

**A  
Project Report  
On  
"TATHYA – The Intelligent Document Extraction Engine"**

(CS453 - Software Project Major)



**Prepared by**  
Parth Savaliya – D20DCS163

**Under the Supervision of**  
Prof. Nilesh Dubey

**Submitted to**  
Charotar University of Science & Technology (CHARUSAT)  
for the Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Technology (B.Tech.)  
in Computer Science & Engineering (CSE)  
for 8<sup>th</sup> semester B.Tech

**Submitted at**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**Devang Patel Institute of Advance Technology and Research (DEPSTAR)**  
**Faculty of Technology & Engineering (FTE), CHARUSAT**  
**At: Changa, Dist: Anand, Pin: 388421.**  
**April, 2023**

## **DECLARATION BY THE CANDIDATE**

I hereby declare that the project report entitled “**TATHYA – The Intelligent Document Extraction Engine**” submitted by me to Devang Patel Institute of Advance Technology and Research, Changa in partial fulfilment of the requirement for the award of the degree of **B.Tech** in Computer Science & Engineering, from Department of Computer Science & Engineering, DEPSTAR-FTE, CHARUSAT, is a record of bonafide CS453 Software Project Major (project work) carried out by me under the guidance of **Prof. Nilesh Dubey**. I further declare that the work carried out and documented in this project report has not been submitted anywhere else either in part or in full and it is the original work, for the award of any other degree or diploma in this institute or any other institute or university.

Parth Savaliya (D20DCS163)

Prof. Nilesh Dubey  
Assistant Professor  
Department of Computer Science & Engineering,  
DEPSTAR-FTE, CHARUSAT-Changa.

25<sup>th</sup> April, 2023

**Subject: Internship Completion Certificate**

**TO WHOMSOEVER IT MAY CONCERN**

This is to certify that **Mr. Parth Savaliya** who is pursuing **Bachelor in Computer Science & Engineering** at **Devang Patel Institute of Advance Technology and Research** affiliated to **Charotar University Science and Technology** (Enrollment Number **D20DCS163**), has completed his internship/project training with us, from **December 7, 2022 to March 31, 2023**.

Project Definition: "TATHYA – The Intelligent Document Extraction Engine"

We found him sincere, hardworking, dedicated and result oriented. He worked well as part of the team during this tenure.

We take this opportunity to thank him and wish him all the best for his future.

With regards,

For **Cloudoffis Technologies LLP**,



**Ankita Gajjar**  
**HR Manager**





Accredited with Grade A+ by NAAC  
Accredited with Grade A by KCG


## CERTIFICATE

This is to certify that the report entitled “**TATHYA–The Intelligent Document Extraction Engine**” is a bonafied work carried out by **Parth Savaliya (D20DCS163)** under the guidance and supervision of **Prof. Nilesh Dubey & Mrs. Mudra Vora** for the subject **Software Project Major (CS453)** of 8<sup>th</sup> Semester of Bachelor of Technology in **Computer Science & Engineering** at Devang Patel Institute of Advance Technology and Research (DEPSTAR), Faculty of Technology & Engineering (FTE) – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred by the examiner(s).

Under the supervision of,

Prof. Nilesh Dubey  
Assistant Professor,  
Department of Computer Science &  
Engineering, DEPSTAR-FTE,  
CHARUSAT, Changa, Gujarat

  
Mrs. Mudra Vora  
Technical Lead,  
Cloudoffis Technology LLP,  
Gift City, Gandhinagar, Gujarat

Dr. Chirag Patel  
I/c. Head- Department of Computer Science  
& Engineering, DEPSTAR-FTE,  
CHARUSAT, Changa, Gujarat

Dr. Amit Nayak  
I/c. Principal-DEPSTAR,  
CHARUSAT, Changa, Gujarat.

---

**Devang Patel Institute of Advance Technology and Research  
(DEPSTAR)**

**Faculty of Technology & Engineering (FTE), CHARUSAT**

At: Changa, Ta. Petlad, Dist. Anand, Pin:388421. Gujarat.

## **ABSTRACT**

Auditing is often considered a tedious and time-consuming task due to its detailed scrutiny, complex regulations, large volume of data, multiple parties involved, and continuous nature.

However, Intelligent Document Extraction and Classification can automate and streamline the data extraction and classification process, improving accuracy, efficiency, and transparency. By automating document management, document classification, and data extraction, auditors can focus on more complex tasks and analysis, ensuring that financial statements are accurate and reliable.

The use of Intelligent Document Extraction and Classification can significantly enhance the efficiency and effectiveness of the auditing process, resulting in better transparency and accountability.

TATHYA (**T**o extr**A**ct data **T**hrough p**Y**thon **A**pplication) is an innovative technology that can significantly improve the efficiency and accuracy of the auditing process. This technology uses core python to automatically extract documents based on predefined rules and categories, reducing the need for manual classification.

## **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to Prof. Nilesh Dubey, my internal university guide, for his invaluable guidance, support, and encouragement throughout the development of the TATHYA module. His insights and feedback have been instrumental in shaping the direction of this project, and I am truly grateful for his mentorship.

I would also like to extend my heartfelt thanks to Mrs. Mudra Vora, my external guide, for her expert guidance and support in this project. Her extensive experience and knowledge in the field of ecommerce have been immensely valuable in the development of this application.

I would also like to thank the faculty members of the Computer Science department for their support and encouragement throughout this project.

Parth Savaliya (D20DCS163)

## Table of Contents

<b>Chapter 1 Introduction.....</b>	<b>1</b>
1.1 Project Summary .....	2
1.2 Purpose .....	3
1.3 Objective .....	4
1.4 Technology Review.....	5
<b>Chapter 2 Project Management .....</b>	<b>7</b>
2.1 Project Planning .....	8
2.1.1 Project Development Approach and Justification .....	9
2.1.2 Project Effort and Time, Cost Estimation .....	10
<b>Chapter 3 System Requirements Study .....</b>	<b>11</b>
3.1 User Characteristics.....	12
3.2 Hardware and Software Requirements.....	13
<b>Chapter 4 System Analysis.....</b>	<b>14</b>
4.1 Comparison with Current System .....	15
4.2 Requirements of New System .....	16
4.2.1 Functional Requirements.....	16
4.2.2 Non-Functional Requirements.....	17
<b>Chapter 5 Implementation .....</b>	<b>18</b>
5.1 Key Terminologies .....	19
5.2 Tabula Overview and Implementation .....	21
5.2.1 Introduction .....	21
5.2.2 Working and Implementation.....	22
5.2.3 Results .....	25
5.3 PyMuPDF Overview and Implementation .....	26
5.3.1 Introduction .....	26
5.3.2 Working and Implementation.....	27
5.3.3 Results .....	29
5.4 PDFTron Overview and Implementation .....	30
5.4.1 Introduction .....	30
5.4.2 Working and Implementation.....	31

5.4.3 Results .....	34
5.5 Flask Overview and Implementation .....	35
5.5.1 Introduction .....	35
5.3.2 Working and Implementation.....	36
5.3.3 Results .....	37
5.5 Crucial Problems and Achievements .....	39
5.5.1 Need for New Formats Extraction.....	39
5.5.2 New Categories and Requirements .....	40
5.6 Version Control .....	42
<b>Chapter 6 UML Diagrams .....</b>	<b>43</b>
6.1 Use Case Diagram .....	44
6.2 Activity Diagram.....	45
6.2 Class Diagram .....	46
<b>Chapter 7 Conclusion .....</b>	<b>47</b>
<b>Chapter 8 Limitations and Future Enhancements .....</b>	<b>49</b>
8.1 Limitations .....	50
8.2 Future Enhancements .....	51
<b>Chapter 9 References.....</b>	<b>52</b>



## List of Figures

<b>Figure 5.1 ASIC Document .....</b>	<b>24</b>
<b>Figure 5.2 Extraction Result .....</b>	<b>25</b>
<b>Figure 5.3 PyMuPDF Annotations .....</b>	<b>28</b>
<b>Figure 5.4 Annotation Result .....</b>	<b>29</b>
<b>Figure 5.5 ASIC-CS without Modification .....</b>	<b>33</b>
<b>Figure 5.6 ASIC-CS with Modification.....</b>	<b>34</b>
<b>Figure 5.7 Flask Gui for Data Identification .....</b>	<b>37</b>
<b>Figure 5.8 Flask Gui for Selection of Directory .....</b>	<b>37</b>
<b>Figure 5.9 Flask Gui for Selected Directory .....</b>	<b>38</b>
<b>Figure 5.10 Flask Gui for Processed Files .....</b>	<b>38</b>
<b>Figure 6.1 Use Case Diagram.....</b>	<b>44</b>
<b>Figure 6.2 Activity Diagram.....</b>	<b>45</b>
<b>Figure 6.3 Class Diagram .....</b>	<b>46</b>

# **CHAPTER 1**

## **INTRODUCTION**

## 1.1 Project Summary

Auditing is often considered a tedious task due to the following reasons:

- Detailed scrutiny: Auditing involves a detailed scrutiny of financial statements, records, transactions, and internal controls. This process can be time-consuming and requires a lot of attention to detail.
- Complex regulations: Auditors are required to follow complex regulations and guidelines, which can make the audit process challenging and time-consuming.
- Large volume of data: Auditing involves the analysis of a large volume of data, including financial statements, invoices, receipts, and other documents. Sorting through this data can be a tedious and time-consuming task.
- Multiple parties involved: Auditing involves interactions with various parties, including management, stakeholders, and regulators. Coordinating with all these parties can be challenging and time-consuming.
- Continuous process: Auditing is a continuous process that requires regular updates and reviews. This means that auditors need to constantly monitor and update their findings, which can be a tedious and ongoing task.

