitoring all their Expenses through scanned receipts.

1.3.1 AIM

Our Aim is to create an application useful for our users by means of-

✓ Expense Monitoring

✓ Analyze Buying Trends

✓ Help in Financial Goals

✓ Better sense of Expenditure

1.3.2 OBJECTIVES

The following are the objectives that we want to achieve through Expense Manager: -

• Minimize manual effort with daily record of expenditures and incomes.

• Immediate and easy retrieval of report.

• Secured and transparent data.

• Graphical overview of transactions.

• Help in decision making with related results.

• Help in preparing wish list for pre planning your expenses.

1.4 Motivation

To understand whether individuals actually need an expense tracking application and to get their point of view on the issues that individuals are facing while managing their expenses and to understand whether our application to all intents and purposes going to be useful or not, we carried out a survey and the outcome that we came across was that-

- 67% of surveyed individuals are unaware of what is happening with their money, which is a crucial problem. If you are clueless about how much is your inflow and how much you are spending, you will not know at the end of the month what happened to your money.
- 80% are constantly stressed and worried about their finances
- 55% of individuals between the age group 21 to 27 become broke at the end of the month & 72% spend and save in haphazard manner.

Therefore, we thought of developing an android application which will help individuals overcome these problems and manage their finances wisely and effortlessly.

The first chapter tells us about the importance of the need of expense tracker. The next chapter focuses on the comparison of similar systems implemented and their limitations. The third chapter contains details about the proposed system, methodology, feasibility study and the designing of the system. The fourth chapter gives information about the hardware and software used, making of prototype and GUI, steps of image processing and the system code. Chapter five shows the comparison of results and discussion about the same. The next chapter highlights the concerns related to our project. In the final chapter we conclude our project along with the future scope of Expense Manager.

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY

We found various similar projects that have already been developed in the Technology Field. Unlike all those projects, Our Project provides security and graphical results as well as feature to classify all the products into categories. These projects are very helpful in the way to that they provide very efficient ways to extract data. These methods ensure more accurate results which are possible due to various Machine Learning and Deep Learning Algorithms. One of the widely algorithms for text recognition has been the LSTM (Long-Short Term Memory) Algorithm. This approach is based on improved extraction of information by ensemble of different features from invoices. In paper [1], the authors have used Bi-Directional LSTM algorithm for text extraction. Such a method improves the efficiency because it vectorizes each text box with semantic features enabling them to be distinct to each other. Whereas, in paper [3], the author has used CRNN Architecture alongside Single Directional LSTM. CRNN is able to take input images of varying dimensions and produces predictions with different lengths. One of the features of our project is to extract certain keywords from the scanned receipts like Store Name, Date, Total and the names of the Products along with their respective costs. We have encountered a similar approach in paper [2], where the authors have extracted keywords such as Store Name, Date and Total. They approached this issue by applying Deep Learning Algorithms such as CTPN and AED. The CTPN Algorithm is used for text detection, whereas AED algorithm is used for text recognition. Our Project builds on this approach by extracting more keywords such as the Product names and the costs. One of the major tasks in terms of Receipt Recognition is to decode Handwritten Receipts. Paper [4] has used similar techniques to approach this issue for recognizing hand written equations. This approach can be similarly applied along with the different Algorithms to achieve greater accuracy.

The previous projects have used a dataset of 1000 images of Scanned Invoices provided by the SROIE which is the same dataset which we have used to train our model.

Table 2.1: Comparison of Literature Surveyed

Title of paper and authors	Place and Year of Publication	Features	Methodology	Future Scope
1. Abstractive Information Extraction from Scanned Invoices (AIESI) using End-to-end sequential Approach, Shreeshiv Patel, Dvijesh Bhatt[5].	International Conference on Industrial and Information Systems, 2020	This paper presents the use of sequential approach for extracting information from Scanned Receipts.	Extraction of Text Using Bi-Directional LSTM Algorithm.	To create a model that can accept Handwritten Invoices along with the electronic receipts.
2. Deep Learning Approach for Receipt Recognition, Anh Duc Le, Dung Van Pham, Tuan Anh Nguyen[4].	Center of Open Data in Humanities Research Organization of Information Systems Tokyo, Japan & Deep Learning and Application, Vietnam, 2019	Pre-processing to extract receipt area and OCR verification to ignore handwriting. Achieved 71.9% of the F1 score for detection and recognition task.	CTPN for text detection and AED for text recognition.	The System can also be compatible with Languages other than English.
3. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition, Baoguang Shi, Xiang Bai, Cong Yao[7].	IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017	It is end-to-end trainable, it naturally handles sequences in arbitrary lengths, achieves remarkable performances in both lexicon-free and lexicon-based scene text recognition tasks.	CRNN Configuration with 2-Layer Single Directional LSTM.	CRNN is a general framework, it can be applied to other domains and problems (like Chinese character recognition).
4. Training an End-to-End System for Handwritten MathematicalExpression Recognition by Generated Patterns, Anh Duc Le, Masaki Nakagawa[3].	IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017	A convolution neural network for feature extraction from the image of an OHME, End-to-End system for recognizing OHMEs.	Local and global distortion models for patterns generation.	Improve the symbol recognition and expression recognition rate inside the end-to-end system by employing tree structured LSTM.

5. Family Expense Manager Application in Android, M N Rajaprabha[1].	IOP Conference Series: Materials Science and Engineering, 2017	Record bills or receipts, get status of bills paid or not.	Uses AI to learn expense patterns and generates report.	Expenses uploaded should be downloadable as a CSV/ Excel file.
6. Online Income and Expense Tracker, S. Chandini[12].	International Research Journal of Engineering and Technology (IRJET), 2019	Add bills with an option to attach the image of the bill or not, add the information about how the payment was made i.e., via check, card or cash, o view owes and lend expenses which adds or gets deducted from the overall budget.	Takes Income from user and divides in daily expense allowed. If user exceed that day's expense it will cut if from your income and give new daily expense allowed amount, and if that day's expense is less, it will add it in savings. Expense tracker will generate report at the end of month to show Income-Expense via multiple graphs.	The application can be extended to include scanning of barcode on the price tag which decreases the effort of entering the data in the input fields.
7. Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms, Joel Odd Emil Theologou[2].	Linköping University Department of Computer and Information Science, 2018	Receipt categories were correctly classified with 94% accuracy, sub categories which are much smaller subsets with less differences between categories got a very high accuracy with the evaluated model.	Collecting the physical receipts, transforming them to digital data, selecting and training machine learning algo on that data, writing a custom made algo to extract specific fields of data in and evaluating the classifiers performance.	Accuracy of retrieving price and extracting date can be improved.

CHAPTER 3

SYSTEM DESIGN

PROPOSED SYSTEM

As we know, a lot of people continuously struggle to keep up with their regular expenses and most of the times fail to control or manage their budget. This is often caused because of lack of structural maintenance of their expenses and budget which leads to over expenditure. To overcome this, we can create a platform for such people by providing an application which not only tracks all their expenses but also notify them in case of over expense.

The application will provide the following functionalities: -

1. Upload a receipt by clicking an image or choosing from Gallery and directly convert them into expenses by items.
2. The Application will categorize these items in specific categories (such as Groceries, Clothes, Stationery, etc) which will help the user to manage their budget wisely ensuring an efficient budget.
3. Users can set monthly limits on their expenses which will be tracked daily and help user spend their future expenses wisely.
4. Every month the user will be given suggestions according to their expenses helping them decide to reduce or stop further expenses as necessary.
5. User can also monitor his/her expenses and analyse categorically using graphs and pie charts provided by the application.

Such a system in place will largely enable consumers to practically understand and decide where to spend their money accordingly.

3.1 Program Architecture

The architecture as shown above has three blocks user interface, application server and cloud server. The user interface displays the application with which the user can interact while the cloud server consists of different datasets and AI model.

The application server further consists of the following modules android interface, receipt extraction, expense field, notification handler, association rules and database which is required.

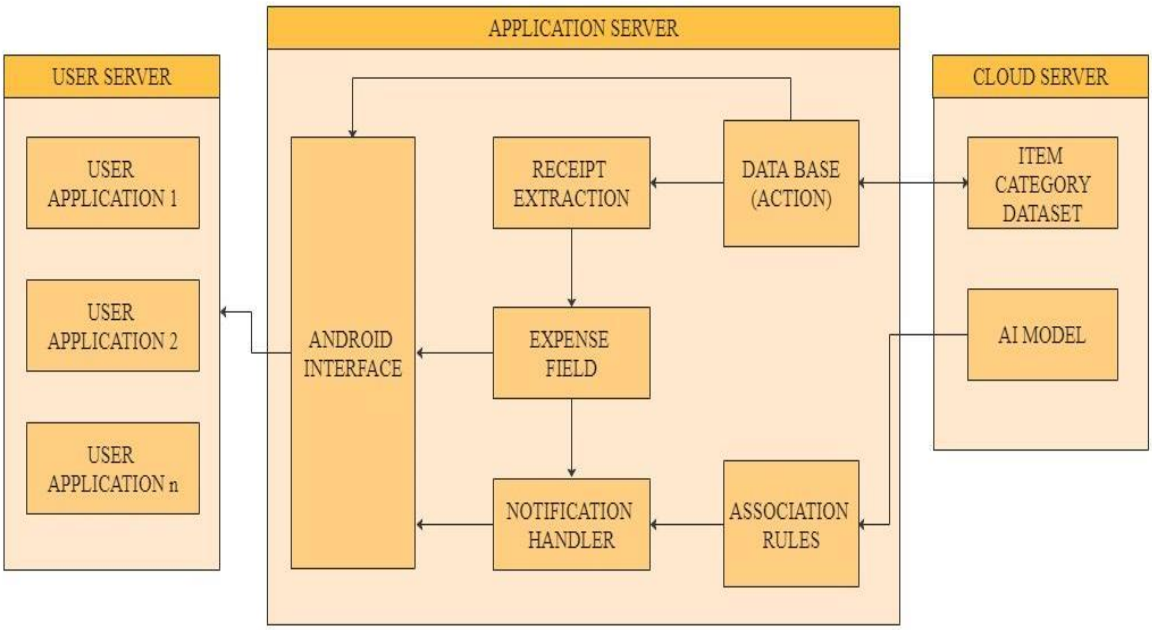

Fig.3.1: Program Architecture

The usual process of the architecture is as follows: -

- The user sends appropriate command or request to the server through the Internet, using the interface of the application.
- The application server is responsible for forwarding the command to the requested server.
- The server finds the results of requested commands (either the data processing or the database querying).
- The software or application delivers the processed information to the server.
- The server provides the user with the requested data.

Understanding each block in Detail: -

1. User Block

This block simply consists of the user's device with the proposed application and internet connection. The user interface convenience is one of the main factors because that is actually what customers face, operate and interact with.

2. Application Server

The main task of the server is managing the commands of the user using the app. Thus, the server

side performs the set commands whereas the database store the data. The database and association rules fetch data from the cloud server. Further the database block is connected to the receipt extraction block while association rules block is connected to the notification handler block. Receipt extraction gives data to expense field while it further sends data to the notification handler. All these blocks including the database is connected to the android interface.

3. Cloud Server

A cloud server is powerful physical or virtual infrastructure that performs application- and information-processing storage. The cloud server consists of item category datasets and AI model. The datasets will consist of different shopping categories like clothes, grocery, electronics etc. while the model will be used to segregate the items into different categories.

3.2 Methodology

Following are the steps involved-

1. Image Processing for receipt recognition.
2. Sorting Items in the receipt into different categories
3. Machine Learning
4. MPAndroidChart

1. Image Processing for receipt recognition.

Image processing is a technique to perform some various operations on an image, in order to get an amplified or improved image or to extract meaningful information from it. In this project we are using Image Processing tools on images such as receipts & bills and perform various Image Processing algorithms for better efficiency and output.

Image processing basically includes the following three steps:

- Importing the image via image processing tools.
- Analyzing, manipulating the image.
- Proving an output which can be an altered image or conclusion based on image analysis.

So, for this project will process all our receipts to provide us with necessary output such as –

- Store Name
- Date of Transaction
- Item Name
- Cost of Items
- Total Amount

Fig.3.2: Scanned Receipt

We will use Python to process all our receipts and will use Image Processing techniques and Python packages such as-

- OpenCV
- PyTorch
- Tesseract

Fig.3.3: Invoice

After extracting the necessary information -

1. We will use keywords such as Product and its Price from the receipts and categorize them into various categories. These categories will be general classes such as (Groceries, Clothing, Stationery, etc.)
2. These Products will be matched against a big database of such categories which will help determine which product belongs to which category.
3. Once all the Products are sorted, the User will be able to view the expenses according to categories helping them analyze their expenses better.
4. If there is any Product which doesn't find any category, with the help of AI we will later sort it into one of the categories or classify it as Miscellaneous.

2. Sorting items in the receipt into different categories

Using Machine Learning algorithms in python like tesseract and pytorch the application will use keywords such as Product and its Price from the receipts and categorize them into various categories. These categories will be general classes such as Groceries, Clothing, Stationery, etc. These Products will be matched against a big database of such categories which will help determine which product belongs to which category.

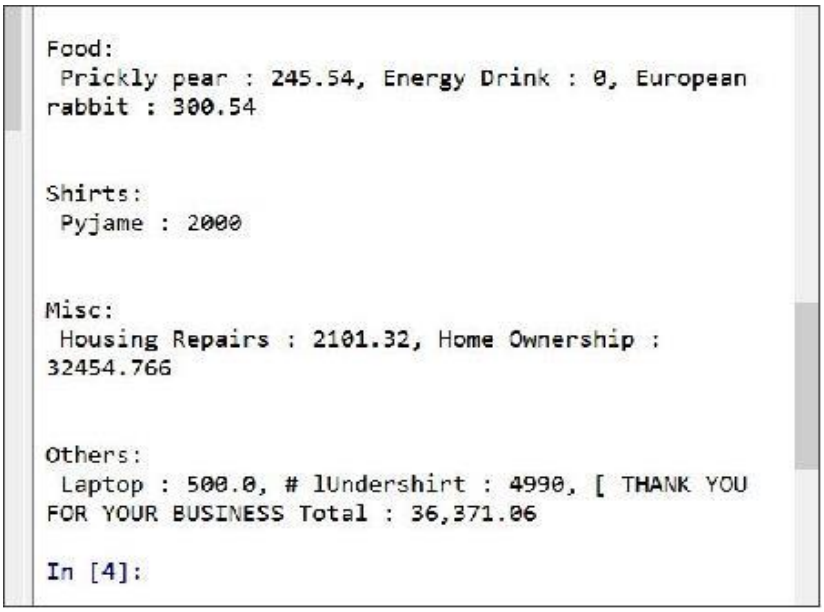

Fig.3.4: Items sorted into different categories

Once all the Products are sorted, the User will be able to view the expenses according to categories helping them analyze their expenses better. If there is any Product which doesn't find any category, with the help of AI we will later sort it into one of the categories or classify it as Miscellaneous.

3. Machine Learning

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

In our project, we will be creating Machine Learning Models to –

1. Train the model to recognize receipt formats using a data-set of almost 1000 receipts of different formats to increase the efficiency of the system against new or different types of receipts.
2. Use various Machine Learning algorithms such as CTPN, LSTM and AED to get better and accurate results on applying receipt recognition.

4. MPAndroidChart

In this project, MPAndroidChart library is used to create beautiful, customized and animated charts. Pie charts, line graphs, bar graphs are used in this project to give the users visual representation of their expenses. The users can view these graphical reports either by transaction date or by category. Adding graphs and charts will make information clearer and therefore improve the user experience. This will give the user a better understanding of their expenditure.