Overall, auditing requires a lot of attention to detail, coordination, and compliance with regulations, making it a time-consuming and often tedious task.

Intelligent Document Extraction and Classification can play a significant role in solving the auditing problem by automating and streamlining the data extraction and classification process, making auditing more efficient and effective.

## 1.2 Purpose

The main purpose of the intelligent document classification and extraction module of SMSF Sorted software is to automatically identify, classify and extract relevant information from various types of documents related to self-managed super funds (SMSFs). This module uses artificial intelligence and machine learning algorithms to analyze and understand the content of documents such as financial statements, tax returns, and investment reports.

The module can identify and extract key data points such as fund names, member details, asset values, income, expenses, and more. It can also classify documents based on their type, such as financial statements, tax returns, or investment reports, making it easier for users to locate and manage their SMSF documents.

By automating the document classification and extraction process, SMSF Sorted software helps users save time and reduce the risk of errors associated with manual data entry. The module can also improve the accuracy and completeness of SMSF records, making it easier for users to meet compliance requirements and make informed financial decisions.

### 1.3 Objective

- The system should be able to extract data and metadata from pre-defined types of documents.
- The system should be able to classify the documents into predefined categories based on user-defined rules.
- The system should be able to learn from user feedback and improve its classification accuracy over time.
- The system should provide a user-friendly interface for users to manage and view extracted documents.
- The system should be able to handle a large volume of documents and process them in a timely and efficient manner.
- To enhance the user experience by making it easier for users to locate and manage their SMSF documents.
- To help users save time and reduce the risk of errors associated with manual data entry.

## 1.4 Technology Review

### 1. Python

- Python is a high-level, interpreted programming language that is simple and easy to learn, making it an ideal choice for beginners and experienced developers alike.
- Python has a vast library ecosystem, which includes thousands of open-source libraries and frameworks, enabling developers to build complex applications and systems with ease.
- Python is platform-independent, meaning that code written in Python can run on different operating systems, including Windows, Linux, and macOS, making it a flexible and versatile language for developing software.

### 2. MongoDB

- MongoDB's document-based data model allows for flexible and scalable data management, as data can be stored in a variety of formats without requiring predefined schemas. This makes it ideal for handling unstructured and semi-structured data, as well as for use in Agile software development environments.
- MongoDB is designed to be highly performant and scalable, allowing it to handle large volumes of data and high traffic loads. It is also designed to work efficiently with distributed systems, making it a good fit for cloud-based applications and microservices architectures.
- MongoDB comes with a rich set of features, including support for complex queries, full-text search, geospatial queries, and data aggregation. Additionally, it provides a range of tools and drivers for integration with popular programming languages, making it easy for developers to work with the database from their preferred development environment.

### 3. NLP

- NLP has numerous applications across various industries, including chatbots and virtual assistants for customer service, sentiment analysis for social media monitoring, speech recognition for voice assistants, machine translation for language localization, and text summarization for content curation, among others.
- NLP presents several complex challenges, such as ambiguity, context dependence, and natural language variability. To overcome these challenges, NLP techniques rely on machine learning algorithms, statistical models, and linguistic rules to extract meaning from language data.
- NLP is a multidisciplinary field that combines elements of computer science, linguistics, and cognitive psychology. NLP researchers and practitioners collaborate across these domains to develop and improve NLP techniques and applications, making it an exciting and dynamic field of study.

### 4. AWS S3

- AWS S3 is designed to provide virtually unlimited storage capacity, allowing users to store and retrieve large volumes of data from anywhere on the internet. S3 is also highly scalable, automatically adjusting to changing data needs and traffic loads.
- AWS S3 provides robust data protection and security features, including data encryption, access control, and compliance with industry standards such as HIPAA and GDPR. S3 also replicates data across multiple servers and data centers, ensuring high data durability and availability.
- AWS S3 integrates with a wide range of AWS services, including Amazon EC2, Amazon EMR, and Amazon RDS, as well as third-party tools and applications. Additionally, S3 provides APIs and SDKs for easy integration with popular programming languages such as Python, Java, and PHP, making it a flexible and versatile storage solution for developers.

## **CHAPTER 2**

# **PROJECT MANAGEMENT**



## 2.1 Project Planning

- We used Agile methodology, as it can be a good fit for developing an intelligent document extraction and classification module because it allows for iterative development, which can be particularly useful for a project with complex requirements and a high degree of uncertainty.
- The agile approach emphasizes collaboration between developers, stakeholders, and end-users, as well as continuous feedback and adaptation. This can be beneficial for a project like document extraction and classification, where the requirements may evolve over time as new types of documents are encountered or new use cases are identified.
- Using agile methodology can also help to mitigate risk by breaking down the project into smaller, more manageable pieces. This allows for regular testing and validation of the solution, which can help to identify and address issues early on, reducing the likelihood of delays or costly rework later in the development process.
- Overall, by adopting an agile approach, development teams can be more responsive to changing requirements, reduce risk, and deliver a higher quality solution that better meets the needs of stakeholders and end-users.

### 2.1.1 Project Development Approach and Justification

- **Flexibility:** The Agile approach allows for flexibility in project planning and development. This is important in an ML based solution, where the document types and can change rapidly. With Agile, the team can adapt to these changes quickly and adjust the project plan accordingly.
- **Iterative development:** Agile emphasizes iterative development, where small portions of the project are developed and tested in short cycles. This allows for frequent feedback and helps ensure that the project meets the customer's requirements and expectations.
- **Collaborative approach:** Agile promotes a collaborative approach, where the development team works closely with the customer and other stakeholders. This helps ensure that the project is developed with the customer's needs in mind and that everyone is aligned on the project goals.
- **Continuous improvement:** Agile encourages continuous improvement, with regular retrospectives to identify areas for improvement and make adjustments. This helps ensure that the project is always progressing towards the desired outcome and that any issues or roadblocks are addressed quickly.
- **Risk management:** Agile includes techniques for managing project risks, such as prioritizing features and developing a Minimum Viable Product (MVP) first. This helps mitigate the risks associated with developing a large-scale ML solution.

### 2.1.2 Project Effort and Time, Cost Estimation

- The COCOMO (Constructive Cost Model) is a widely used model for cost estimation in software development. It is a model that uses a set of equations based on project characteristics to estimate the effort and cost required to develop a software system.
- To estimate the cost and time required for ML based Document Extractor Engine development, we used the COCOMO II model, which is an updated version of the original COCOMO model.
- Based on our project's characteristics, we estimated that our software size was large, our development team had moderate experience, and our software complexity was high. Using these parameters, we estimated that the effort required for our project would be approximately 15 person-months.

- To estimate the cost and time required, we used the COCOMO II model's equation:

$$Efforts = a^{size} * b^{cost-drivers-sum}$$

- Where a and b are constants that depend on the development mode, and the product of all cost drivers is the multiplication of all factors that impact the project cost.
- Based on our project characteristics, we estimated the product of all cost drivers to be 2.1, and we assumed that our development mode was semi-detached.
- Using these values, we estimated the total cost of our project to be approximately \$15,000, with a development time of approximately 6-7 months.

## **CHAPTER 3**

# **SYSTEM REQUIREMENTS STUDY**

### 3.1 User Characteristics

The user classes can be defined based on the roles and responsibilities of the individuals who will interact with the module. Here are some potential user classes:

- **Auditor:** The primary user class for this module would be the auditor responsible for conducting the audit. The auditor would use the module to extract data and metadata from documents and classify them based on predefined categories, such as income, expenses, and investments.
- **Audit Manager:** The audit manager would use the module to oversee and manage the audit process. This could include setting up rules and templates for data extraction and classification, monitoring the progress of the audit, and reviewing and approving the work done by auditors.
- **IT Administrator:** The IT administrator would be responsible for managing and maintaining the software and its underlying infrastructure. This could include installing and configuring the software, ensuring data security and privacy, and providing technical support to users.
- **Compliance Officer:** The compliance officer would use the module to ensure that the audit is being conducted in compliance with regulatory requirements and internal policies. This could include reviewing the audit work done by auditors, verifying the accuracy and reliability of extracted data, and ensuring that the audit is completed within the specified timeframe.
- **Support Staff:** Support staff may use the module to provide administrative support to the audit team. This could include uploading documents, managing access controls, and providing general assistance to auditors.