3.3 Analysis

3.3.1 Process Model - (Interactive Model)

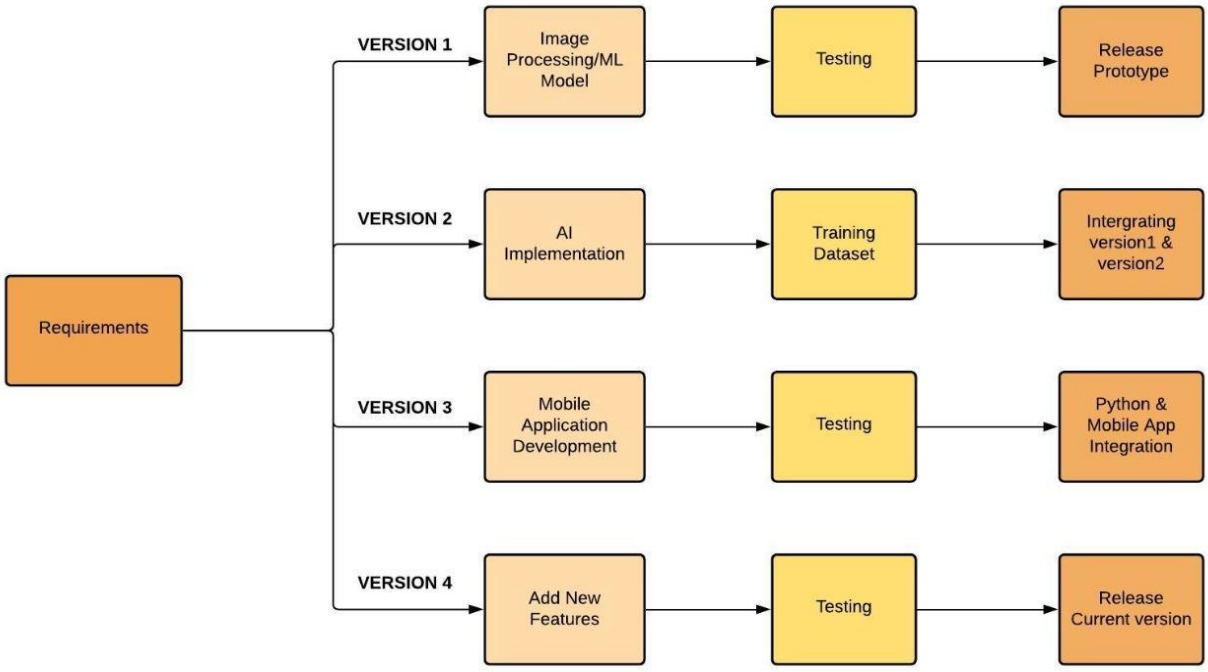

Fig 3.5: Iterative Model

The iterative development model develops a system through building small portions of all the features. This helps to meet initial scope quickly and release it for feedback.

In this Project, we will build various versions of the Application by constantly taking feedback by our guide and panel members. Based on the requirements we will keep adding these modules into our project and at the end of Build 4, we will have a product capable of an Alpha release and we can further go on to release the beta product.

In the first Phase of the Project, we will create various models using Python for image processing and Machine Learning to achieve the necessary output and then release the project for further review.

In this phase we will code use IP and ML techniques for-

1. Image to Text conversion for receipts
2. Classification of Items into different categories
3. Training the Model using a set of 1000 receipts

In the second Phase of the Project, we are going to create some Artificial Intelligence models to continuously improve and evolve our Item Classification process by making the program learn new products and identify their categories on its own, thereby improving our Database.

In the third Phase of the Project, we are going to develop a Mobile Application using Android Studio with all the necessary UI features. We will then integrate the Program Logic coded in Python with this Mobile Application. The completion of this stage will give us our first Alpha version of the Product.

The Fourth Phase of the Project will be receiving feedback from our users and then implementing new features to the Application accordingly.

3.3.2 Feasibility Analysis

1. **Technical feasibility**: Technical feasibility focuses on the technical resources (software and hardware) available to the organization and helps to determine whether the technical team is capable of converting the ideas into working systems. We don't require any hardware component for this project and the software required Android Studio for the development of the application is already with us and is a Free-to-use Software. The project involves solvable technical issues.

2. **Economic feasibility**: This assessment typically involves a cost/ benefits analysis of the project. Economic feasibility also determines whether the required software is capable of generating financial gains. This project developed in minimal amount will emerge as a huge market valued project, beneficial for almost all the users.

3. **Legal feasibility**: This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts, or social media laws, project certificate, license, copyright etc. The proposed project conforms all legal and ethical requirements.

4. **Schedule Feasibility: -** In Schedule Feasibility Study mainly timelines/deadlines are analyzed for proposed project which includes how many times teams will take to complete final project. Timeline of the project is mentioned below.

3.3.3 Timeline Chart

Task	Estimate Start Date	Duration
Topic Selection	15-02-2021	10
Topic Approval	26-02-2021	1
Project Planning	27-02-2021	5
Prototype for App	20-03-2021	2
Image to Text Conversion	22-03-2021	6
Building the App	16-06-2021	15
Coding for Category Sorting	15-07-2021	10
AI implementation	01-08-2021	30
Python and App Integration	15-09-2021	3
Additional App Features	01-10-2021	15
Project Testing	01-11-2021	10
Deployment	01-01-2022	10
Publication & Documentation	01-03-2022	20

Fig 3.6: Timeline chart

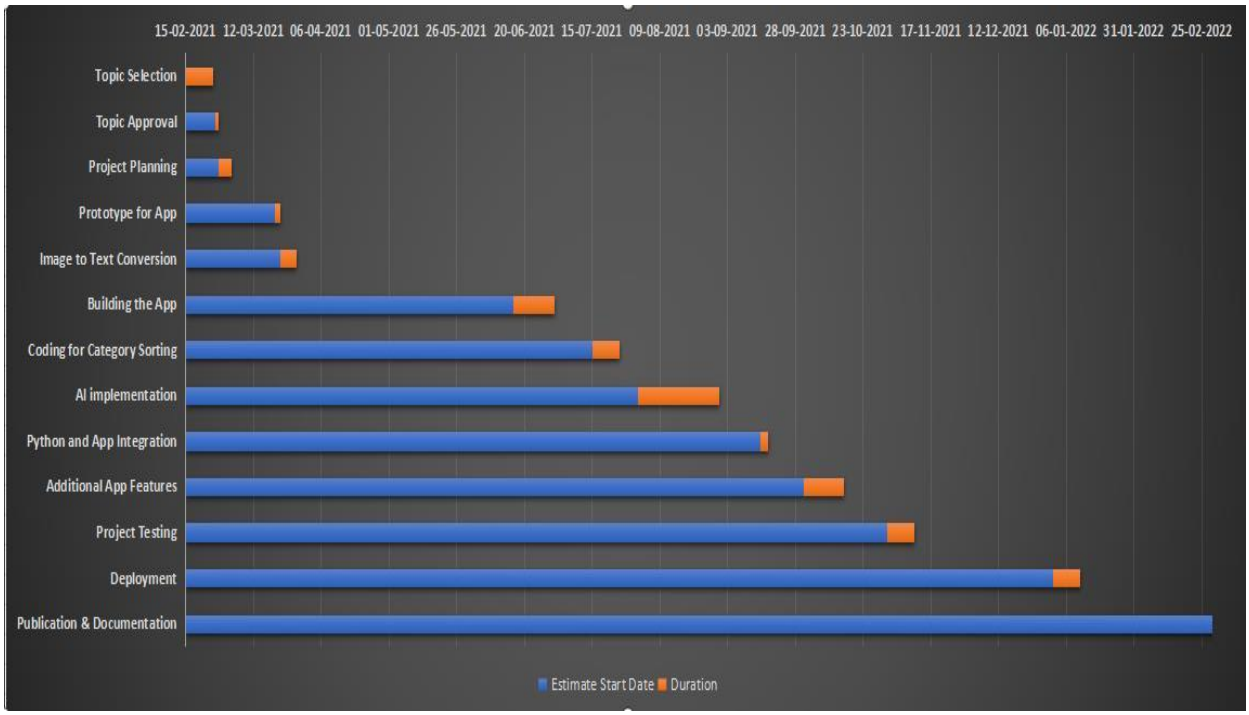

Fig 3.7: Gantt Chart

3.3.4 Design and UML diagram

- SEQUENCE DIAGRAM

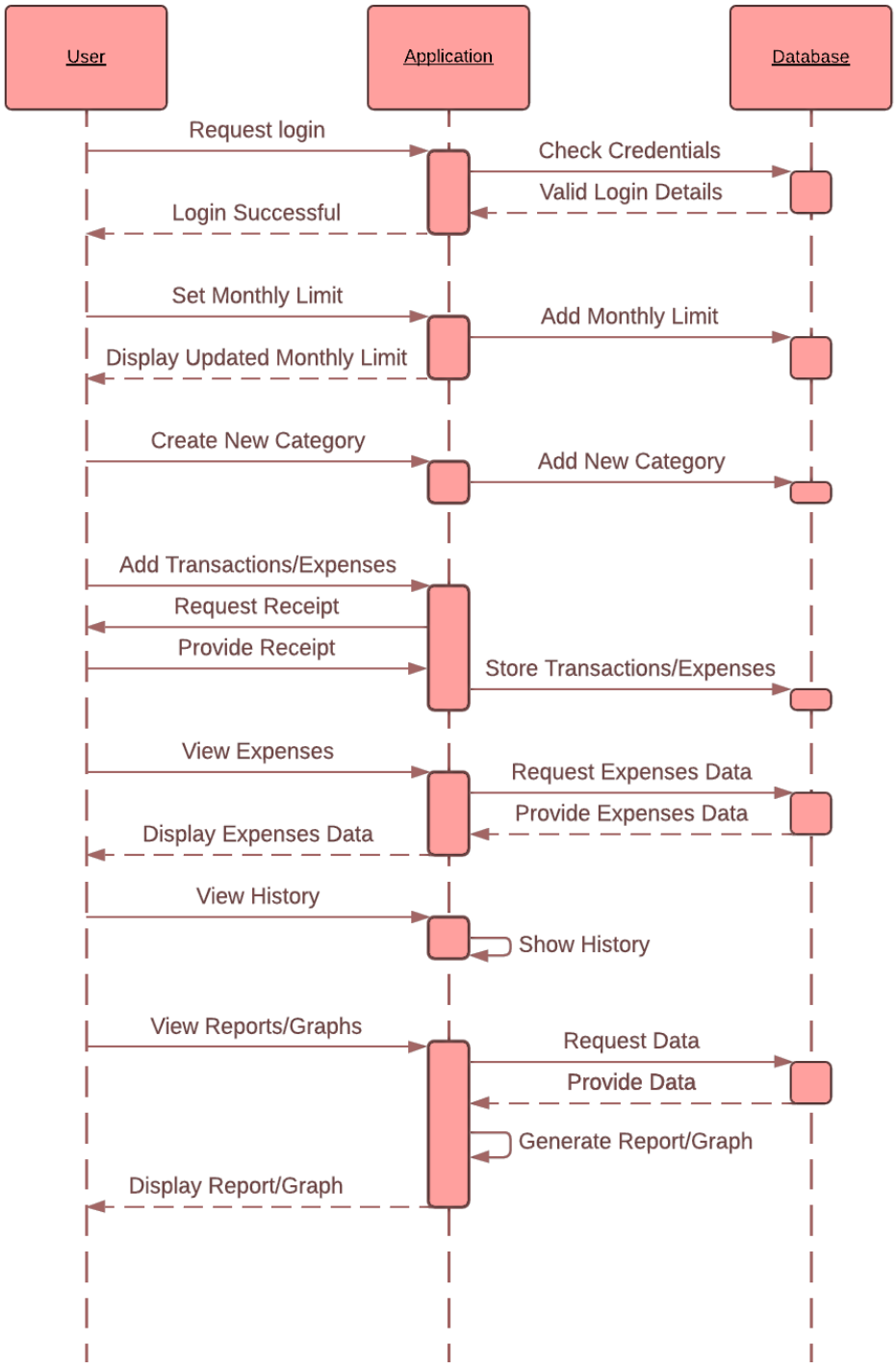

Fig 3.8: Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

- ER DIAGRAM

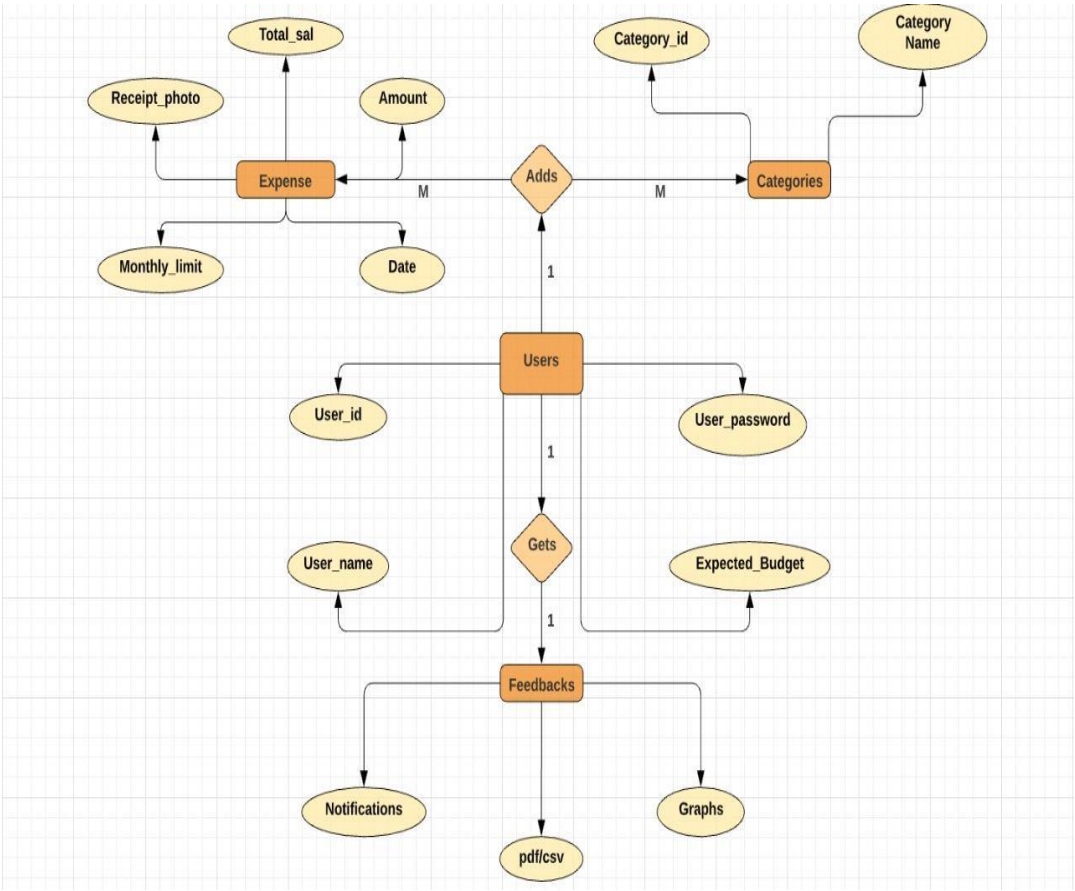

Fig 3.9: ER Diagram

ER Diagram stands for 'Entity Relationship' Diagram. An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

CHAPTER 4

SYSTEM IMPLEMENTATION

SYSTEM IMPLEMENTATION

4.1 Text Extraction

In this project we have used Python-Tesseract OCR for extracting data from the receipts, tesseract is an open-source project that is supported by Google. Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and “read” the text embedded in images. Python-tesseract is a wrapper for Google’s Tesseract-OCR Engine. And OCR stands for ‘Optical Character Recognition’ which is a technology that automatically recognizes characters in images and processes this into a text which enables us to both classify and extract data from them with machine learning models.

The different python libraries used for this process are PIL, pytesseract, cv2, imutils, csv and urllib.request-

Python Imaging Library (PIL): - This library was used for opening and saving the scanned images of the receipt. It incorporates lightweight image processing tools that aids in editing, creating and saving images.

Pytesseract: - Optical Character Recognition tool which can read all image types supported by Pillow/PIL.

Cv2: - Computer vision library in python which was used for loading the images into the python script.

Imutils: - This is a image processing library in python which was used for resizing, rotating the receipt images.

Csv: - This was used for written the extracted data in the format preferred by Excel.

Urlib.request: - As the scanned receipts were stored in the form ‘purl’ in the firebase database this library was used for opening this url. In its simplest form you create a Request object that specifies the URL you want to fetch. Calling urlopen with this Request object returns a response object for the URL requested.

Code: -

```
from PIL import Image
import pytesseract as tess
import argparse
import cv2
import os
import imutils
import re
import csv
import urllib.request

tess.pytesseract.tesseract_cmd=r'C:\Users\Pranay Yenagandula\AppData\Local\Tesseract-OCR\tesseract.exe'

imgs=[]
bills=db.child('Registered_Users').child(users[1]).child('Bills').get()
for b in bills.each():
    tempb=db.child('Registered_Users').child(users[1]).child('Bills').child(b.key()).get()
    print(tempb.val()["imageUrl"],tempb.val()["date"])
    imgs.append(tempb.val()["imageUrl"])
'''
i=db.child("Registered_Users").child(users[-1]).get()
img_url= i.val()["purl"]
'''
print(imgs[-1])
urllib.request.urlretrieve(imgs[-1],"image.jpeg")
img = Image.open("image.jpeg")

text = tess.image_to_string(img)

# Grayscale, Gaussian blur, Otsu's threshold
image = cv2.imread('1.jpeg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (3,3), 0)
thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]

import pandas as pd
dataset = pd.DataFrame(columns=['Product_Name','Price'])

from nltk.tokenize import word_tokenize

flag=False
inputlistv=[]
```

```
inputlistk=[]
keys=['ITEM','QTY','QUANTITY','DESCRIPTION','DES','AMOUNT','AMT','PRICE','ITEM
DESCRIPTION','ITEMS']
for i, line in enumerate(text.splitlines()):
  #print(line.upper())
  if any(l in line.upper() for l in keys) or flag:
     flag=True
  else:
     continue
  word_list = word_tokenize(line)

  if len(word_list) > 1 and any(i.isdigit() for i in word_list[-1]):
     temp=[]
     item_words = word_list[:-1]
     if item_words[-1] == 'R':
        item_words = item_words[:-1]
     for i in item_words:
        if i.isalpha() and not i.isspace():
           temp.append(i)
     item=' '.join(temp)
     #print(item)
     price = word_list[-1]

     #dataset = dataset.append(pd.DataFrame([[item, price]], columns =dataset.columns))
     if 'TOTAL' in line.upper():
        totalprice=price
        break
     inputlistk.append(item)
     inputlistv.append(price)

print(inputlistk)
print(inputlistv)
```

Output: -

1950

Gerbang Alaf Restaurants Sdn Bhd
(65351-M)
formerly known as
Golden Arches Restaurants Sdn Bhd
Licensee of McDonald's
Level 6, Bangunan TH, Damansara Uptown3
No.3, Jalan SS21/39,47400 Petaling Jaya
Selangor
(GST ID No: 000504664064)
McDonald's BHP Taman Melawati(#328)
Tel No. 03-4162-2318
TAX INVOICE

MFY Side 1

INV# 003281900229170
ORD #50 -REG #19- 18/01/2018 10:27:03

QTY	ITEM	TOTAL
2	ChicMcMuffin	11.00
1	M Porridge	5.60
1	BM 2HotCakes	10.00
1	L Coke	

TakeOut Total (incl GST)	26.60
Total Rounded	26.60
Cash Tendered	100.00
Change	73.40
TOTAL INCLUDES 6% GST	1.51

We'd love to hear your feedback!
Visit www.mcdfeedback.com.my
OR
Download My McD Feedback from
Google Play/App Store
to share your experience

Thank You and Please Come Again.
Customer Service Hotline : 03-2726-5600

Fig.4.1: Scanned Receipt

```
********************

Cost:
 ['11.00', '5.60', '10.00']

Items:
 ['2 ChicMcMuf fin', '1M Porridge', '1 BM
2HotCakes']

Total Price=  26.60

********************
```

Fig.4.2: Extracted Text

4.2 Improving Image Quality

Image acquisition

Image processing filters applied to the scanned images of the receipt:

1. Resizing of the selected image.
2. Convert the given image into grayscale.
3. Applying median filter for noise removal.
4. Applying gaussian filter to smoothen the image and to further reduce the noise
5. Threshold is applied to convert the image into black and white form so that the required data labels can be easily detected and extracted.

Code: -

```
# Grayscale, Gaussian blur, Otsu's threshold
image = cv2.imread('1.jpeg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (3,3), 0)
thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]
```

4.3 Sorting the extracted text

The next step after extracting the text was to sort the items in their respective category. For this purpose, we created a dataset which consists of different categories like ‘Grocery’, ‘Clothing’, ‘Footwear’ etc. as shown below.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Clothing	FootWear	Furniture	Stationary	Electronics	Toys	Beauty And	Sports And F	Accessories	Baby Care	Home and K	Groceries	Medical	Vegetables	Food
2	T-Shirts	Heels	Cabinets	Pen	Air purifier	Teddy Bear	Jewellery Va	Yoga Mats	Finger Ring	Elite Collect	Cleaner	Rice - Meals	Aspirin	Kashini gree	Pizza
3	Undershirt	Wedges	Bed	Paper	Air conditio	Remote Cor	Soap	Ball	Rubber ban	Cartoon Sti	Repairing	Brown Basn	Anti-inflamr	Pineapple ju	Burger
4	Henley Shirt	Shoes	Chair	Sharpner	Alarm clock	Barbie	Shower Cap	Football	Belt	Stickers	Disinfectant	Cracked Wh	Vitamins	Thiland Juic	Spaghettie
5	Slips	Sneakers	Sofa	Eraser	Backup cha	Barbie Hous	Perfume	Bat	Wallet	Wallmantra	Rugs	Flattened Ri	Antiseptic	Thiland Jelly	Pasta
6	Pyjama	Flats	Bench	Highlighter F	Bread make	Playing Carc	Hand lotion	Helmet	Purse	Cradle	Sofa Covers	Puffed Rice	Tablets	Orange juice	French Fries
7	Shirts	Slippers	Couch	Pencil Case	Banknote cc	Uno	Razor	Pads	Handbag	Diaper	Blanket	Rava (Crean	Burnheal	Chikkadikai	Ice-Cream
8	Vestes	Flip-Flops	Rocker	Stapler	Blender	Robot	Tissues	Badminton	Clutch	Nappies	Cushion	Whole-Whe	Volin	Grapesh Tas	Fried Rice
9	Hoodies	Sandal	Stool	Whitener	Bluetooth s	Radio Flyer	Hand mirror	Golf Ball	Watch	Cradle	Cushion Cov	Atta Flour	Move Crear	Times Rose	Pancakes
10	Suits	Loafer	Throne	Notepads	Bulb	Chemistry S	Wet wipe	Skates	Smartwatch	Bassinet	Curtain	Millet Flour	Bandage	Washington	Noodles
11	Tanks	Boot	Chests	Push-pin	Calculator	Joy Buzzer	Toothbrush	Skating Kit	Aerobars	Baby Wipe	Cleaner	Besan or Ch	Elastic Tape	Chilles smal	Pumpkin Pie
12	Jumpsuits	Ballet	Coffer	Drawing pin	Car toy	Yo-Yo	Sunscreen	Batting Glov	Jewelry	Feeding Bot	Mop	Tuvar Dalor	Oxygen Mask		Bagel
13	Overcoats	Pumps	Dresser	Paper clip	Ceiling fan	Pop-Up Boc	Deodorant	Flying discs	Hats	Baby Rocke	Mop Fibre	Chana Dal	Oximeter		Muffins
14	Cardigans	Mule	Wardrobe	Rubber stan	Chandelier	Stuffed Mic	Toothpaste	Nets	Bag	Baby Plaype	Curtain Rod	Urad Dal	Syringe		Cake

Fig.4.2: Dataset

Empty lists with the name of these categories were created in the python program. To sort the items python program traverses this dataset and checks for the item extracted from the receipt, if it encounters that item in the dataset it will append the item name to that particular list and if the program doesn’t find the item in the dataset it will append it to a miscellaneous category.

Code: -

```
foodf=[]
shirts=[]
externale=[]
othero=[]
food=[]
external=[]
shirt=[]
temp=""

with open('2016-17-3.csv') as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=',')
    line_count = 0
    for row in csv_reader:
        if line_count == 0:

            line_count += 1
        else:
            {external.append(row[0])}
            line_count += 1

    print(f'Processed {line_count} Lines.')
with open('men.csv') as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=',')
    line_count = 0
    for row in csv_reader:
        if line_count == 0:

            line_count += 1
        else:
            {shirt.append(row[1])}
            line_count += 1

    print(f'Processed {line_count} Lines.')
with open('generic-food.csv') as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=',')
    line_count = 0
    for row in csv_reader:
        if line_count == 0:

            line_count += 1
```

```
        else:
            {food.append(row[0])}
            line_count += 1

    print(f'Processed {line_count} Lines.')

food_dict={}
shirt_dict={}
other_dict={}
external_dict={}
for i in range(len(inputlistk)):
    if(inputlistk[i] in food):
        temp=inputlistk[i],":",inputlistv[i]
        foodf.append(" ".join(temp))
        food_dict[inputlistk[i]]=inputlistv[i]
    elif(inputlistk[i] in shirt):
        temp = inputlistk[i], ":", inputlistv[i]
        shirts.append(" ".join(temp))
        shirt_dict[inputlistk[i]]=inputlistv[i]
    elif(inputlistk[i] in external):
        temp = inputlistk[i], ":", inputlistv[i]
        externale.append(" ".join(temp))
        external_dict[inputlistk[i]]=inputlistv[i]
    else:
        temp = inputlistk[i], ":", inputlistv[i]
        othero.append(" ".join(temp))
        other_dict[inputlistk[i]]=inputlistv[i]

print("\n")
print("Food:\n",", ".join(foodf))
print("\n")
print("Clothing:\n",", ".join(shirts))
print("\n")
print("Misc:\n",", ".join(externale))
print("\n")
print("Others:\n",", ".join(othero))
```

Output: -

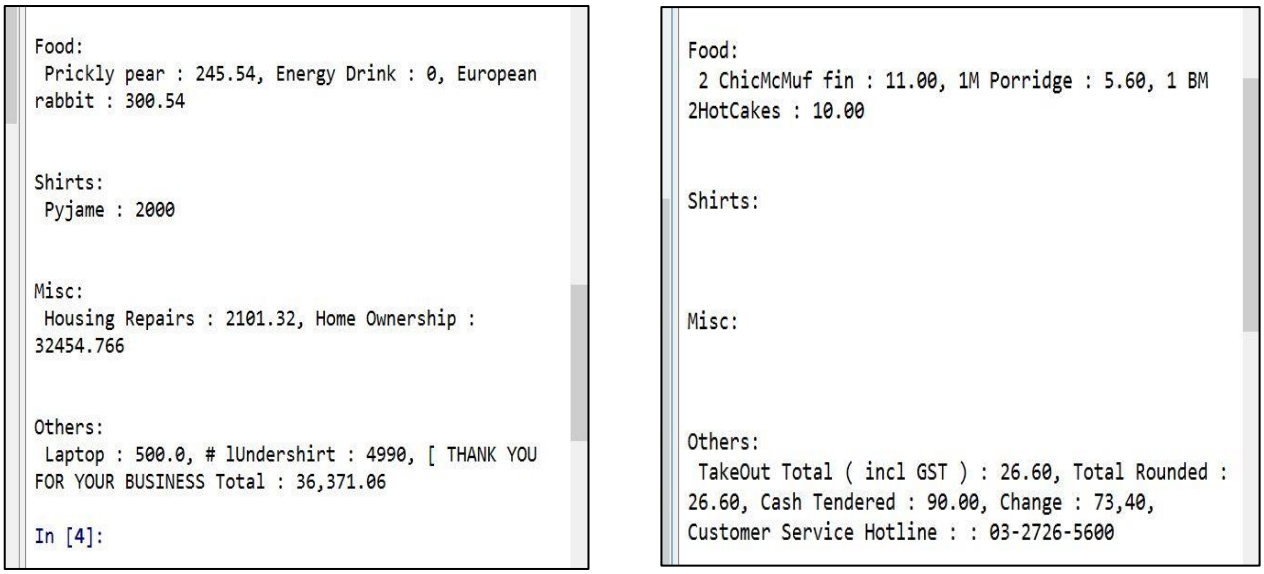

Fig.4.4: Sorted Items

4.4 Database Connection

We used 'Firebase' for the purpose of database in this project. Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is a real time database; data is synced across all clients in real time and remains available even when an app goes offline.

Code: -

```
import pyrebase
config = {

  "apiKey": "AIzaSyC_ra0XUsWlTJISdfzPohZOae5flO59GTY",

  "authDomain": "verificationotp-e6b9b.firebaseapp.com",

        "databaseURL": "https://verificationotp-e6b9b-default-rtdb.firebaseio.com",

  "storageBucket": "verificationotp-e6b9b.appspot.com"
```

```
}


firebase = pyrebase.initialize_app(config)
db=firebase.database()
storage=firebase.storage()
users=[]
all_users = db.child("Registered_Users").child().get()
for user in all_users.each():
    print(user.key())
    users.append(user.key())
```

Output: -

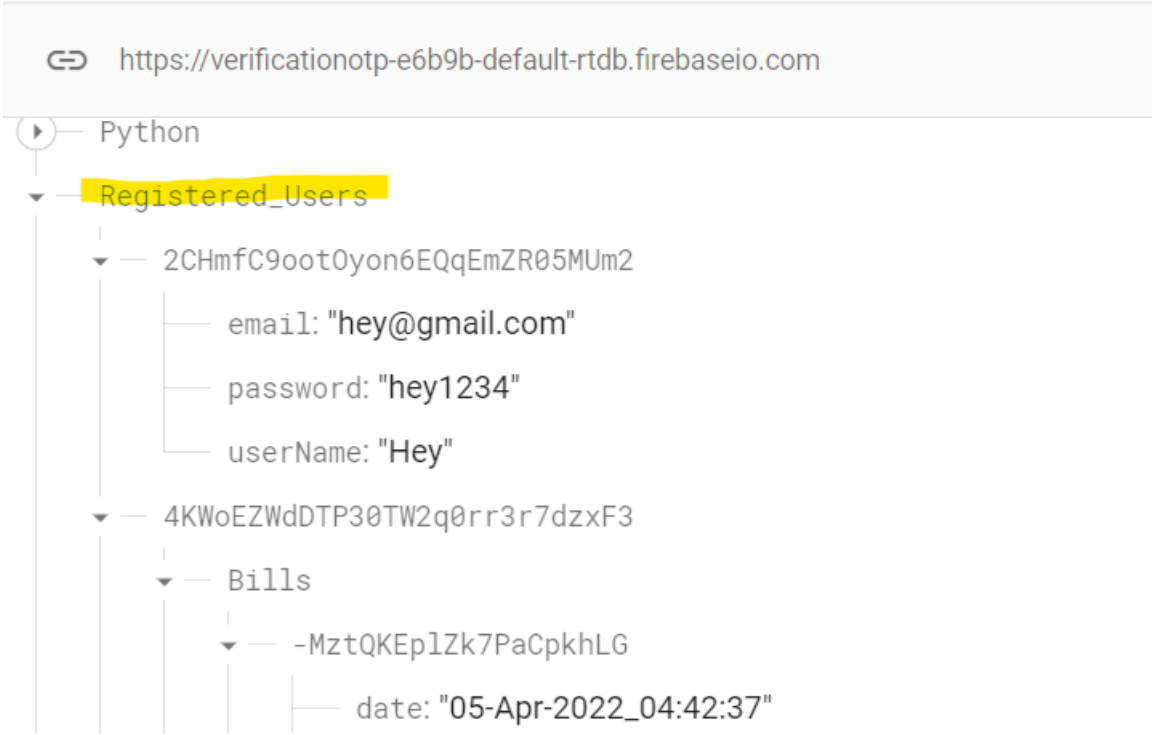

Fig.4.5: Database

The scanned bills were stored in the form of purl in the database and the extracted data was also stored in the database according to its category.