### 3.2 Hardware and Software Requirements

Here are the software and hardware requirements for the NETRA Module of the IDEAC Engine:

- **Software Requirements:**
  - Python runtime environment
  - Java runtime environment
  - Required Packages
  - Postman as a development server
  - MongoDB database management system
  - Git version control system for source code management
  - PyCharm or any other code editor for development
  - RDC for server access
- **Hardware Requirements:**
  - A computer with a minimum of 8GB RAM
  - Sufficient disk space for development and deployment purposes.
  - A reliable internet connection for accessing cloud services and third-party APIs

Meeting these requirements will ensure that the Engine is developed, deployed, and run efficiently and effectively. Additionally, it is important to regularly review and update these requirements to ensure that the module remains compatible with the latest software and hardware technologies.

## **CHAPTER 4**

### **SYSTEM ANALYSIS**

## 4.1 Comparison with Current System

- Intelligent Document Extraction and Classification (IDEAC) software for Self-Managed Super Funds (SMSF) is designed to help automate the process of extracting and classifying data from various types of documents related to SMSF management, such as financial statements, bank statements, invoices, and receipts. This software can be compared to other SMSF software solutions such as BGL and Class, which also offer features for SMSF administration and compliance.
- One of the main advantages of this SMSF software is its ability to automate the process of extracting and classifying data from various types of documents. This can save a significant amount of time and reduce the risk of errors associated with manual data entry. Additionally, IDEAC-supported software can help improve accuracy and consistency in data entry, which can be especially important in the context of SMSF compliance.
- In contrast, traditional SMSF software solutions like BGL and Class typically rely on manual data entry and may not offer the same level of automation and accuracy as IDEAC-supported software. However, these solutions may offer other features that IDEAC-supported software does not, such as portfolio management tools, investment reporting.



## 4.2 Requirements of New System

### 4.2.1 Functional Requirements

- **Data Extraction:**

- The system should be able to extract data and metadata from different types of documents, including but not limited to PDF, Word, and Excel files.
- The system should be able to extract data from both structured and unstructured documents.
- The system should be able to handle different data formats, including text, tables, and images.
- The system should be able to extract relevant information, such as names, addresses, dates, and amounts.

- **Document Classification:**

- The system should be able to classify documents into predefined categories based.
- The system should be able to handle different types of categories, including but not limited to invoices, contracts, and reports.
- The system should be able to learn from user feedback and improve its classification accuracy over time.

- **User Management:**

- The system should provide a user-friendly interface for users to manage and view extracted documents.
- The system should allow users to search and filter extracted documents based on various criteria, such as document type, date, and content.
- The system should provide role-based access control to ensure the security and confidentiality of extracted documents.

#### **4.2.2 Non-Functional Requirements**

- **Performance Requirements**

- The system should be able to handle a large volume of documents and process them in a timely and efficient manner.
- The system should be able to handle multiple document processing requests concurrently.
- The system should have a response time of less than 5 seconds for document processing requests.

- **Accuracy**

- The system should have a high accuracy rate for data extraction and document classification, with an error rate of less than 1%.

- **Security Requirements**

- The system should be designed with security in mind, with appropriate measures to ensure the confidentiality and integrity of extracted documents.
- The system should comply with relevant data privacy laws and regulations.

## **CHAPTER 5**

# **IMPLEMENTATION**

## 5.1 Key Terminologies

This terminologies listed here are specific to TATHYA part of the IDEAC module.

- **Document Extraction:**

The process of automatically extraction data from documents on the bases of predefined function, such as account details, income, expenses, or investments, using Tabula-Python.

- **Unstructured Data:**

Data that does not have a predefined structure, making it difficult to analyze using traditional methods. Examples include text documents, images, and audio files.

- **Structured Data:**

Data that is organized and formatted in a specific way, making it easy to analyze using traditional methods. Examples include data stored in spreadsheets or databases.

- **Metadata:**

Data that provides information about other data, such as the author, date, and file format of a document.

- **Data Verification:**

The process of validating extracted data against the source documents to ensure accuracy and reliability.

- **Rule-based System:**

A system that uses predefined rules or algorithms to analyze data and make decisions.

- **Accuracy:**

The degree to which extracted or classified data matches the actual data present in the source document.

- **Precision:**

The degree to which extracted or classified data is relevant to the task at hand.

- **Recall:**

The degree to which all relevant data is extracted or classified from the source document.

## 5.2 Tabula Overview and Implementation

### 5.2.1. Introduction

**Tabula-Python** is a Python library that can be used to extract tables from PDF documents. Here are some key points about Tabula-Python:

- Tabula-Python is built on top of the Tabula Java library, which is a tool for extracting tables from PDF documents.
- With the help of Tabula-Python, interacting with Tabula from within Python is made straightforward, making it simple to extract tables and incorporate them into your Python programmes.
- With a liberal MIT licence, Tabula-Python is open-source and cost nothing to use.
- The table extraction process can be fine-tuned using a variety of variables supported by Tabula-Python, including defining the area of the PDF document to extract tables from and modifying the table recognition method.
- Applications for Tabula-Python include data analysis, document processing, and other tasks.
- The ongoing maintenance and updating of Tabula-Python results in the regular addition of new features and issue fixes.
- Since Tabula-Python is designed to be speedy and effective, it can readily manage even very large and complex PDF documents.

In summary, Tabula-Python is a powerful and flexible Python library for extracting tables from PDF documents, with a range of features and options for fine-tuning the table extraction process. With its simple interface and comprehensive documentation, Tabula-Python is an excellent choice for developers looking to extract tables from PDF documents with Python.

### 5.2.2. Working and Implementation

Tabula-Python works by using the Tabula Java library to extract tables from PDF documents, and then providing a Python interface for working with the extracted tables. Here's an overview of how Tabula-Python works:

1. First, the user specifies the PDF document they want to extract tables from, either by providing a file path or a URL.
2. Next, the user specifies the area of the PDF document they want to extract tables from, either by specifying the page number or by providing a rectangle that defines the area.
3. Tabula-Python then uses the Tabula Java library to extract tables from the specified area of the PDF document.
4. The extracted tables are returned to the Python script as a Pandas DataFrame object, which can be easily manipulated and analyzed using Python.
5. Finally, the user can save the extracted tables to a CSV file or other format, or integrate them into their Python application for further processing.

Tabula-Python provides a range of options for fine-tuning the table extraction process, including adjusting the table recognition algorithm, specifying the column and row separators, and more. These options can be specified using a set of parameters passed to the Tabula-Python function that performs the table extraction.

For extracting the data like date or specific pattern or for find we used re it is a python library for working with regular expressions:

- re provides a powerful set of tools for working with regular expressions, including pattern matching, substitution, and more.
- re is built into Python, so it's available out of the box with no additional installation required.
- re provides a simple and intuitive interface for working with regular expression in Python, with functions like match(), search(), findall(), and sub().

- re can be used in a variety of applications, including text processing, data validation, and more.
- re is designed to be fast and efficient, and can handle even large, complex regular expressions quickly and easily.

So we can say that re is a powerful and flexible Python library for working with regular expressions, with a range of features and tools for pattern matching, substitution, and more. With its simple interface and comprehensive documentation, re is an excellent choice for developers looking to work with regular expressions in Python.

```
def fetch_acn(self):
    acn = ""
    try:
        # read pdf through tabula
        df = read_pdf(self.file, pages=self.pages_to_use, multiple_tables=True,
                      area=(0, 0, 1200, 900), columns=[0, 470], pandas_options={"header": None})

        # find the pattern for ACN
        for row_index in range(df[0].shape[0]):
            if re.findall("ACN(.*)", str(df[0].iloc[row_index, 1])):
                acn = re.findall("ACN(.*)", df[0].iloc[row_index, 1])[0]
                # acn = acn.replace(" ", "")
                break

        # remove extra white spaces
        acn = acn.lstrip(" ")

        # string for coordinates whichever is found
        acn_str_for_coordinate = "ACN"

        # create list of acn value and word before that value
        acn_list = [[acn_str_for_coordinate, acn]]


        # create dataframe of that list to pass to coordinate extractor
        acn_df = pd.DataFrame(acn_list)

        # call coordinate extractor passing file_location and df
        coordinate_support_str = fetch_supporting_extractor(self.file, self.coordinate_list, acn_df)

    return 1, coordinate_support_str
except:
    return 0, None
```



3/28/2019
View company details


**ASIC**  
Australian Securities & Investments Commission

**Forms Manager**  
Registered Agents

---

**Company:** GOLDWISHER NOMINEES PTY LIMITED ACN 999 004 689

---

**Company details**

Date company registered 01-04-2014  
Company next review date 01-04-2019  
Company type Australian Proprietary Company  
Company status Registered  
Home unit company No  
Superannuation trustee company Yes  
Non profit company No

---

**Registered office**

15 THE ALPINE ROAD VARROVILLE NSW 2100

---

**Principal place of business**

15 THE ALPINE ROAD VARROVILLE NSW 2100

---

**Officeholders**

TOM, JOHN  
Born 09-07-1949 at DHOLKA INDIA  
15 THE ALPINE ROAD VARROVILLE NSW 2100  
Office(s) held: Director, appointed 01-04-2014  
Secretary, appointed 23-05-2018

---

**Company share structure**

Share class	Share description	Number issued	Total amount paid	Total amount unpaid
ORD	ORDINARY SHARE	12	12.00	0.00

---

**Members**

TOM , JOHN
15 THE ALPINE ROAD VARROVILLE NSW 2100

Share class	Total number held	Fully paid	Beneficially held
ORD	12	Yes	Yes

---

**Document history**

These are the documents most recently received by ASIC from this organisation.

Received	Number	Form Description	Status
23-05-2018	0EAG657281 484	CHANGE TO COMPANY DETAILS	Processed and imaged
01-04-2014	2E0210862 201	APPLICATION FOR INCORPORATION (DIVN 1)	Processed and imaged

[ASIC Home](#) | [Privacy Statement](#) | [Conditions of use](#) | [Feedback](#)  
Copyright 2003 Australian Securities & Investments Commission.

<https://www.edge.asic.gov.au/001/regportal?update/requestViewCompany/s=1e91b3b707a3131110b236026f5958ea4cf8>
1/1

Figure 5.1 ASIC Document

### 5.2.3. Results

```
"ASICData": [  
  {  
    "acn": "999004689",  
    "company_name": "GOLDWISHER NOMINEES PTY LIMITED",  
    "date": "28/03/2019",  
    "directors": [  
      {  
        "family_name": "TOM",  
        "first_name": "JOHN",  
        "second_name": "",  
        "address": "15 THE ALPINE ROAD VARROVILLE NSW 2100",  
        "date_of_birth": "09/07/1949",  
        "date_of_appointment": "01/04/2014"  
      }  
    ]  
  }  
]
```

Figure 5.2 Extraction Result

## 5.3 PyMuPDF Overview and Implementation

### 5.3.1. Introduction

**PyMuPDF** is a Python library for working with PDF documents. Here are some key points about PyMuPDF:

- PyMuPDF is built on top of the MuPDF library, which is a lightweight, high-performance PDF rendering engine.
- For working with PDF files, PyMuPDF offers a complete range of capabilities, including document creation and parsing, text extraction, page manipulation, annotation, and more.
- Numerous operating systems, including Windows, Linux, and macOS, are supported by PyMuPDF.
- Several programming languages, including Python 2 and 3, can be utilised with PyMuPDF.
- PyMuPDF has a permissive MIT licence and is open-source and cost-free to use.
- PyMuPDF is made to be swift and effective; it can easily handle even very huge and complex PDF documents.
- To assist developers in getting started quickly and simply, PyMuPDF offers comprehensive documentation and examples.

In conclusion, PyMuPDF is a strong and flexible Python library for working with PDF files. It has a variety of features and tools for managing even complex PDF documents. Developers wishing to create Python-based PDF-related apps have a lot of options, and PyMuPDF is a great one because to its lightweight architecture and thorough documentation.

### 5.3.2. Working and Implementation

PyMuPDF is a Python library for working with PDF documents. It is built on top of the MuPDF library, which is a high-performance PDF rendering engine. Here are the key steps involved in working with PyMuPDF:

1. Opening a PDF document: The first step in working with PyMuPDF is to open a PDF document. This can be done using the `open()` function, which takes the path to the PDF file as an argument.
2. Accessing the document pages: PyMuPDF allows you to access individual pages of a PDF document using the `getPage()` function. You can then work with each page as a separate object, including rendering it, extracting text or images, and more.
3. Manipulating the document pages: PyMuPDF provides a range of tools for manipulating PDF pages, including rotating, cropping, and resizing. You can also add or remove pages from the document, and merge or split PDF files.
4. Working with annotations: PyMuPDF supports a range of annotation types, including text boxes, highlighters, and stamps. You can add annotations to PDF documents, edit existing annotations, and extract annotations for further processing.
5. Extracting text and images: PyMuPDF provides tools for extracting text and images from PDF documents, including options for OCR (optical character recognition) and image enhancement.
6. Saving the modified document: Once you have made changes to a PDF document, you can save the modified document using the `save()` function. This will overwrite the original file with the modified version.

PyMuPDF is a robust and adaptable Python library for working with PDF files, to sum up. PyMuPDF is a great option for programmers wishing to create Python-based applications that interact with PDF files thanks to its wide selection of tools and functions.

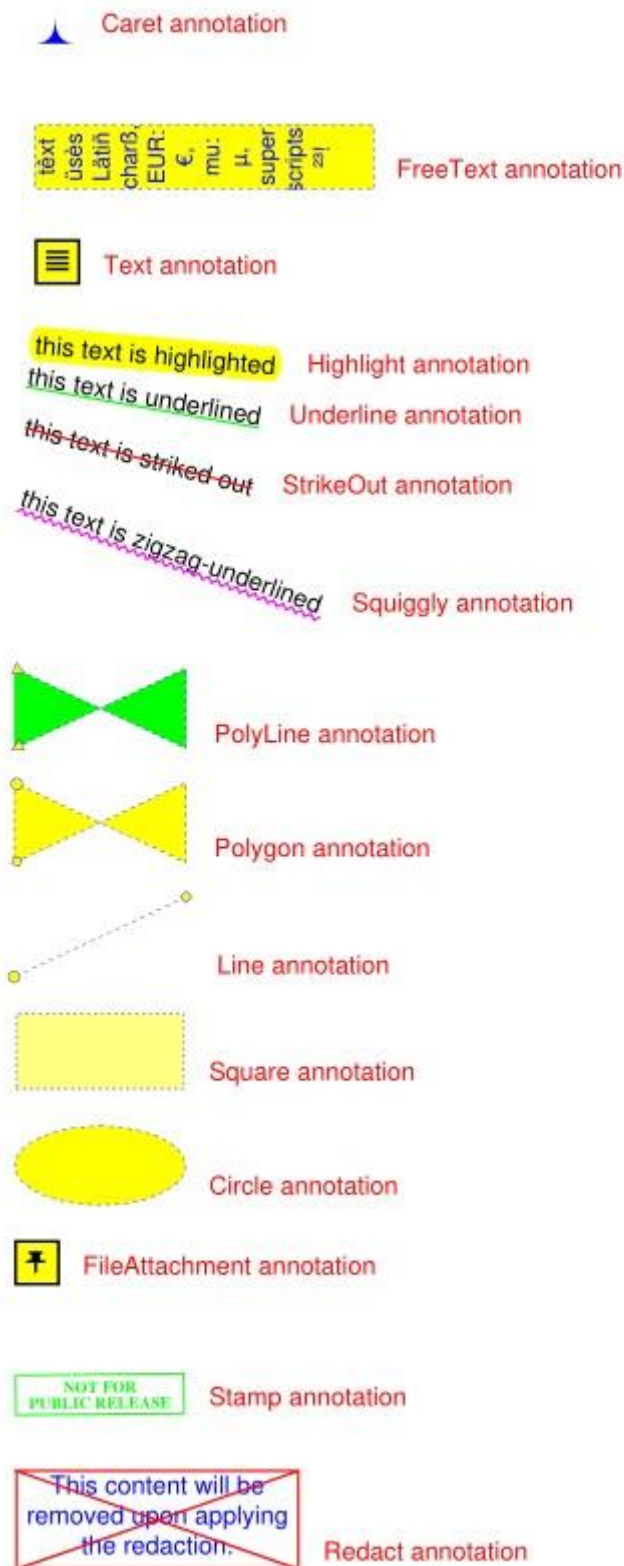



Figure 5.3 PyMuPDF Annotation

### 5.3.3. Results

```
return_list = {list: 1} [{"1", '999 004 689', (348.9540100097656, 107.00947570800781, 409.0313415527344, 119.0806884765625)}]
```

3/28/2019 [View company details](#)



**ASIC**  
 Australian Securities & Investments Commission

**Forms Manager**  
 Registered Agents

---

Company: **GOLDWISHER NOMINEES PTY LIMITED** ACN **999 004 689**

---

#### Company details

Date company registered 01-04-2014  
 Company next review date 01-04-2019  
 Company type Australian Proprietary Company  
 Company status Registered  
 Home unit company No  
 Superannuation trustee Yes  
 company  
 Non profit company No

---

#### Registered office

15 THE ALPINE ROAD VARROVILLE NSW 2100

---

#### Principal place of business

15 THE ALPINE ROAD VARROVILLE NSW 2100

---

#### Officeholders

TOM, JOHN  
 Born 09-07-1949 at DHOLKA INDIA  
 15 THE ALPINE ROAD VARROVILLE NSW 2100  
 Office(s) held: Director, appointed 01-04-2014  
 Secretary, appointed 23-05-2018

---

#### Company share structure

Share class	Share description	Number issued	Total amount paid	Total amount unpaid
ORD	ORDINARY SHARE	12	12.00	0.00

---

#### Members

TOM , JOHN 15 THE ALPINE ROAD VARROVILLE NSW 2100

Share class	Total number held	Fully paid	Beneficially held
ORD	12	Yes	Yes

---

#### Document history

These are the documents most recently received by ASIC from this organisation.