Code: -

```
imgs=[]
bills=db.child('Registered_Users').child(users[1]).child('Bills').get()
for b in bills.each():
    tempb=db.child('Registered_Users').child(users[1]).child('Bills').child(
b.key()).get()
```

```
        print(tempb.val()["imageUrl"],tempb.val()["date"])
        imgs.append(tempb.val()["imageUrl"])

    '''
    i=db.child("Registered_Users").child(users[-1]).get()
    img_url= i.val()["purl"]
    '''
    print(imgs[-1])
    urllib.request.urlretrieve(imgs[-1],"image.jpeg")
    img = Image.open("image.jpeg")
```

Output: -

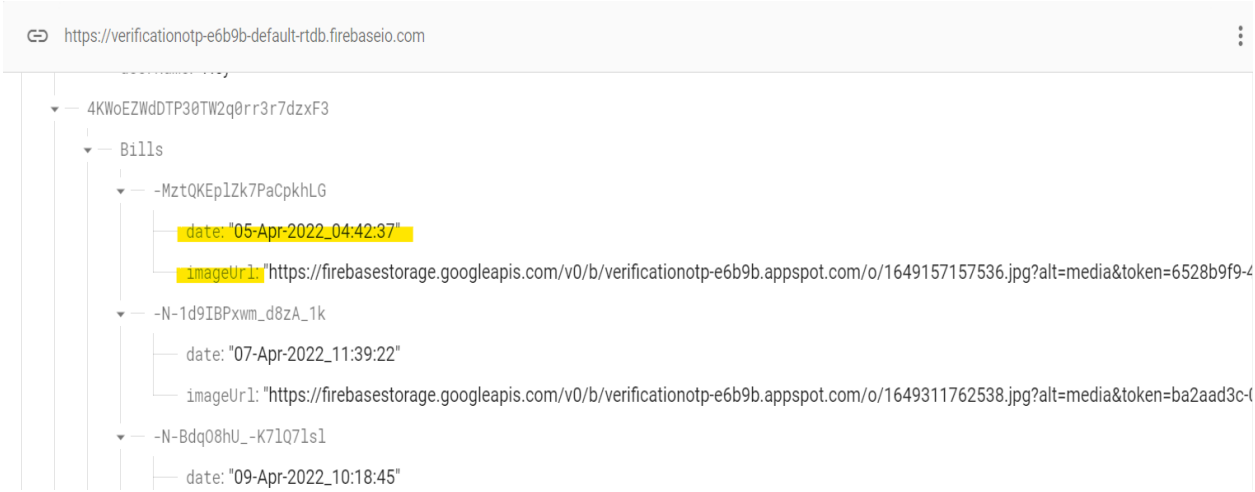

Fig.4.6: Image url

Code: -

```
print(type(othero),othero)
print(other_dict)
data={"name":"Pranay","age":20,"comment":"Pyrebase testing"}

db.child("Extracted Text").child(users[1]).child("Food & Dining").update(food_dict)
db.child("Extracted Text").child(users[1]).child("Clothing").update(shirt_dict)
db.child("Extracted Text").child(users[1]).child("External").update(external_dict)
db.child("Extracted Text").child(users[1]).child("Others").update(other_dict)

def stream_handler(message):
    print(message["event"]) # put
    print(message["path"]) # /-K7yGTTEp7O549EzTYtI
```

```
    print(message["data"]) # {'title': 'Pyrebase', "body": "etc..."}

my_stream = db.child("posts").stream(stream_handler)

explist=[]
exp=db.child('Expense List').child().get()
ekey= exp.key()
print(ekey,"okay")
itemkeys=db.child('Expense List').child(ekey).child(exp).get()
for e in itemkeys.each():
    explist.append(e.key())
print(explist[1])

medical,foodexp,stationery=0,0,0
for i in explist:
    t=db.child('Expense List').child(ekey).child(exp).child(i).get()
    if t.val()['item']=='Medical':
        medical+=t.val()['amount']
    if t.val()['item']=='Food':
        foodexp+=t.val()['amount']
    if t.val()['item']=='Stationery':
        stationery+=t.val()['amount']

print(medical)

db.child('Extracted Text').child(user[1]).update({'Medical':medical})
db.child('Extracted Text').child(user[1]).update({'Food & Dining':foodexp})
db.child('Extracted Text').child(user[1]).update({'Entertainment':stationery})
db.child('Extracted Text').child(user[1]).update({'Others':stationery})
```

Output: -

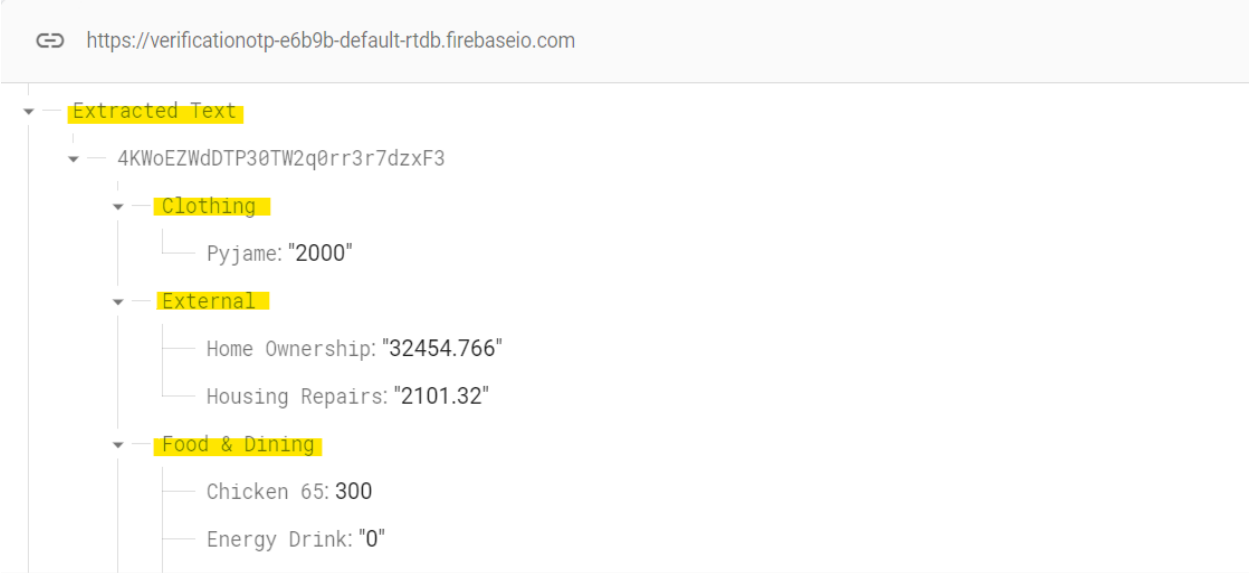

Fig.4.7: Extracted Text stored in Database

Manually added items from the application were also stored in the database along with its date and respective category.

Code: -

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_today_analytics);

    mAuth = FirebaseAuth.getInstance();
    onlineUserId = mAuth.getCurrentUser().getUid();
    expensesRef = FirebaseDatabase.getInstance().getReference("Expense
List").child(onlineUserId);
    personalRef =
FirebaseDatabase.getInstance().getReference("personal").child(onlineUserId);

private void getTotalWeekFoodExpense(){
    MutableDateTime epoch = new MutableDateTime();
    epoch.setDate(0); //Set to Epoch time
    DateTime now = new DateTime();
    DateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy");
    Calendar cal = Calendar.getInstance();
    String date = dateFormat.format(cal.getTime());
    String itemNday = "Food And Dining"+date;

    DatabaseReference reference =
FirebaseDatabase.getInstance().getReference("Expense
List").child(onlineUserId);
    Query query = reference.orderByChild("itemNday").equalTo(itemNday);
    query.addValueEventListener(new ValueEventListener() {
```

```
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if (snapshot.exists()){
                int totalAmount = 0;
                for (DataSnapshot ds :  snapshot.getChildren()) {
                    Map<String, Object> map = (Map<String, Object>)
ds.getValue();
                    Object total = map.get("amount");
                    int pTotal = Integer.parseInt(String.valueOf(total));
                    totalAmount += pTotal;
                    analyticsFoodAmount.setText("Spent: " + totalAmount);
                }
                personalRef.child("dayFood").setValue(totalAmount);
            }else {
                linearLayoutFood.setVisibility(View.GONE);
            }

        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            Toast.makeText(TodayAnalytics.this, error.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    });
}
```

Output: -

Fig.4.8: Manually added data

The database consists of 5 tables which are –

- Expense List
- Extracted Text
- Other
- Python
- Registered_Users
- Personal

Fig.4.9: Tables in Database

4.5 OTP Verification

One of the important features of any application is the security it provides to the user i.e., how secure is the data of the user. As this is expense tracking application users are going to input their expense and budget related information in the application and to keep this data safe and secure if are providing a secure login option to the user i.e., one time password. When the user logs in into the application, he/she will be provided with an OTP on the registered mobile number which will be use to authenticate the user.

Code: -

```
@Override
public void onClick(View view) {
    PhoneAuthProvider.getInstance().verifyPhoneNumber("+91" +
getIntent().getStringExtra("mobile"),
            60, TimeUnit.SECONDS,
            LoginActivity.this,
            new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {
                @Override
                public void onVerificationCompleted(@NonNull
PhoneAuthCredential phoneAuthCredential) {



                }

                @Override
                public void onVerificationFailed(@NonNull FirebaseException
e) {


                    Toast.makeText(LoginActivity.this,e.getMessage(),
Toast.LENGTH_SHORT).show();


                }

                @Override
                public void onCodeSent(@NonNull String current_backendotp,
@NonNull PhoneAuthProvider.ForceResendingToken forceResendingToken) {
                    getOtpBackend=current_backendotp;
                    Toast.makeText(LoginActivity.this, "OTP Sended
Successfully", Toast.LENGTH_SHORT).show();




                }
            });