Received	Number	Form Description	Status
23-05-2018	0EAG657281 484	CHANGE TO COMPANY DETAILS	Processed and imaged
01-04-2014	2E0210862 201	APPLICATION FOR INCORPORATION (DIVN 1)	Processed and imaged

[ASIC Home](#) | [Privacy Statement](#) | [Conditions of use](#) | [Feedback](#)  
 Copyright 2003 Australian Securities & Investments Commission.

<https://www.edge.asic.gov.au/001/regaportal?update/requestViewCompany/s=1e91b3b707a3131110b236026f5958ea4cf8>

1/1

Figure 5.4 Annotation Result

## 5.4 PDFTron Overview and Implementation

### 5.4.1. Introduction

**PDFTron** is a powerful and versatile PDF SDK (software development kit) that provides a comprehensive set of tools for working with PDF documents. Here are some key points about PDFTron:

- PDFTron can be used to read, write, and edit PDF documents programmatically. It supports a wide range of platforms and programming languages, including .NET, Java, C++, Python, and more.
- Form filling, digital signatures, redaction, and other functions are among the many tools that PDFTron offers for working with PDF documents, including document viewing and annotation, PDF production and conversion, and more.
- Due to its efficiency and scalability-focused architecture, PDFTron can rapidly and effectively handle even very big and complicated PDF documents.
- To make it simple and quick for developers to get started, PDFTron offers a wealth of documentation, code samples, and tutorials.
- Many businesses and organisations, including Fortune 500 enterprises, governmental bodies, and startups, rely on PDFTron.
- In order to maintain compatibility with the newest software and hardware platforms, PDFTron is dedicated to supporting the most recent PDF standards and technologies, such as PDF 2.0 and PDF/A.
- To fulfil the demands of various users, from small businesses to major corporations, PDFTron provides a variety of licencing alternatives.

In conclusion, PDFTron is a complete PDF SDK that offers a robust set of tools and functionality for working with PDF documents, thorough documentation, and support for a multitude of platforms and programming languages. Building a straightforward PDF viewer or a sophisticated enterprise application? PDFTron can help you do the task quickly and effectively.

### 5.4.2. Working and Implementation

A cross-platform PDF SDK called PDFTron enables programmers to include PDF features into their programmes. Developers can include PDF capabilities into their Python programmes with the help of PDFTron for Python, which offers a Python interface to the PDFTron SDK.

1. Import the PDFTron library into your Python project.
2. Use the PDFTron API to load, parse, edit, and save PDF documents as well as perform other activities on them.
3. To edit PDF document elements like text, photos, annotations, and more, use the PDFTron API.
4. To convert PDF files to other formats like HTML, SVG, or XPS, use the PDFTron API.
5. To conduct complex actions on PDF documents, such as encryption, digital signatures, and redaction, use the PDFTron API.
6. For quicker loading and smaller file sizes, optimise and compress PDF documents using the PDFTron API.
7. Use the PDFTron API to extract text, tables, and form fields from PDF documents.
8. Use the PDFTron API to compare PDF files and look for changes and discrepancies.

For working with PDF files in Python, PDFTron offers a complete collection of utilities. Developers wishing to create Python-based PDF-related applications have a tonne of options, but PDFTron stands out thanks to its robust API and feature set.



```
def annotate_using_pdf_tron(file_location, dimension_tuple, page_number):  
    # open the file in doc for PyMuPDF  
    doc = fitz.open(file_location)  
  
    # select the page from the doc, which is passed in the function  
    page = doc[page_number]  
  
    # get page_height from above page object  
    page_height = page.rect.height  
  
    # create variables (x0, y0, x1, y1), for highlighting text  
    # here as PDFTron and PyMuPDF uses different coordinate system, we need to  
    # subtract page_height from given y-axis dimensions for y0 and y1  
    x0 = dimension_tuple[0]  
    y0 = page_height - dimension_tuple[1]  
    x1 = dimension_tuple[2]  
    y1 = page_height - dimension_tuple[3]  
  
    # now open the file in PDFTron object  
    doc = PDFDoc(file_location)  
  
    # as we have using PDF pages now, we need to add 1 in page_number because  
    # page_number has started with 0  
    page = doc.GetPage(page_number + 1)  
  
    # create rectangle to highlight at given coordinate  
    sq = Square.Create(doc.GetSDFDoc(), Rect(x0, y0, x1, y1))  
  
    # set padding to the rectangle  
    sq.SetPadding(-2)  
  
    # generates the appearance stream for the annotation  
    sq.RefreshAppearance()  
  
    # push the rectangle in the PDF page  
    page.AnnotPushBack(sq)  
  
    # save the document after highlighting  
    doc.Save(file_location, SDFDoc.e_linearized)
```

Inquires 1300 300 630

Issue date 20 Jul 20

---

## Company Statement

Extract of particulars - s346A(1) Corporations Act 2001


CORPORATE KEY: 95162925

**Check this statement carefully**


You are legally obligated to ensure that all your company details listed on this company statement are complete and correct. This is required under s346C (1) and/or s346B and s346C (2) of the *Corporations Act 2001*.

You must check this statement carefully and inform ASIC of any changes or corrections immediately. **Do not return this statement.** You must notify ASIC within 28 days after the date of change, and within 28 days after the date of issue of your annual company statement. Late lodgement of changes will result in late fees. These requirements do not apply to the **Additional company information**.


**You must notify ASIC of any changes to company details — Do not return this statement**



- To make changes to company details or amend incorrect information
- go to [www.asic.gov.au/changes](http://www.asic.gov.au/changes)
- log in to our online services and make the required updates
- first time users will need to use the corporate key provided on this company statement



Phone if you've already notified ASIC of changes but they are not shown correctly in this statement.  
Ph: 1300 300 630



Use your agent.

ACN 145 285 000

FOR ABC SCHOOL PTY. LTD.

REVIEW DATE: 19 July 20

Figure 5.5 ASIC-CS without Modification

### 5.4.3. Results

Inquires 1300 300 630  
 Issue date 20 Jul 20

---

## Company Statement

Extract of particulars - s346A(1) Corporations Act 2001

CORPORATE KEY: 95162925

**Check this statement carefully**

You are legally obligated to ensure that all your company details listed on this company statement are complete and correct. This is required under s346C (1) and/or s346B and s346C (2) of the Corporations Act 2001.

**You must check this statement carefully and inform ASIC of any changes or corrections immediately. Do not return this statement.** You must notify ASIC within 28 days after the date of change, and within 28 days after the date of issue of your annual company statement. Late lodgement of changes will result in late fees. These requirements do not apply to the Additional company information.

**You must notify ASIC of any changes to company details — Do not return this statement**

To make changes to company details or amend incorrect information

- go to [www.asic.gov.au/changes](http://www.asic.gov.au/changes)
- log in to our online services and make the required updates
- first time users will need to use the corporate key provided on this company statement

Phone if you've already notified ASIC of changes but they are not shown correctly in this statement.

Ph: 1300 300 630

ACN 145 285 000  
FOR ABC SCHOOL PTY. LTD.

---

REVIEW DATE: 19 July 20

---

## Company Statement

These are the current company details held by ASIC. You must check this statement carefully and inform ASIC of any changes or corrections immediately. Late fees apply. **Do not return this statement.**

**1 Registered office**  
SUITE 3 LEVEL 6 68-80 GEORGE STREET PARRAMATTA NSW 2150

---

**2 Principal place of business**  
UNIT 2 105 WOODLAND STREET S BALGOWLAH NSW 2093

---

**3 Officeholders**

Name:	AB DE VILLIERS
Born:	SYDNEY NSW
Date of birth:	23/10/1957
Address:	UNIT 2 105 WOODLAND STREET S BALGOWLAH NSW 2093
Office(s) held:	DIRECTOR, APPOINTED 19/07/2010

---

**4 Company share structure**

Share class	Shares description	Number issued	Total amount paid on these shares	Total amount unpaid on these shares
ORD	ORDINARY	1	\$1.00	\$0.00

---

**5 Members**

These details continue on the next page

ABC SCHOOL PTY. LTD. ACN 145 285 000 Page 1 of 2

Figure 5.6 ASIC-CS with Modification

## 5.5 Flask Overview and Implementation

### 5.5.1 Introduction

Flask is a lightweight and flexible Python web framework for building web applications. Here are some key points about Flask:

- Flask is designed to be simple, lightweight, and easy to use, with a minimalistic core and optional extensions for added functionality.
- Flask is compatible with a variety of web servers and hosting systems since it employs the WSGI (Web Server Gateway Interface) standard to communicate with web servers.
- Flask offers a straightforward and understandable API with capabilities like routing, templates, request and response handling, and more for developing online apps.
- For additional functionality, Flask offers a broad variety of third-party extensions, including those for database integration, authentication, and security.
- To assist developers in getting started quickly and simply, Flask offers copious documentation and examples.
- Flask may be used to create a broad variety of web applications, from straightforward static websites to intricate web applications. It is very adaptable.

Briefly put, Flask is a robust and adaptable Python web framework for creating online applications. It has a simple core and a variety of configurable extensions for further functionality. Flask is a superb option for programmers wishing to create web apps using Python because of its clear and understandable API, comprehensive documentation, and vibrant development community.

### 5.5.2 Working and Implementation

A Python web framework called Flask offers a selection of resources and packages for creating web applications. A summary of how Flask functions is given below:

1. Import the Flask library into your Python project.
2. Define a Flask application object by creating an instance of the Flask class.
3. Define routes and views using the application object. A route is a URL path that is mapped to a function that generates a response for that path.
4. Define templates and static files using the application object.  
Templates are used to generate dynamic HTML content, while static files like images, style sheets, and JavaScript files are served directly to the client.
5. Use Flask's built-in request and response handling functions to interact with incoming requests and generate responses.
6. Use Flask's built-in error handling functions to handle errors and exceptions in the application.
7. Use Flask's built-in session management functions to manage user sessions and authentication.
8. Use Flask's built-in testing functions to write unit tests for the application.
9. Deploy the Flask application to a web server or hosting platform, such as AWS, or Google Cloud Platform.

For creating web applications in Python, Flask offers a suite of tools and libraries that include routing, views, templates, static files, database integration, request and response handling, error handling, session management, and testing. For programmers wishing to create web apps with Python, Flask is a fantastic option thanks to its clear and straightforward API and comprehensive documentation.