}
```

Output: -

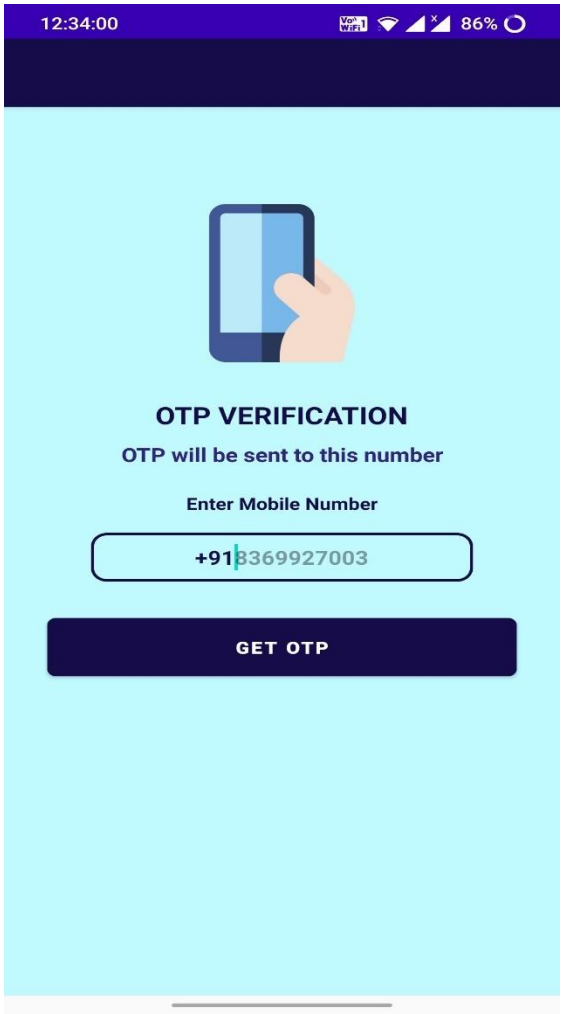

Fig.4.10: OTP Verification

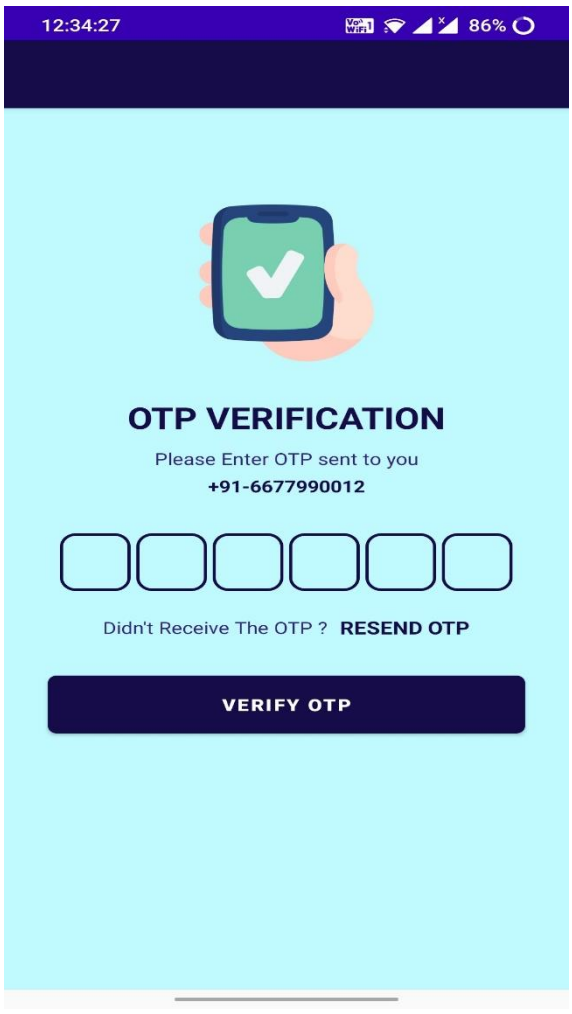

Fig.4.11: OTP Screen

4.6 Making of GUI

The entire GUI of the application is divided into 3 fragments namely the 'Home Screen', the 'Dashboard' and the 'Settings' screen. Apart from this there is a registration screen which is be displayed to first time users and this data is stored in the database which is used to authenticate the user during the login process on the login screen. In the next screen monthly income as well as monthly limit of the user is taken as an input as shown below.

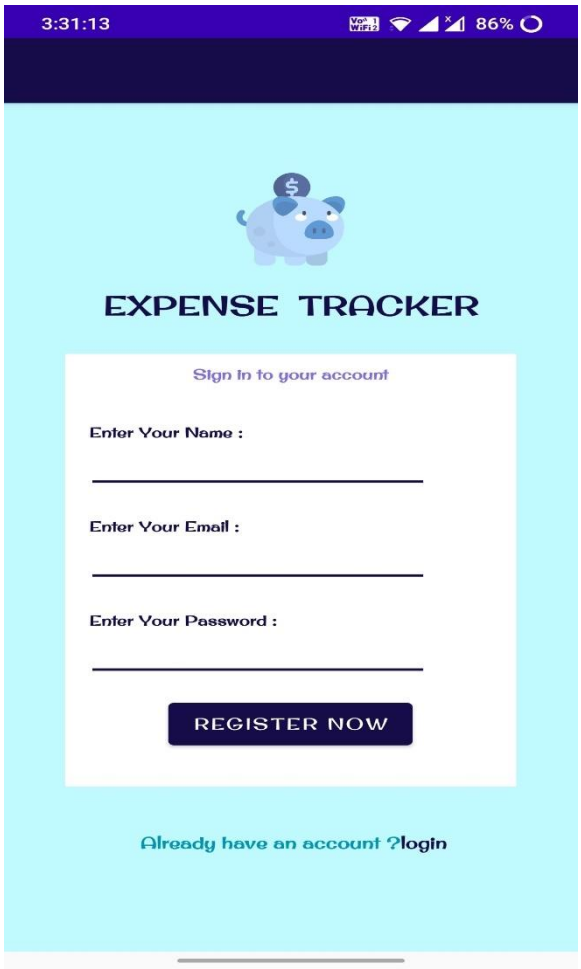

Fig.4.12: Registration Screen

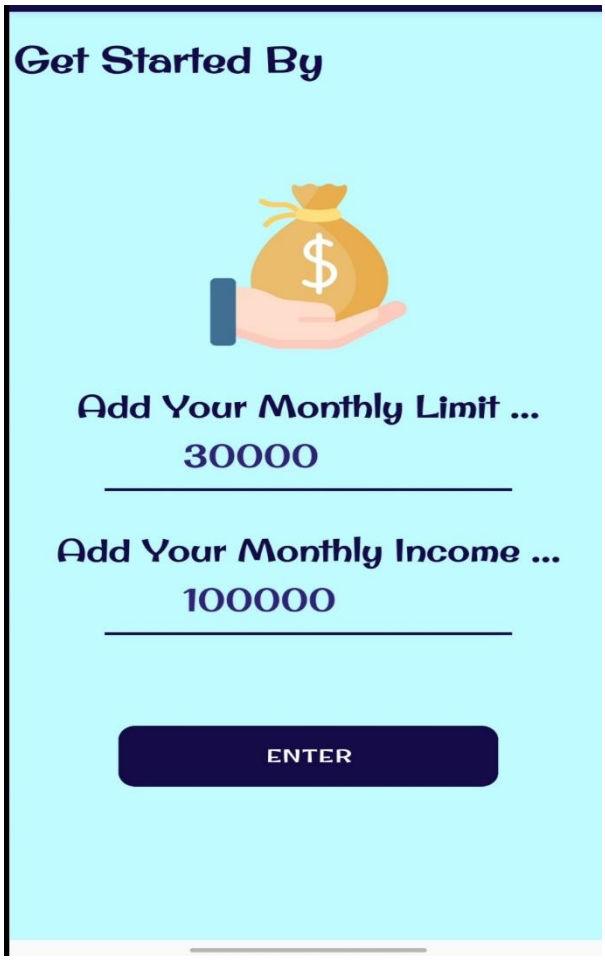

Fig.4.13: Monthly Limit Screen

Home Screen

The home screen displays the monthly expenses of the user in graphical form. It also has two buttons namely ‘This Week’ and ‘This Month’ clicking which the user can view his/her weekly and monthly expenses as shown below. This week screen will show the expenses for that particular week and this month screen will show the expenses for the entire month according the categories.

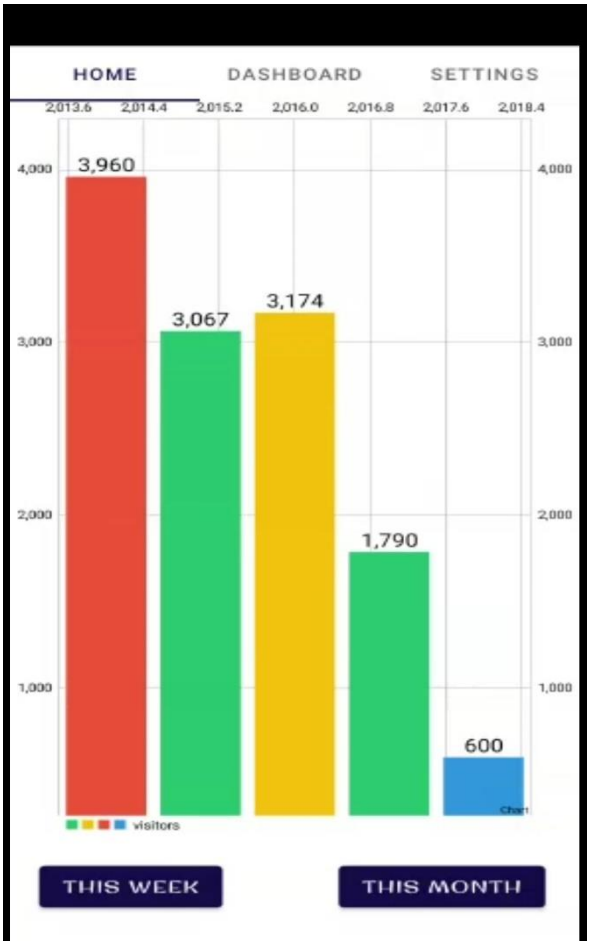

Fig.4.14: Home Screen

Dashboard Screen

The dashboard screen consists of five tabs namely ‘Update Limit’, ‘Expenses’, ‘Upload Bills’, ‘Bill Extractions’ and ‘Suggestions’ tab. Each tab has its own functionality as explained below.

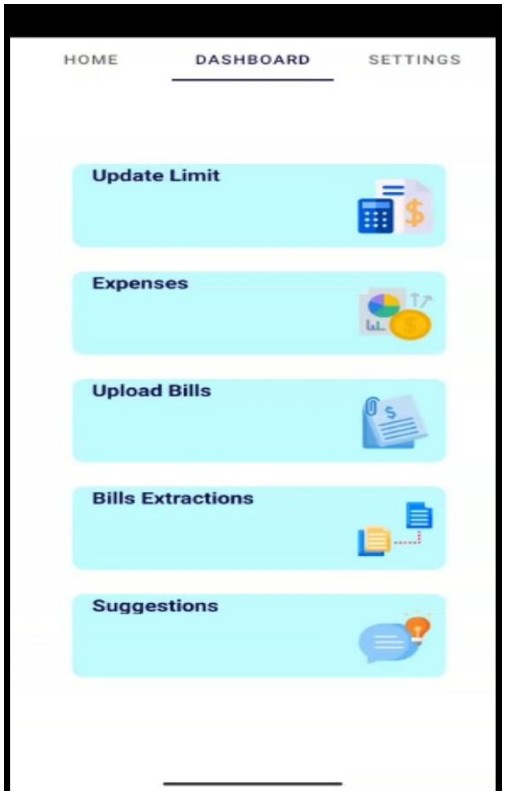

Fig.4.15: Dashboard Screen

- In the ‘Update limit’ tab the user can update the monthly limit and the monthly expense set by them earlier. In the expense tab the user can view the pie chart of the monthly expenses as well as the user can add the expenses manually in this section as shown below. This can be done by entering the price of the item and selecting a category from the drop-down menu provided by us, the pie chart and the monthly limit will be updated accordingly and automatically. In the expense tab the user can view the pie chart of the monthly expenses as well as the user can add the expenses manually in this section as shown below.

Fig.4.16: Monthly Expense Pie Chart

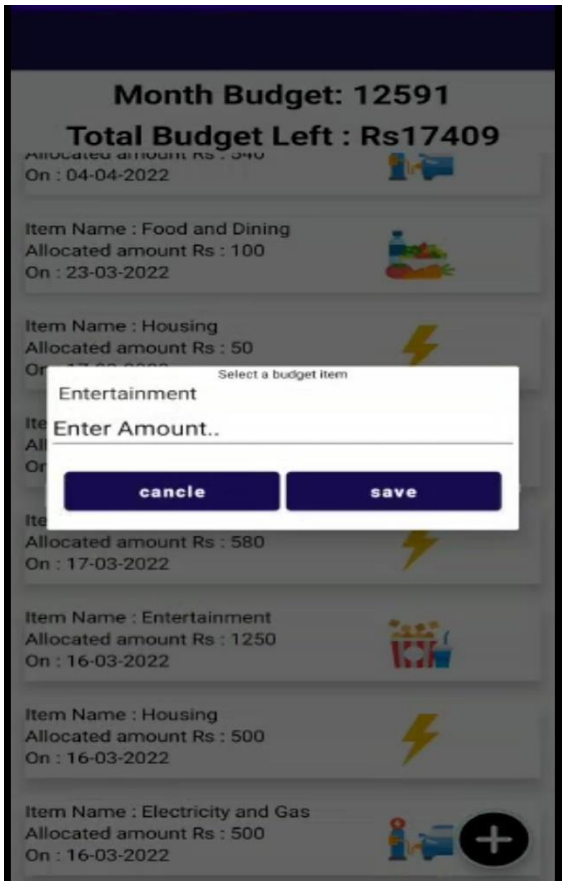

Fig.4.17: Add Expenses

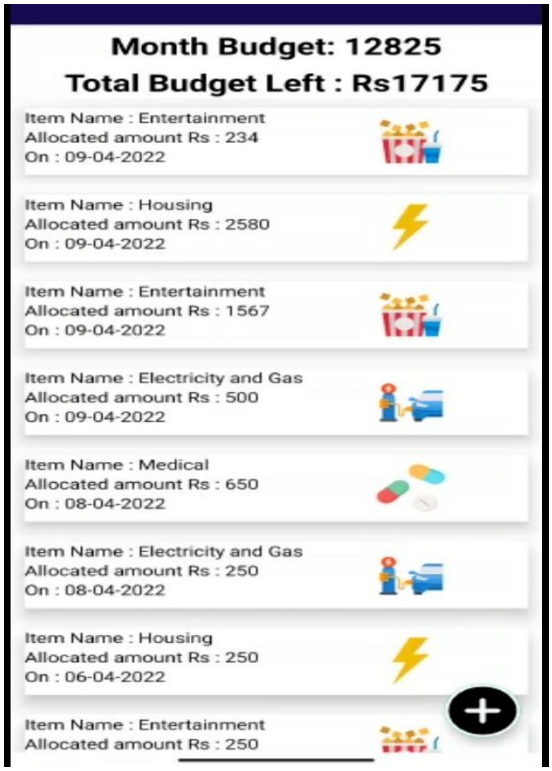

Fig.4.18: Monthly Expenses

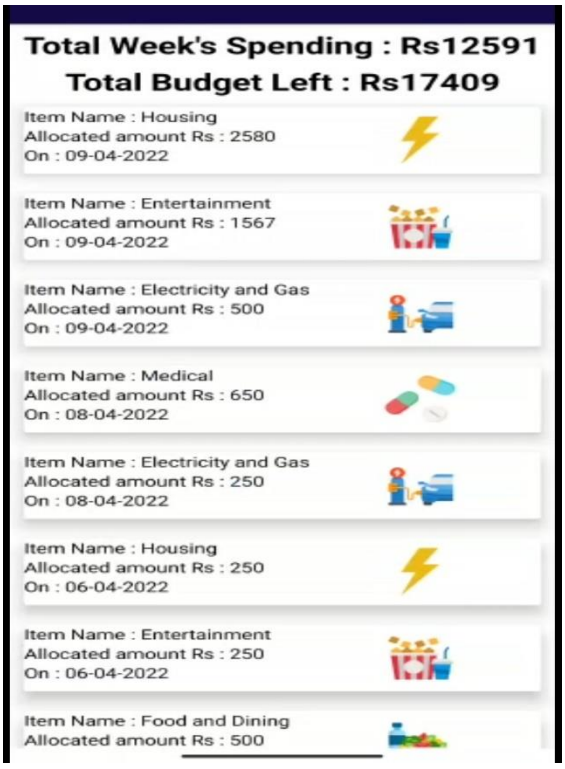

Fig.4.19: Weekly Expenses

- The ‘Upload Bills’ is the tab where the user can upload a scanned receipt. This can be done by clicking the image of the receipt or choosing it from the gallery. This uploaded bill will be used to extract the data and sort it into its respective category. This uploaded bill can be viewed by the user anytime in the ‘View Bills’ tab along with the date on which it was uploaded.

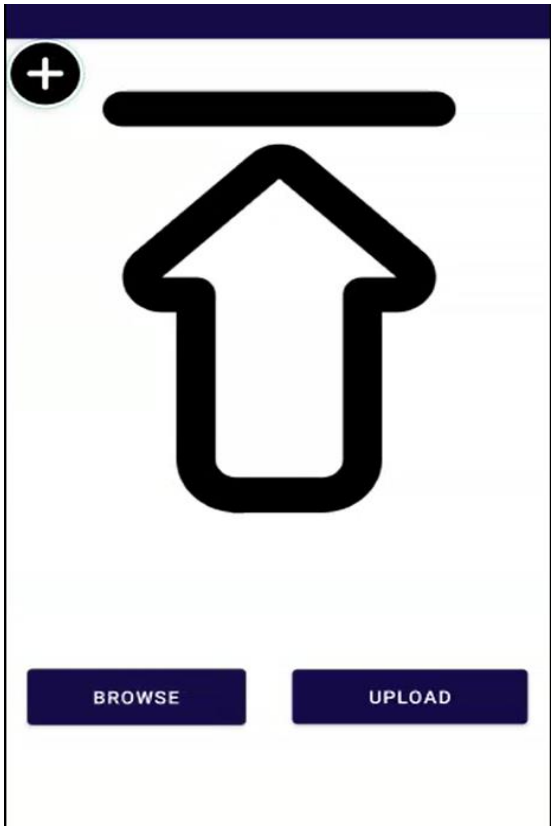

Fig.4.20: Upload Receipt Screen

Fig.4.21: Uploaded Receipt

- In the ‘Bills Extraction’ tab the user can view the extracted data from the receipts or bills in their respective category. The categories are namely ‘Food & Dinning’, ‘Entertainment’, ‘Medical’, ‘Travel’ and ‘Housing’. All the items with the help of the python program mentioned earlier is sorted into their respective category, if there is not item for a particular category it will display no data as shown below.

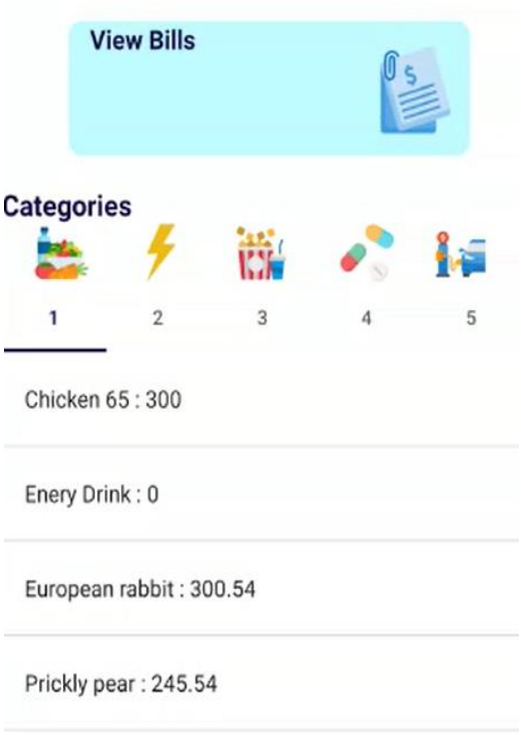

Fig.4.22: Extracted Data for Food

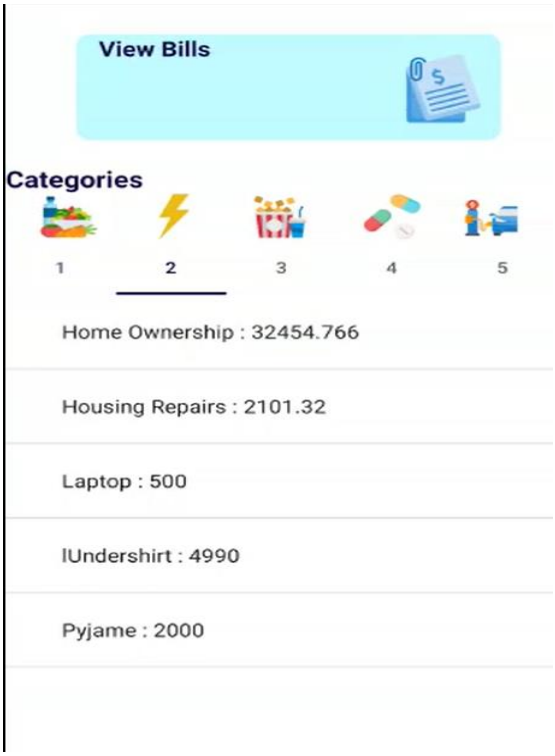

Fig.4.23: Extracted Data for Housing

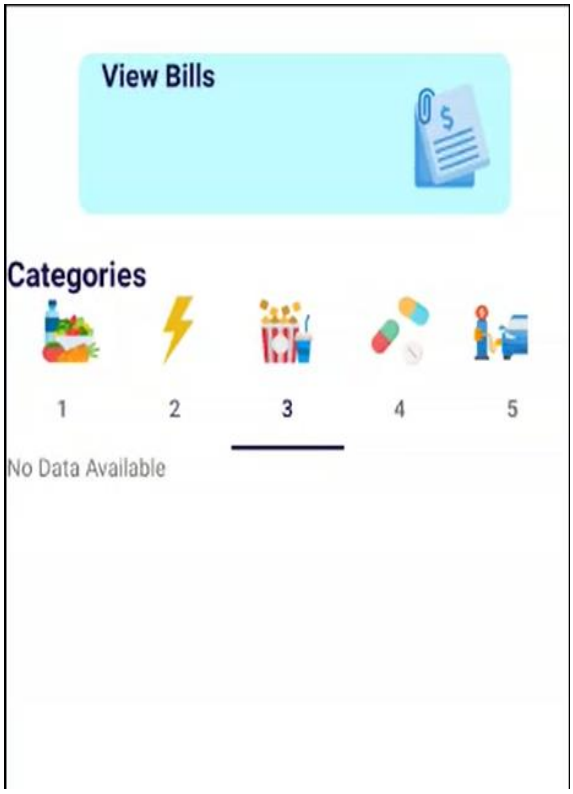

Fig.4.24: No Data Available

Fig.4.25: View Uploaded Bills

- Final tab on the dashboard screen is the ‘Suggestions’ tab which will display a message for the user, after analysing and understanding the expenses and expense pattern for a particular month the application will come up with a suggestion for the user. This suggestion is displayed in the suggestions tab as shown below.

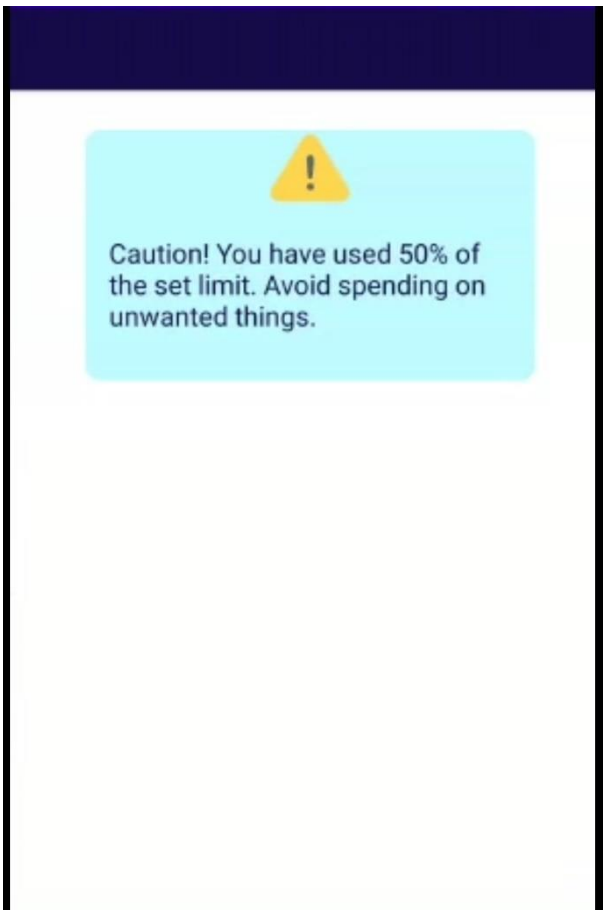

Fig.4.26: Suggestion

Settings

The third fragment in this application is the ‘settings’ fragment. This gives the user the option to view the previously uploaded bills, to view the daily, weekly and monthly statistics and to logout of the application as shown below. The uploaded bills are also stored in the firebase database, all the previously uploaded bills will be available in this section along with the date on which it was uploaded. Once the user logs out of the application the state of the application is saved and after the user logs back into the application, he/she will be able to see previously updated changes.

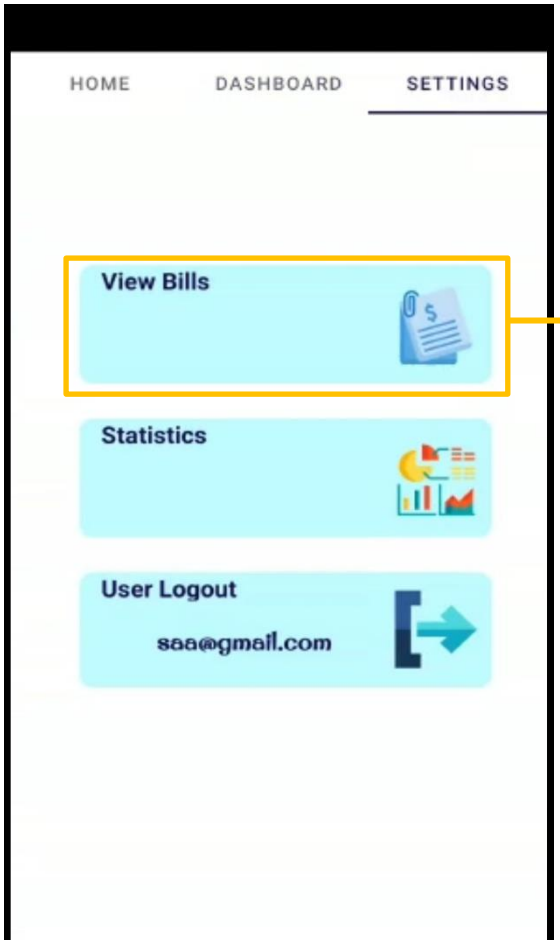

Fig.4.27: Settings Screen

Fig.4.28: Uploaded Bills History

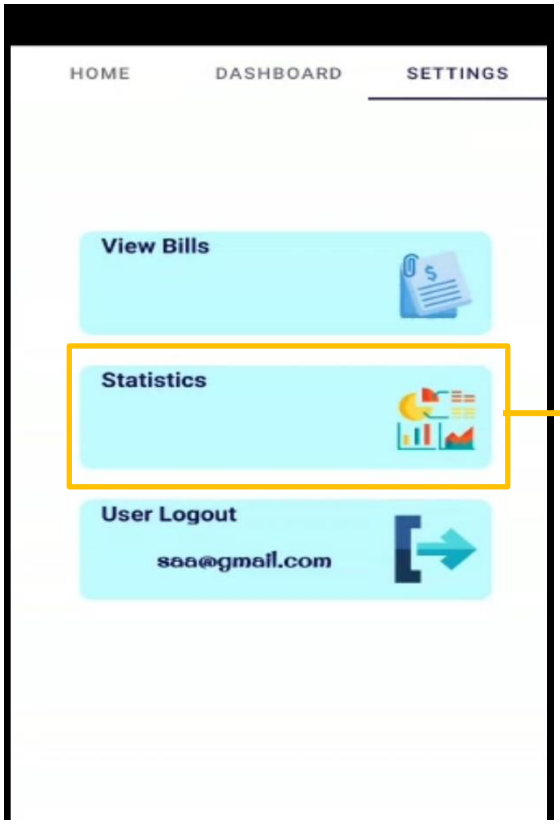

Fig.4.29: View Statistics

Fig.4.30: Statistics Screen

In the statistics activity the user has the option to view the daily, weekly and monthly statistics. The data is displayed in textual as well as graphical format in the form of pie-chart as shown below.

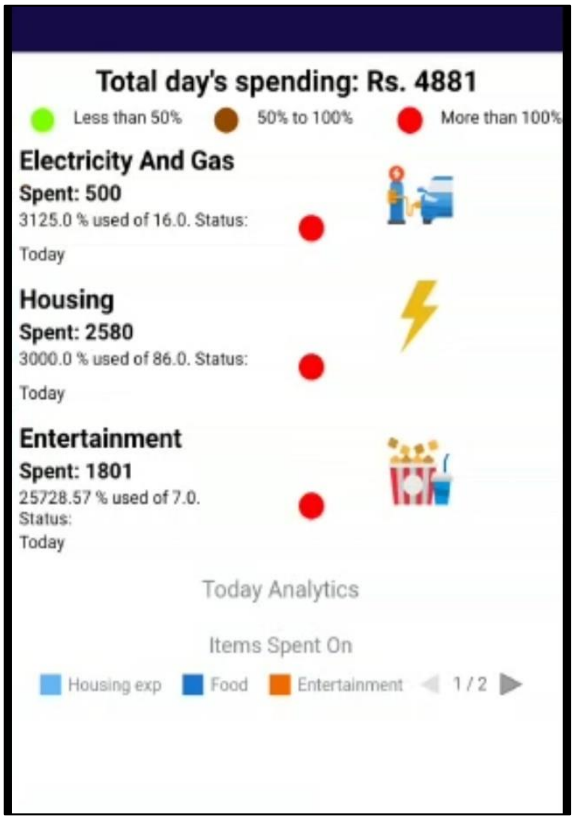

Fig.4.31: Daily Expense

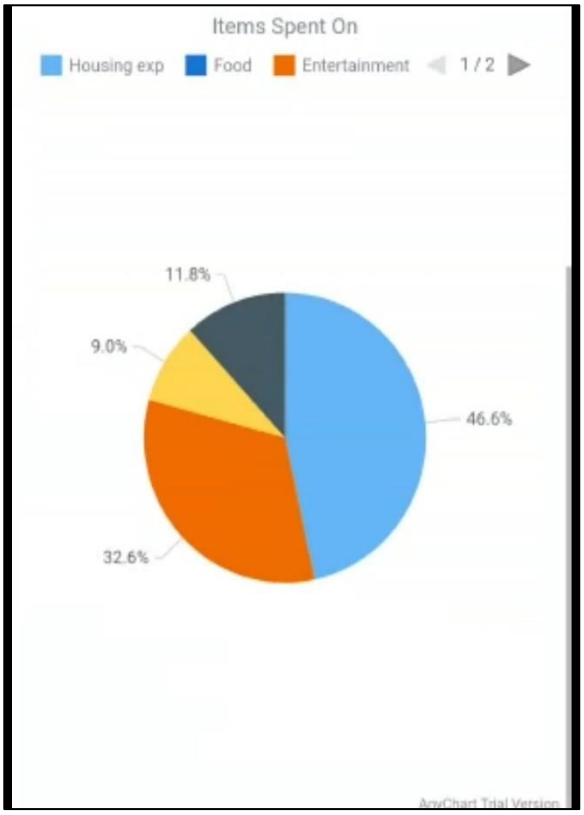

Fig.4.32: Expenses sorted by Percentage

The above image shows that the total expense of the user for that particular day was Rs. 4881 which comprised of Rs.500 for electricity and gas, Rs. 2580 for Housing and Rs. 1801 for Entertainment. Here the user has crossed the set limit for the above mentioned three categories which is depicted by small red circle. If the expenses would have been less than 50% of the set limit then it would have been depicted by a green circle on the other hand if the expenses would have been between 50% to 100% of the set limit it would have been depicted as a brown circle. This promisingly helped the user understand his expense pattern and in which category he is spending more which will be used to reduce the expenses in the future. The above pie chart displays expenses done by the user in each category converted to percentage. The conclusion that can be derived from the above chart is that maximum expense lies in housing category while minimum lies in medical category i.e., 46.6% and 9% respectively.

CHAPTER 5

RESULTS AND DISCUSSIONS

RESULTS AND DISCUSSIONS

In the proposed system we implemented image processing libraries - OpenCV and PIL to recognize the text in scanned receipts and bills. The input receipts were successfully processed to give necessary outputs such as date of transaction, item name and cost of the item. This extracted data was satisfactorily sorted into different categories and was displayed to the user in the form of bar graph as shown below.

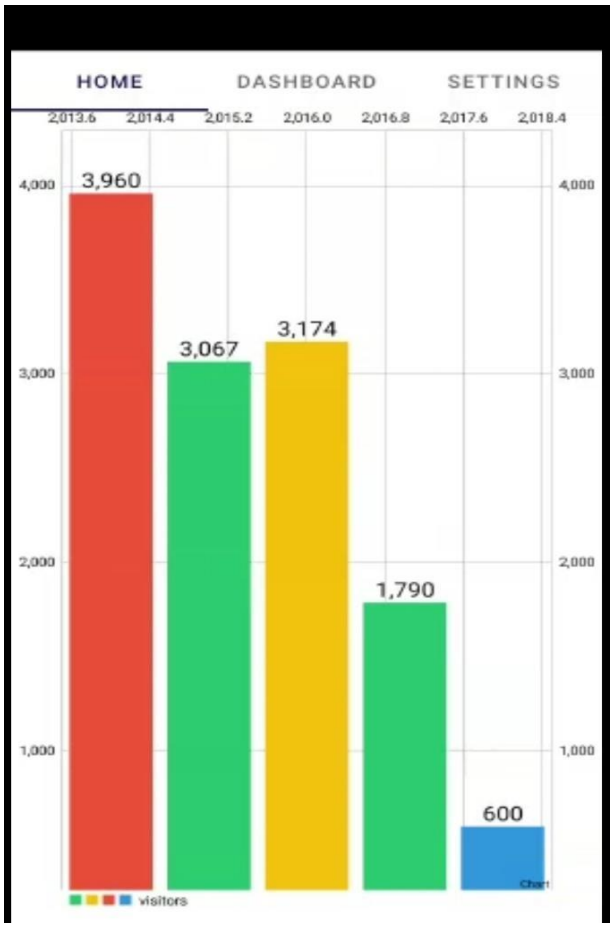

Fig 5.1: Sorted items with respective expenses

The above bar graph shows the monthly expenses of a user sorted into their respective categories i.e., ‘Food & Dinning’, ‘Transportation’, ‘Housing’, ‘Entertainment’ and ‘Medical’ respectively. The above bar chart depicts that the user has spent maximum that is Rs.3960 and minimum that is Rs.600 in Entertainment and Housing category respectively.

The application also displayed the daily expenses of the user and successfully showed if the expenses crossed the set limit. The data was displayed in text as well as graphical format as shown below.

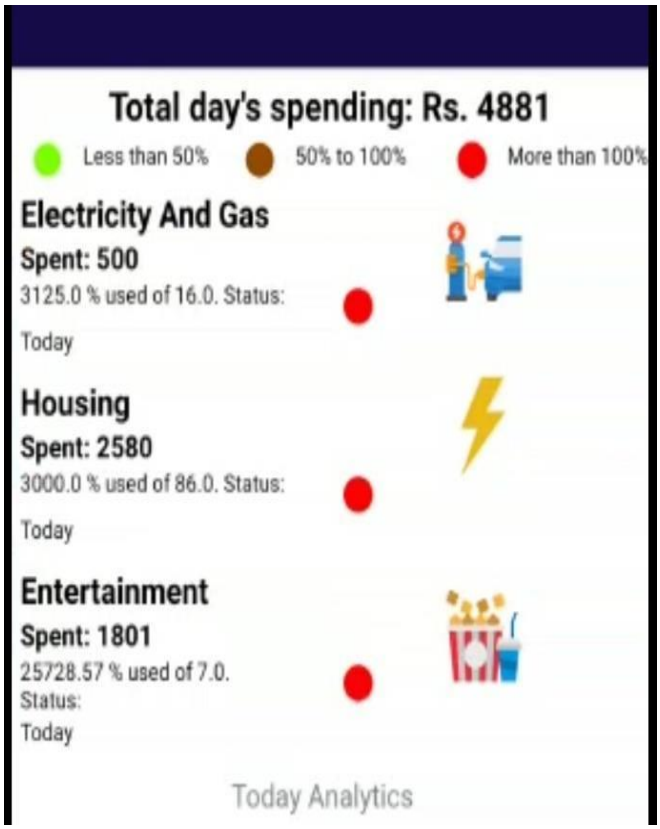

Fig 5.2: Daily Expenses

The above image shows that the total expense of the user for that particular day was Rs. 4881 which comprised of Rs.500 for electricity and gas, Rs. 2580 for Housing and Rs. 1801 for Entertainment. Here the user has crossed the set limit for the above mentioned three categories which is depicted by small red circle. If the expenses would have been less than 50% of the set limit then it would have been depicted by a green circle on the other hand if the expenses would have been between 50% to 100% of the set limit it would have been depicted as a brown circle. This promisingly helped the user understand his expense pattern and in which category he is spending more which will be used to reduce the expenses in the future.

Graphical representation of the expense data is as shown below: -

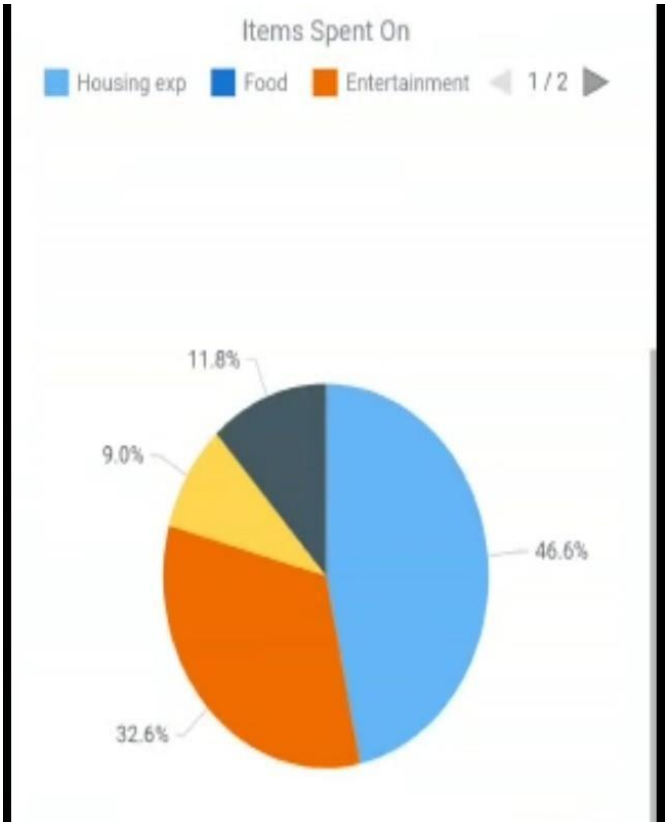

Fig 5.3: Expenses in each category by Percentage

The above pie chart displays expenses done by the user in each category converted to percentage. The conclusion that can be derived from the above chart is that maximum expense lies in housing category while minimum lies in medical category i.e., 46.6% and 9% respectively.

The extracted text from the receipt was also displayed in textual format sorted into respective categories. This extracted text was visible in the ‘Bills Extractions’ Tab as shown below.