### 5.5.3 Results

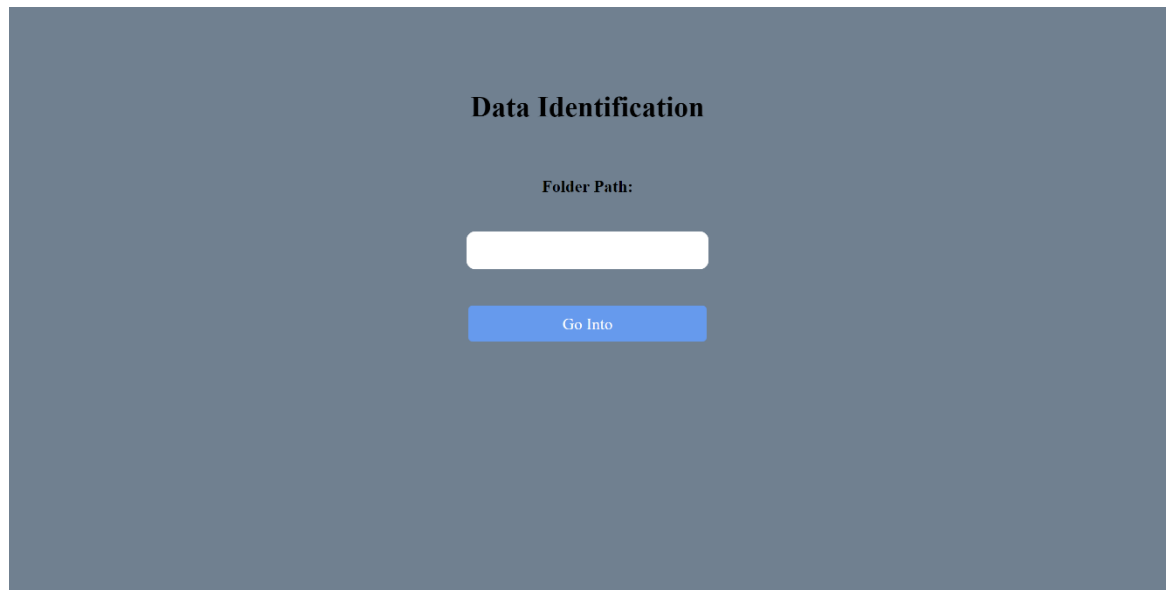


Figure 5.7 Flask Gui for Data Identification

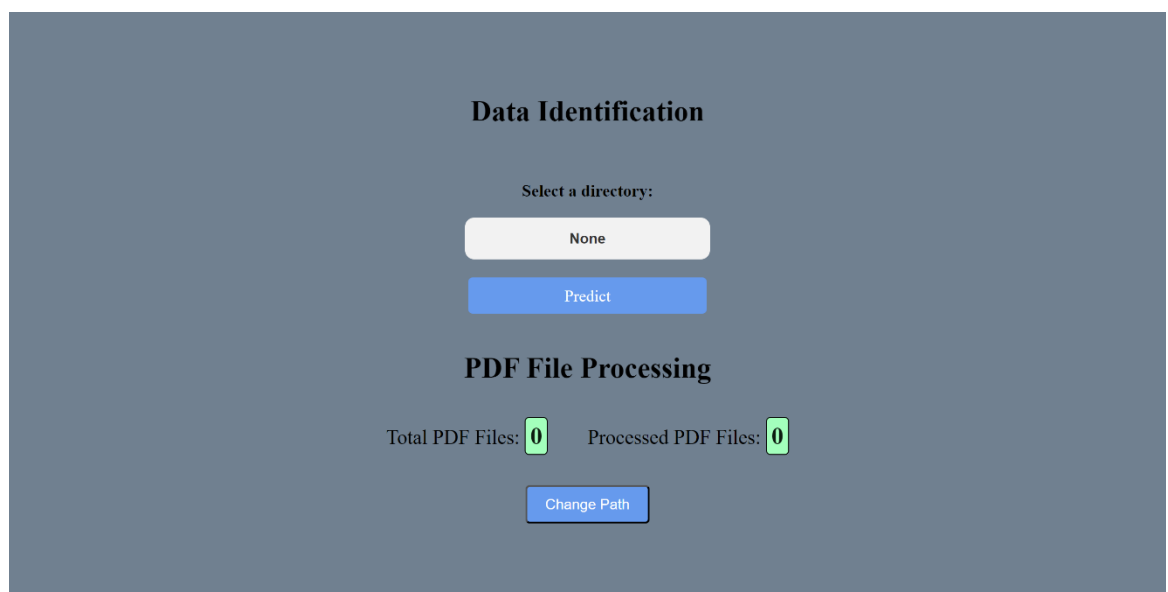


Figure 5.8 Flask Gui for Selection of Directory

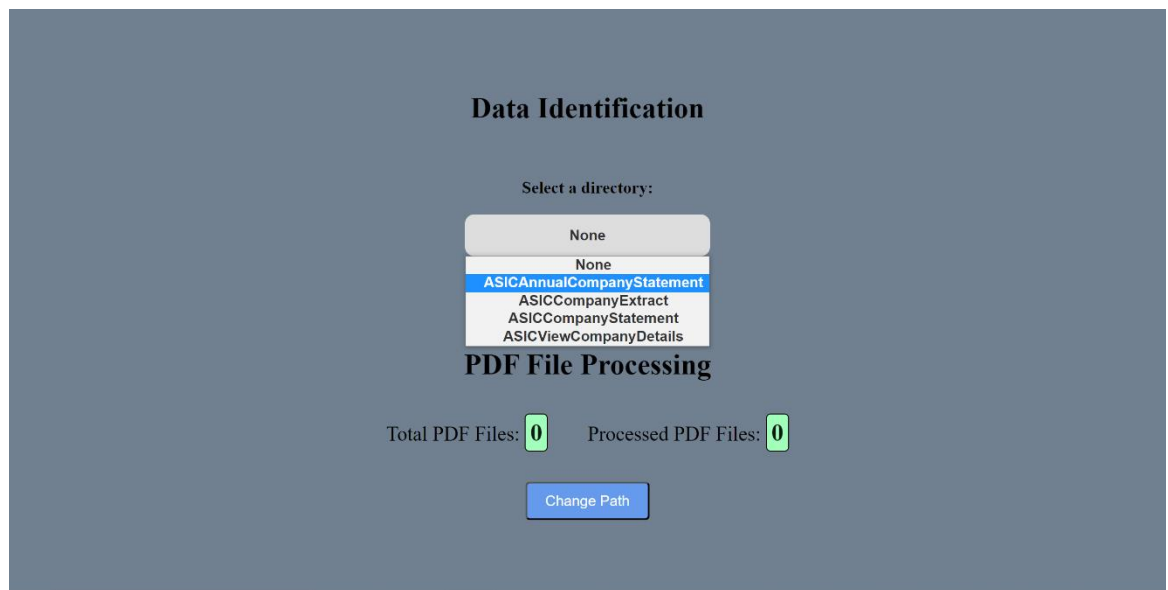


Figure 5.9 Flask Gui for Selected Directory

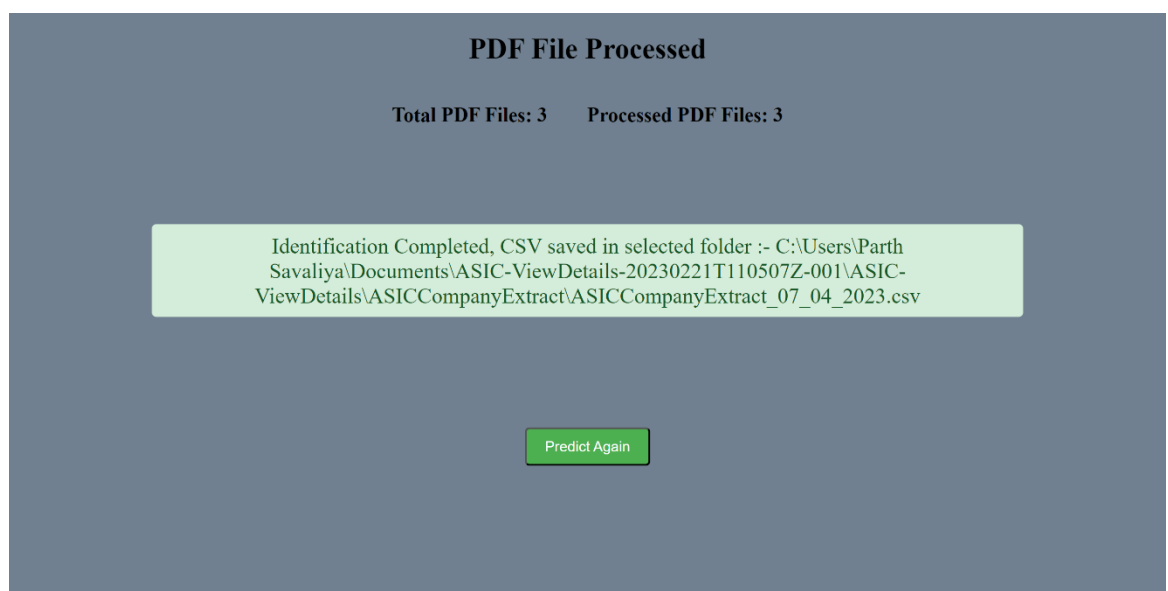
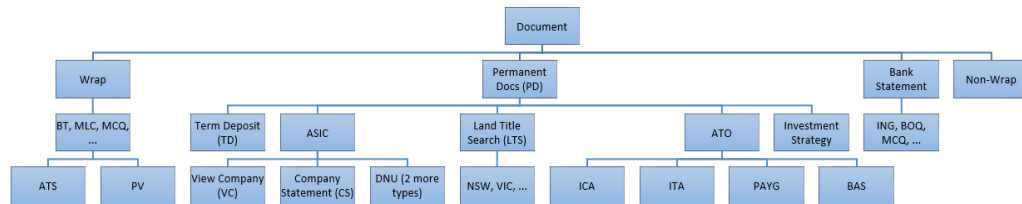


Figure 5.10 Flask Gui for Data Identification

## 5.6 Crucial Problems and Achievements

### 5.5.1. Need for New Formats Extraction



The hierarchy above depicts the 3 layer classification pipeline that was into production till mid-2022 and the extraction was in limit because of the sub classification module. It was classifying the documents in limit but now the update classification module support more then 100+ sub format categories so now the extraction turn comes to support that formats in the extraction engine. The extraction should be completely accurate for the specific format for example ASIC has 4 formats like ASIC-CS, ASIC-VS, ASIC-CE, and ASIC-ACS so at this point we are supporting 2 formats of ASIC. We need to analysis the single format multiple documents for start extraction working on. In that multiple conditions of particular value to extract. Additionally, new formats support we create the documentation of the Sub Classified documents format for extraction values form the accurate place. And for user friendly expression we now started working on the coordinates so that user get to know from which page of document and in page from where the data is extracted. Overall, supporting old working of extraction formats with update data, and also trying to implement new format extraction which needed to meet the demands of the customer.



### 5.5.2. New Categories and Requirements

In the new version of the TATHYA module we were assigned the requirements that were needed to be fulfilled.

Sr. No.	Netra V1 (Main Classifier)		Netra V2 (Sub Classifiers)		Tathya V1
	Main Document Type	Sub Type Count	SubType	Sub-Sub Type	Extraction Ready?
1	Wrap	10-15	MLC Wrap	ATS	Yes
				DPV	Yes
			BT Panorama	ATS	Yes
				DPV	Yes
			Macquarie Wrap	ATS	Yes
				DPV	Yes
			Asgard	ATS	Yes
				DPV	Yes
			Net Wealth Wrap	ATS	Yes
				DPV	Yes
			Hub24	ATS	Yes
				DPV	Yes
			Credit Suisse	ATS	Yes
				DPV	Yes
			North	ATS	Yes
				DPV	Yes
			Patersons	ATS	Yes
				DPV	Yes
2	ATO	4+	Lee Clarke	ATS	Yes
				DPV	Yes
			Morgan Stanly		Pending
			CommSec		
			Westpac		
			JB Were		
			ORD Minnett		
			PAYG		Yes
			ICA		Yes
			BAS		Yes
			ITA		Yes
			Some more sub types	Yet to Name them	

Sr. No.	Netra V1 (Main Classifier)		Netra V2 (Sub Classifiers)		Tathya V1
	Main Document Type	Sub Type Count	SubType	Sub-Sub Type	Extraction Ready?
3	ASIC Statement	4	Annual Company Statement		
			Company statement		Yes
			Company Extract		
			View company details		Yes
4	Bank Statement	15 Banks (Extraction of 25 Formats)	Macquarie	2	Yes
			NAB	2	Yes
			ANZ	2	Yes
			CBA	2	Yes
			Westpac	1	Yes
			St. George	4	Yes
			Rabo bank	1	Yes
			ING Bank	2	Yes
			Bank SA	2	Yes
			BOQ	2	Yes
			Suncorp	2	Yes
			Adelaide Bank	-	Yes
			AMP North & AMP	1	
			BankWest	2	Yes
5	Land Title Search	7	People's Choice Credit Union	-	
			NSW		Yes
			QLD		Yes
			SA		Yes
			VIC		Yes
			WA		Yes
			ACT		Yes
6	Share Registry	3	Tasmania		Yes
			Linked Market		Yes
			Computer Share		Yes
			Boardroom		

Sr. No.	Netra V1 (Main Classifier)		Netra V2 (Sub Classifiers)		Tathya V1
	Main Document Type	Sub Type Count	SubType	Sub-Sub Type	Extraction Ready?
7	Actuarial Certificate				
8	ATO Trustee Declaration				
9	Lease Agreement				
10	Loan Agreement				
11	Rollover Statement				
12	Term Deposit	15+	Similar to Bank Statement		
13	Loan Statement				
14	Trust Deed				
15	Bare Trust Deed				
16	Death Benefit Nomination				
17	Expense	7	Accountancy Fees		
			ASIC Invoice		
			Audit Fees		
			Depreciation		
			Member Insurance		
			Property Expense	10+	
18	Work Test Declaration			Council Rates	
19	Investment Strategy			Electricity Expenses	
20	Notice of Intent to Claim				
21	Property	4	Property Appraisal		
			Property Buy and Sale		
			Property Settlement Statement		
			Property Valuation		
22	Rental Statement				

There were 22 main categories to be supported along with 65+ sub categories, 100+ sub format categories and 50+ sub format extraction.

## 5.7 Version Control

Bitbucket is a popular version control system that enables developers to collaborate on projects, track changes, and manage code. For our project we used Bitbucket as our primary version control system. We created two branches named "Identification" and "MitraNewIdentification" to manage our codebase effectively from the origin branch.

This origin branch was where we made changes that were ready for production. We ensured that our code was thoroughly tested and reviewed before merging into the master branch. This helped us to maintain a stable and reliable version of our code.

The “identification” branch was where we made changes that were still in development. This branch was created to allow us to experiment with new features and functionalities without affecting the stable version of our code in the master branch. We worked on this branch until we were satisfied with the changes made, then we moved on to "MitraNewIdentification" branch where we integrated the new NETRA module with the TATHYA which later will be merged to master branch which is production ready.