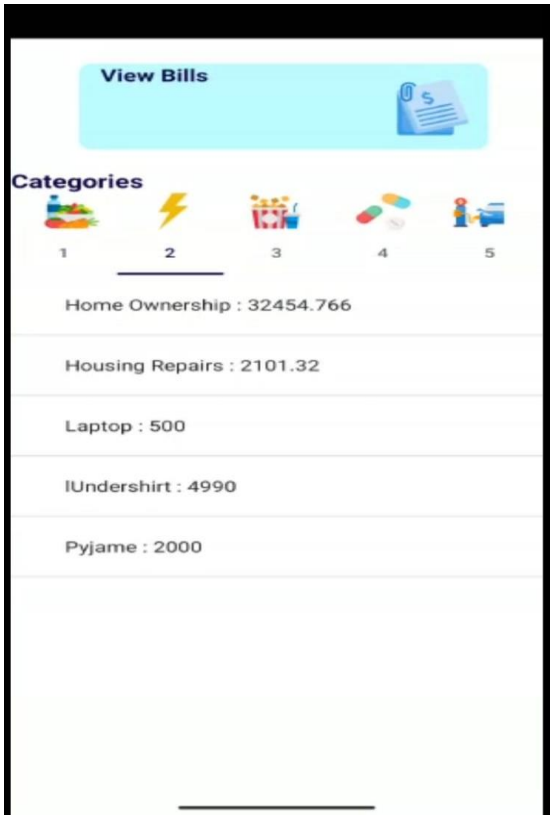

Fig 5.4: Textual representation of sorted items

The third important feature of the application of giving timely suggestions to the user was also successfully implemented. After analyzing and understanding the expenses and expense pattern of the user for a particular month the application came up with a suggestion for the user. The suggestion was displayed in the suggestions tab as shown below.

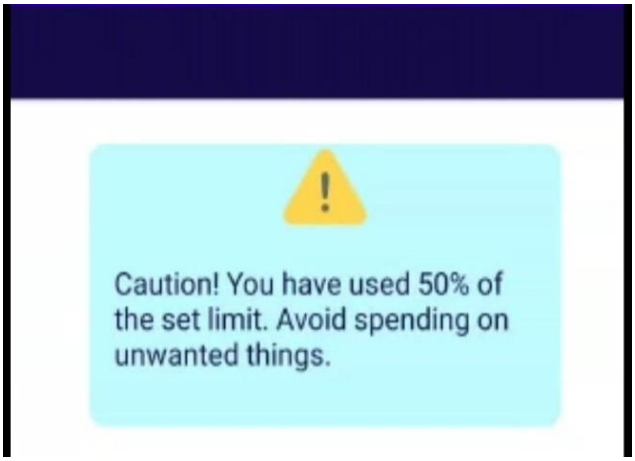

Fig. 5.5: Suggested Message

CHAPTER 6

CONCERNS

CONCERNS

Q.1 Why is there a need for Expense Manager?

- On average, more than half of the individuals in India and all over the world are constantly worried and stressed about their finances.
- Individuals are clueless about how much is their inflow and how much they are spending and hence they don't know at the end of the month what happened to their money.
- Individuals get into debt and it becomes difficult to pay back due to poor money management skills.
- Thus, there is a definite need for an automated expense manager which will manage your expenses wisely and effortlessly.

Q.2 How does it help?

The proposed system helps in the following ways:

- Expense Monitoring
- Analyze Buying Trends
- Helps in Financial Goals
- Better Sense of Expenditure

Q.3: How accurate is text recognized and sorted into categories?

In the proposed system we have used Tesseract OCR, Tesseract is an open-source project that is supported by Google. This is a desktop application where you can select which images to extract text from locally and OCR stands for 'Optical Character Recognition' which is a technology that automatically recognizes characters in images and processes this into a text.

The specific data points that were extracted from the receipts were 'item name', 'cost' and 'date

of purchase', item name and date gave an accuracy of 72.32% while price showed an accuracy of 81.2%.

Out of the 30 receipts scanned by the application 22 receipts were scanned accurately giving 73.3% accuracy as shown below.

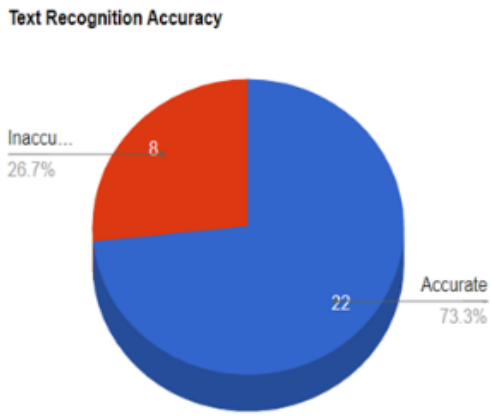

Fig 6.1: Text Recognition Accuracy

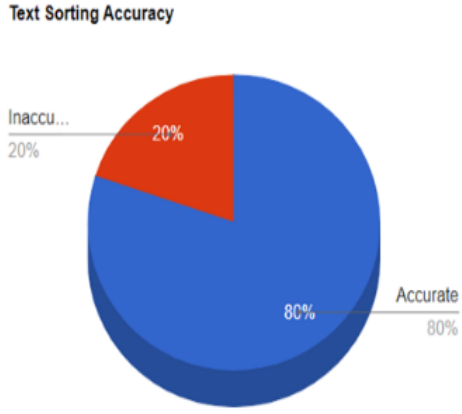

Fig 6.2: Text Sorting Accuracy

Table 6.1: Accuracy for Data Points

Data Point	Accuracy
Item Name	72.32%
Cost	81.2%
Date of Purchase	72.32%

Sorting of the items present in the receipt was done by traversing the huge dataset which consists of different categories like 'clothing', 'food', 'groceries' etc. and list of all items in its respective category. Sorting the items into different categories gave an accuracy of 80% as shown above which can be improved by increasing the dataset in the future.

Q.4: How is Expense Manager different than the expense tracking applications already present in the market?

We have tried to keep the proposed application as unique as possible except for features which is mandatory for any expense tracker all other features are unique. The key distinct features of this application are: -

- Gives you indication that overspent for this month.
- No need to feel long forms for using application.
- Information displayed in simplified way.
- Daily, Weekly, Monthly Analysis.
- Free of cost.

Q .5: What are the improvements that can be done to the application?

A possible source of error could be our definition on how to structure and create the different categories. At present we are using a already created dataset. It is possible that a separate algorithm can be used to automatically cluster the data into sections and if used as a base for constructing and dividing the categories, this might yield better results.
Increasing the data set size would also give more indicative results and will efficiently improve the application. Another possible problem with our solution is how well these results would map to unseen data. This can also be improved by increasing the number of receipts and using different patterns and types of receipts in the future.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

The major advantage of an Expense Manager Application is its ability to automatically and efficiently sort items present in the bills or receipts into different categories without the need of manually adding each item to its respective category. Use of image processing and machine learning techniques will prove to be useful giving more accurate results. The timely suggestions provided by the application will prove to be useful for everyone who is using this application to track their expenses. This application will successfully help all those people who wants to track their expenses to get a better sense of their expenditure thereby promoting savings. The development of this application has been conducted in a stepwise manner using the well-defined methodology, customized according to the requirements of the system. Most
of the goals set at the beginning of the development phase have been met.

7.2 Future Scope

There exists a huge scope of advancements in the present form of image processing-based expenses tracking application. More features can be compounded in this application like we can integrate this app with any of our credit / debit / cash card & keep track directly from our saving account. Feature of splitting the bills among friends and family can be integrated. Retrieving bills directly from SMS, emails or UPI transactions will add to a very good feature. Changing the language of the application can also be integrated to reach a larger and a global audience.

The accuracy of text recognition and sorting the items can be further improved further by using advanced image processing techniques as well as by training the model with more number of receipts which will help the system to understand different patterns of receipt which will greatly increase the efficiency of the application. Some handwritten text recognition can also be studied to scan handwritten bills.

REFERENCES

REFERENCES

[1] Rajaprabha, M. N. (2017). Family Expense Manager Application in Android. IOP Conference Series: Materials Science and Engineering, *263*, 042050. doi:10.1088/1757-899x/263/4/042050.

[2] Joel Odd Emil Theologou. (2018). Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms. Linköping University Department of Computer and Information Science.

[3] Le, A. D., & Nakagawa, M. (2017). Training an End-to-End System for Handwritten Mathematical Expression Recognition by Generated Patterns. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). doi:10.1109/icdar.2017.175

[4] Anh Duc Le 1,2, Dung Van Pham2, Tuan Anh Nguyen2 (2019). Deep Learning Approach for Receipt Recognition. Center of Open Data in Humanities Research Organization of Information Systems Tokyo, Japan & Deep Learning and Application, Vietnam.

[5] Satav, M. S., Varade, T., Kothavale, D., Thombare, S., & Lokhande, P. (2020). Data Extraction from Invoices Using Computer Vision. 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS). doi:10.1109/iciis51140.2020.9342722

[6] Bekarao Girish., & Sunhaloo Sameer (2007). Intelligent Online Budget Tracker. Proceedings of the 2007 Computer Science and IT Education Conference [7] Shi, B., Bai, X., & Yao, C. (2017). An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(11), 2298- 2304. doi:10.1109/tpami.2016.2646371R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[8] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, Jiajun Liang, EAST: An Efficient and Accurate Scene Text Detector, CVPR 2017, pp. 5551-5560

[9] Baoguang Shi, Xiang Bai, Cong Yao, An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition, PAMI, pp. 2298--2304.

[10] D. Jurafsky and J.H. Martin. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and SpeechRecognition. Pearson Prentice Hall.

[11] Ravina Mithe, Supriya Indalkar, and Nilam Divekar.2013. Optical character recognition. International journal of recent technology and engineering (IJRTE) 2, 1 (2013),72–75.

[12] Chandini, S., et al. "Online Income and Expense Tracker." International Research Journal of Engineering and Technology (IRJET) 6.3, 2019, pp. 2395-0056.

PAPER PUBLISHED

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

Expense Manager: An Expense Tracking Application using Image Processing

Nupur Sawarkar[1], Pranay Yenagandula[2], Devang Shetye[3], Prof. Shruti Agrawal[4]

[1,2,3]*Student, Dept. of IT Engineering, Vidyalankar Institute of Technology, Mumbai, India*
[4]*Professor, Dept. of IT Engineering, Vidyalankar Institute of Technology, Mumbai, India*

Abstract - *We present an intelligent expense tracker to efficiently manage the monthly expenses. Our system will help everyone who are planning to know their expenses and save from it. The user will be given the facility to set a monthly limit and if the user crosses that limit our app will notify the user about the same. The user can give receipts as an input, using AI our app will sort it into different categories. Here user can also define their own categories like food, clothing, rent and bills and the user can also set limits for a particular category. User will be provided with visual statistics of expenses by transaction date or by category. This project is not indented for a particular user or age group but anyone and everyone who wants to track their expense can use this app. So, the general idea of this Project is to help people view and study their overall expenditure pattern by developing a mobile application to analyse all the purchases made by the user by simply scanning the receipts.*

***Key Words*: Android Application, Receipt Recognition, Image Processing, Expense Management.**

1. INTRODUCTION

In order to trace their expenses, People generally use traditional paper system to stay the record of their income and expenditures. this sort of traditional system is burdensome and takes longer. So, there must be a management system which must help us to manage our daily earnings and expenses easily, and also helps us to find the records efficiently. So, we puzzled out the simplest way to eliminate the normal system with digital, portable, easier and easy to record these data in precisely few clicks with our Android application called "Expense Manager". The thought of this Project is to assist people view and study their overall expenditure analysis with the assistance of a mobile app to research all the purchases made by the user by simply scanning all the bills and receipts. The User should just click a picture of a receipt or choose one from the Gallery and track their payments. Using machine learning, the app is able to group items category-wise, for example- food, clothes, stationery, etc. The User can monitor his/her expenses and analyse them categorically using graphs and tables, enabling them to raised understand their expense patterns and help them spend intelligently. Using this app, the users may set monthly limits over specific categories helping them to avoid over-spend on those items. The app also will prompt users after they overspend or make repeat purchases. Our main aim is to create our users capable of achieving Personal Life goals by giving them the flexibility to – monitor their expenses; analyse the buying trends; and assess their account's future transactions. And hence, the most objectives of developing this intelligent expense tracker are to supply a higher sense of expenditure to our users thereby promoting savings.

2. RELATED WORK

A. Family Expense Manager Application in Android

There is numerous app within the markets now to trace & keep recorder of our spending's. applications like spend book, pocket graud, home budget, Wally, level money, spendee, every dollar so forth. an intensive a part of these application screens our money related adjust and learn the expenses by that. All those applications have only record of bill paid. This work like dairy & save your & your family bill record which has been settled or in pending to [1]. we will mark the status of our bill. this is often suitable for socio-economic class people that always don't use to settlements and find things on credits [1].

B. Deep Learning Approach for Receipt Recognition Automatically identifying and extracting key texts from scanned structured and semi-structured receipts and saving them as structured data can function a really beginning to enable a range of applications. Although there are significant improvements of OCR in many tasks receipt OCR continues to be challenging thanks to its higher requirements in accuracy to be practical. Therefore, fast and reliable OCR has to be developed so as to scale back or maybe eliminate manual works. during this paper, a deep learning approach for text detection and recognition are presented. CTPN for text detection and AED for text recognition were employed. Moreover, preprocessing and OCR certification were employed to extract small receipt and take away handwriting, respectively [4].

C. OCR Text to Extract Receipt Data and Classify Receipts with Common Machine Learning Algorithm Most companies have some type of system to handle

receipts from differing kinds of purchases for the corporate. Usually, the receipts are handled manually and entered into a database [2]. This sequence requires time which makes it a fashionable process. A more practical solution can be to convert this manual process to an automatic one. this might be done by taking a picture of the receipt so let a program automatically extract the knowledge and input that data on to the database. Extracting the text from the image can be done by common machine learning algorithm.

3. Proposed System

The application will provide the following functionalities: -

- Upload a receipt by clicking an image or choosing from Gallery and directly convert them into expenses by items.
- The Application will categorize these items in specific categories (such as Groceries, Clothes, Stationery, etc.) which will help the user to manage their budget wisely ensuring an efficient budget.
- Users can set monthly limits on their expenses which will be tracked daily and help user spend their future expenses wisely.
- An Alert will be generated in case the user crosses their monthly limit helping them decide to reduce or stop further expenses as necessary.

User can also monitor his/her expenses and analyze categorically using graphs and tables provided by the application.

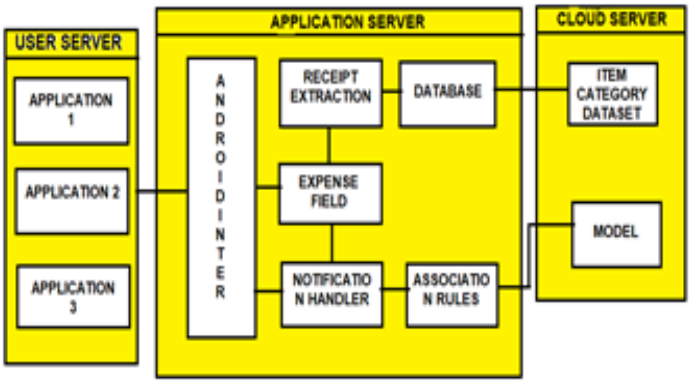

Fig -1. Flow Chart of System Architecture

The architecture as shown above has three blocks program, application server and cloud server. The interface displays the appliance with which the user can interact while the cloud server consists of various datasets and AI model. the appliance server further consists of the subsequent modules android interface, receipt extraction, expense field, notification handler, association rules and database which is required for appropriate action.

The usual process of the architecture is as follows: -

- The user sends appropriate command or request to the server through the net, using the interface of the applying.
- The application server is accountable for forwarding the command to the requested server.
- The server finds the results of requested commands (either the info processing or the database querying).
- The software or application delivers the processed information to the server.
- The server provides the user with the requested data.

4. Methodology

A. Receipt Recognition

In the proposed system we are using Image Processing libraries provided by python to process the receipts provided by the user. Image processing basically includes the following three steps:

- Importing the image via image processing tools;
- Analyzing, manipulating the image;
- Proving an output which can be an altered image or conclusion based on image analysis [3].

So, for this project will process all our receipts to provide us with necessary output such as

- Store Name
- Date of Transaction
- Item Name
- Cost of Items
- Total Cost

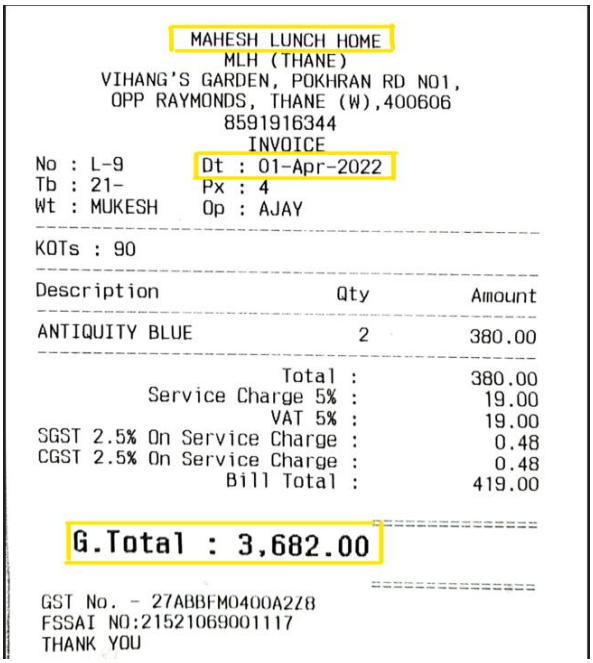

Fig -2. Text Recognition

B. Sorting Items into Different Categories

Using Machine Learning algorithms in python like tesseract and pytorch the application will use keywords such as Product and its Price from the receipts and categorize them into various categories. These categories will be general classes such as Groceries, Clothing, Stationery, etc. These Products will be matched against a big database of such categories which will help determine which product belongs to which category. Once all the Products are sorted, the User will be able to view the expenses according to categories helping them analyze their expenses better. If there is any Product which doesn't find any category, with the help of AI we will later sort it into one of the categories or classify it as Miscellaneous.

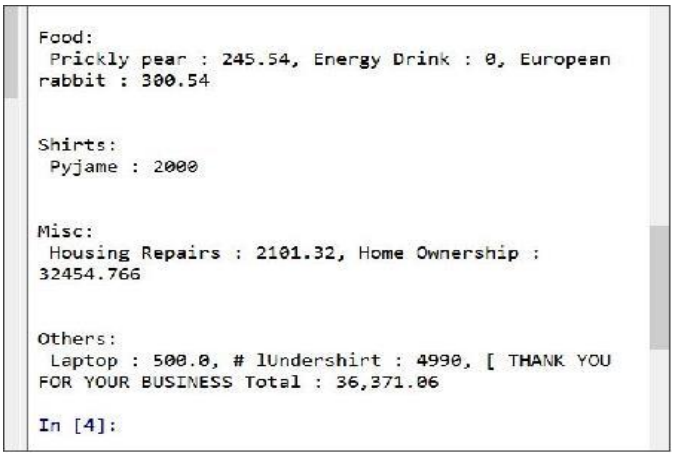

Fig -3. Items Sorted into Different Categories

C. Machine Learning

Machine learning may be a method of knowledge analysis that automates analytical model building [7]. it's a branch of computer science supported the thought that systems can learn from data, identify patterns and make decisions with minimal human intervention. In our project, we'll be creating Machine Learning Models to –

- Train the model to acknowledge receipt formats employing a data-set of just about 1000 receipts of various formats to extend the efficiency of the system against new or differing types of receipts.
- Use various Machine Learning algorithms like CTPN, LSTM and AED to induce better and accurate results on applying receipt recognition.

D. MPAndroidChart

In this project, MPAndroidChart library is used to create beautiful, customized and animated charts. Pie charts, line graphs, bar graphs are used in this project to give the users visual representation of their expenses. The users can view these graphical reports either by transaction date or by category. Adding graphs and charts will make information clearer and therefore improve the user experience. This will give the user a better understanding of their expenditure.

5. RESULTS

In this project we implemented image processing libraries-OpenCV and PIL to recognize the text in scanned receipts and bills. The input receipts were successfully processed to give necessary outputs such as date of transaction, item name and cost of the item. This extracted data was satisfactorily sorted into different categories and was displayed to the user in the form of bar graph as shown below.

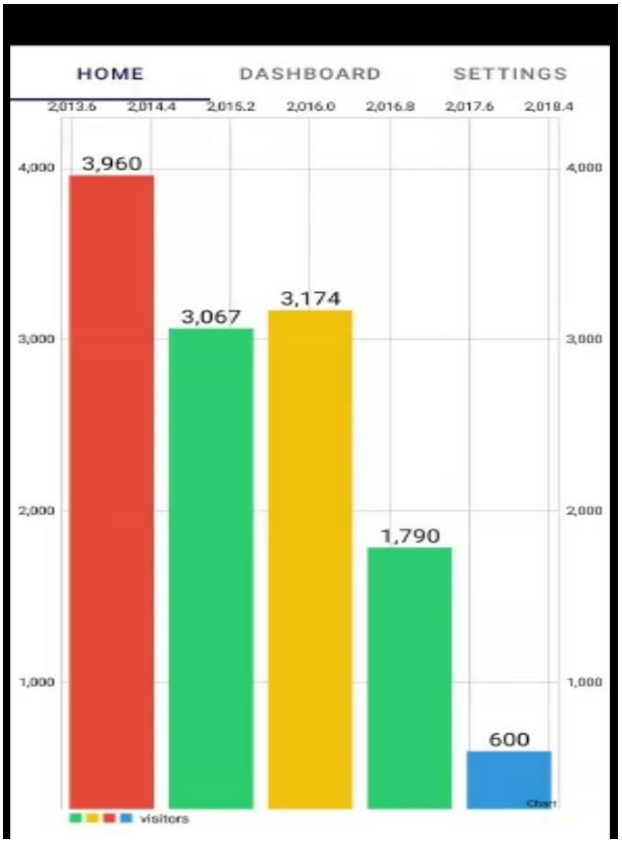

Fig -4 Sorted Items

The above bar graph shows the monthly expenses of a user sorted into their respective categories i.e., 'Food & Dinning', 'Transportation', 'Housing', 'Entertainment' and 'Medical' respectively.

The application also displayed the daily expenses of the user and successfully showed the user if the expenses crossed the set limit. The data was displayed in text as well as graphical format as shown below.

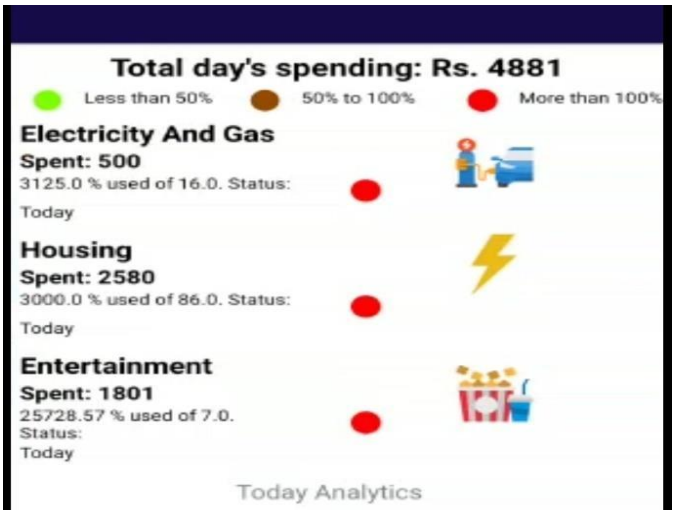

Fig -5 Daily Expenses

The above image shows that the total expense of the user or that particular day was Rs. 4881 which comprised of Rs.500 for electricity and gas, Rs. 2580 for Housing and Rs. 1801 for Entertainment. Here the user has crossed the set limit for the above mentioned three categories which is depicted by small red circle. If the expenses would have been less than 50% of the set limit then it would have been depicted by a green circle on the other hand if the expenses would have been between 50% to 100% of the set limit it would have been depicted as a brown circle. This promisingly helped the user understand his expense pattern and in which category he is spending more which will be used to reduce the expenses in the future.

Graphical representation of the expense data is as shown below: -

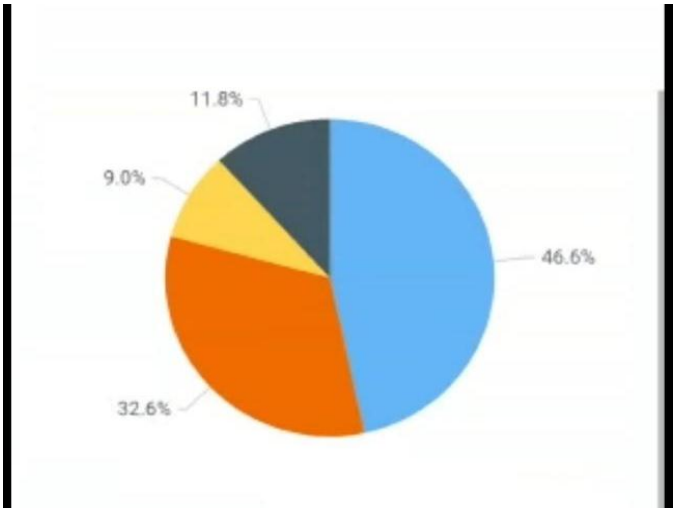

Fig -6 Expenses in each Category by Percentage

The above pie chart displays expenses done by the user in each category converted to percentage. The conclusion that can be derived from the above chart is that maximum expense lies in housing category while minimum lies in medical category i.e., 46.6% and 9% respectively.

The third important feature of the application of giving timely suggestions to the user was also successfully implemented. After analyzing and understanding the expenses and expense pattern of the user for a particular month the application came up with a suggestion for the user. The suggestion was displayed in the suggestions tab as shown below.

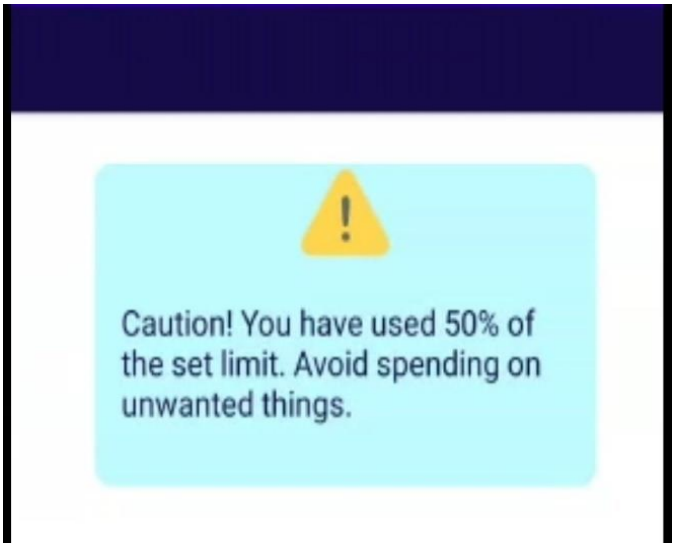

Fig -7 Suggested Message

6. CONCLUSION AND FUTURE WORK

We have presented a working prototype of intelligent expense tracker. The development of this application has been conducted in a stepwise manner using the well-defined methodology, customized according to the requirements of the system. This application will successfully help all those people who wants to track their expenses to get a better sense of their expenditure thereby promoting savings. The development of this application has been conducted in a stepwise manner using the well-defined methodology, customized according to the requirements of the system. Most of the goals set at the beginning of the development phase have been met. Future work can be like we can integrate this app with any of our credit / debit / cash card & keep track directly from our saving account. Once when we start doing that this will help us to update details automatically rather than updating in manually.

REFERENCES

[1] Rajaprabha, M. N. (2017). Family Expense Manager Application in Android. *IOP Conference Series: Materials Science and Engineering, 263*, 042050. doi:10.1088/1757- 899x/263/4/042050.

[2] Joel Odd Emil Theologou. (2018). Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms. *Linköping University Department of Computer and Information Science.*

[3] *Le, A. D., & Nakagawa, M. (2017). Training an End-to End System for Handwritten Mathematical Expression Recognition by Generated Patterns. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). doi:10.1109/icdar.2017.175*

[4] Anh Duc Le 1,2, Dung Van Pham2 , Tuan Anh Nguyen2

(2019). Deep Learning Approach for Receipt Recognition.*Center of Open Data in Humanities Research Organization of Information Systems* Tokyo, *Japan & Deep Learning and Application, Vietnam.*

[5] *Satav, M. S., Varade, T., Kothavale, D., Thombare, S., & Lokhande, P. (2020). Data Extraction from Invoices Using Computer Vision. 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS). doi:10.1109/iciis51140.2020.9342722*

[6] Bekarao Girish., & Sunhaloo Sameer (2007). Intelligent Online Budget Tracker. *Proceedings of the 2007 Computer Science and IT Education Conference*

[7] Shi, B., Bai, X., & Yao, C. (2017). An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(11), 2298- 2304. doi:10.1109/tpami.2016.2646371R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[8] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, Jiajun Liang, EAST: *An Efficient and Accurate Scene Text Detector*, CVPR 2017, pp. 5551-5560.

[9] Baoguang Shi, Xiang Bai, Cong Yao, *An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition*, PAMI, pp. 2298--2304.

[10] D. Jurafsky and J.H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and SpeechRecognition*. Pearson Prentice Hall.

[11] Ravina Mithe, Supriya Indalkar, and Nilam Divekar.2013. *Optical character recognition. International journal of recent technology and engineering* (IJRTE) 2, 1 (2013),72–75.

CERTIFICATIONS

e-ISSN: 2395-0056 p-ISSN: 2395-0072

International Research Journal of Engineering and Technology (IRJET)

(An ISO 9001 : 2008 Certified Journal)

Is hereby awarding this certificate to

Nupur Sawarkar

In recognition the publication of the manuscript entitled

Expense Manager: An Expense Tracking Application using Image Processing

published in our Journal Volume 9 Issue 4 April 2022

Editor in Chief

E-mail : editor@irjet.net

Impact Factor : 7.529

www.irjet.net

e-ISSN: 2395-0056 p-ISSN: 2395-0072

International Research Journal of Engineering and Technology (IRJET)

(An ISO 9001 : 2008 Certified Journal)

Is hereby awarding this certificate to

Pranay Yenagandula

In recognition the publication of the manuscript entitled

Expense Manager: An Expense Tracking Application using Image Processing

published in our Journal Volume 9 Issue 4 April 2022

Editor in Chief

Impact Factor : 7.529

www.irjet.net

E-mail : editor@irjet.net

e-ISSN: 2395-0056 p-ISSN: 2395-0072

International Research Journal of Engineering and Technology (IRJET)

(An ISO 9001 : 2008 Certified Journal)

Is hereby awarding this certificate to

Devang Shetye

In recognition the publication of the manuscript entitled

Expense Manager: An Expense Tracking Application using Image Processing

published in our Journal Volume 9 Issue 4 April 2022

Editor in Chief

Impact Factor : 7.529

www.irjet.net

E-mail : editor@irjet.net

e-ISSN: 2395-0056 p-ISSN: 2395-0072

International Research Journal of Engineering and Technology (IRJET)

(An ISO 9001 : 2008 Certified Journal)

Is hereby awarding this certificate to

Prof. Shruti Agrawal

In recognition the publication of the manuscript entitled

Expense Manager: An Expense Tracking Application using Image Processing

published in our Journal Volume 9 Issue 4 April 2022

Impact Factor : 7.529

www.irjet.net

Editor in Chief

E-mail : editor@irjet.net