## **CHAPTER 6**

### **UML Diagrams**

## 6.1 Use Case Diagram

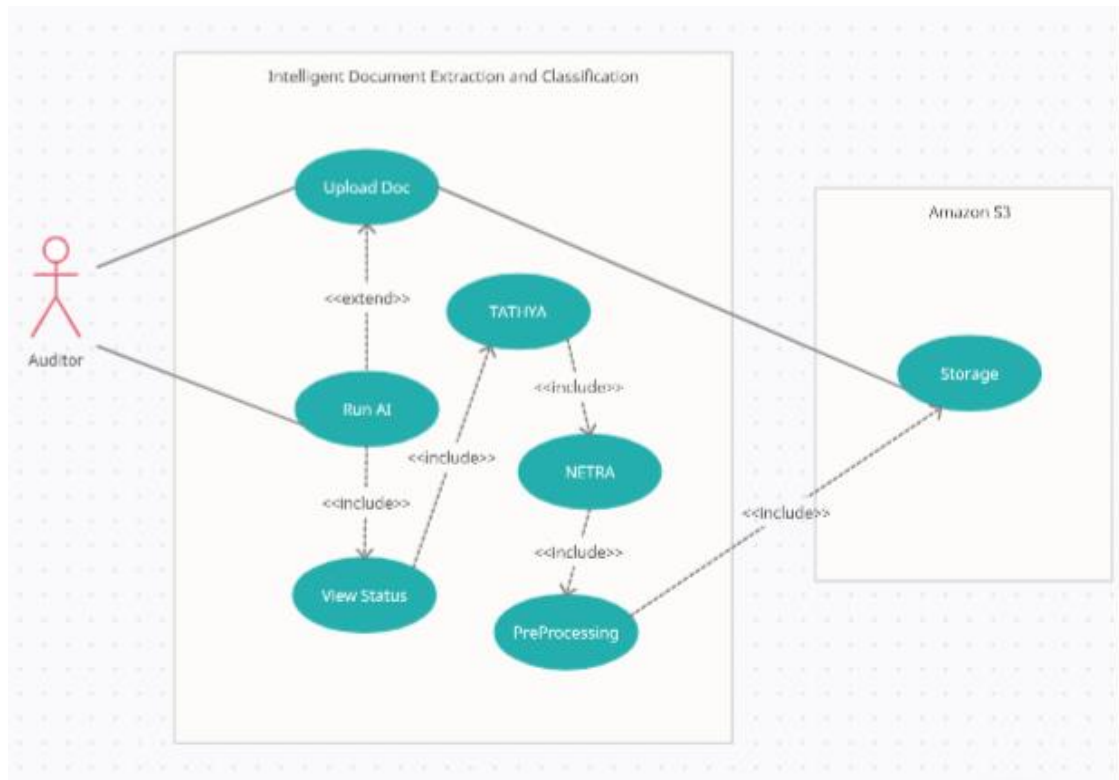


Figure 6.1 Use Case Diagram

## 6.2 Activity Diagram

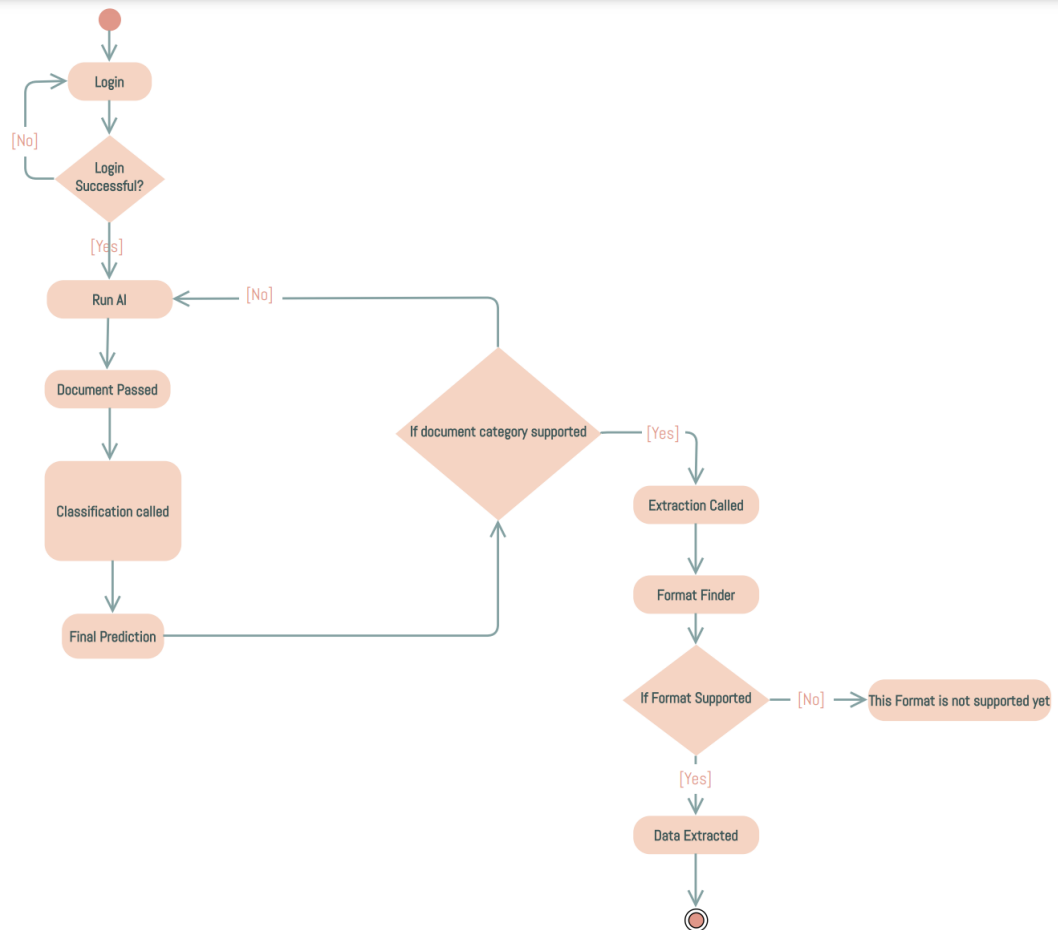


Figure 6.2 Activity Diagram

### 6.3 Class Diagram

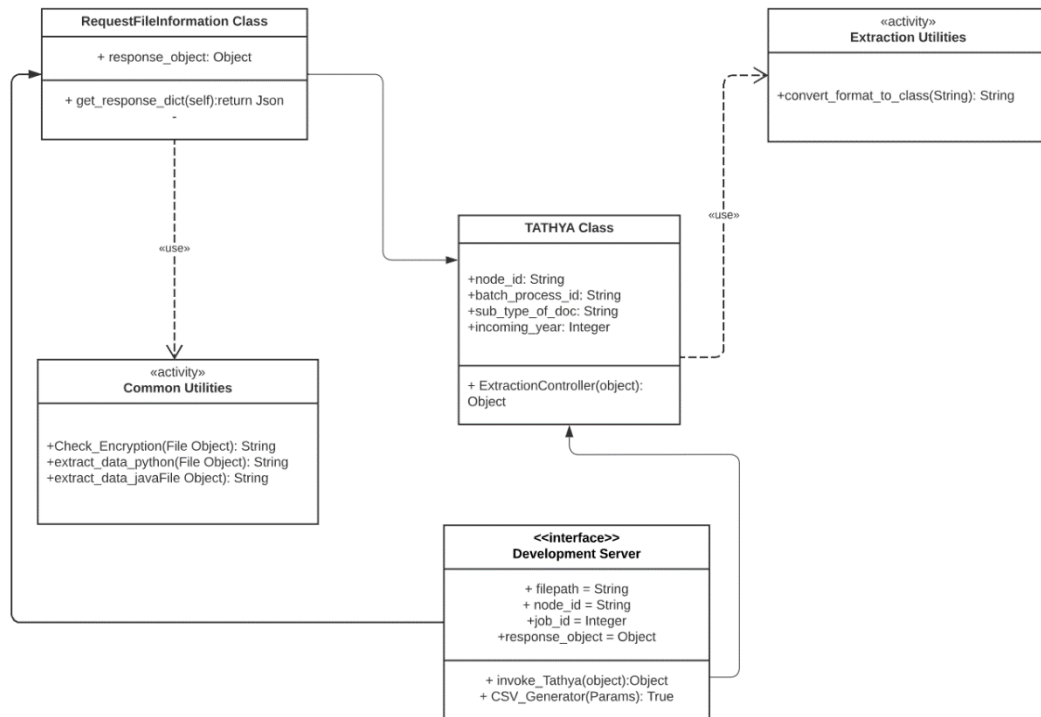


Figure 6.3 Class Diagram

## **CHAPTER 7**

### **Conclusion**



- The IDEAC module of SMSF audit software is a critical tool for auditors in the SMSF industry. The module automates the process of extracting and classifying data from complex documents such as invoices, receipts, and forms, reducing the time and effort required for auditors to complete their work.
- The module can properly extract data and classify it into predetermined categories by utilizing machine learning techniques and artificial intelligence, which lowers the possibility of errors and ensures compliance with industry rules. Additionally, the module's capacity to learn and modify to new data and document types makes it a useful tool for auditors operating in a regulatory environment that is continuously changing.
- Overall, the Intelligent Document Extraction and Classification module of SMSF audit software streamlines the audit process, improves accuracy, and enhances productivity, making it a must-have tool for auditors in the SMSF industry.

## **CHAPTER 8**

### **Limitations and Future Enhancements**

## 8.1 Limitations

The Intelligent Document Extraction and Classification module of the SMSF (Self-Managed Super Fund) software has several limitations that could be improved upon in the future. Some of these limitations include:

- **Accuracy:**

The accuracy of the TATHYA depends on the quality and consistency of the document formats. Inconsistent or low-quality document(OCR) can lead to inaccurate results. Future enhancements should focus on improving the accuracy of the module.

- **Speed:**

The current module may not be able to process large volumes of data quickly, which can be a hindrance in situations where quick decision-making is needed. Enhancements should be made to improve the speed of the module.

- **Flexibility:**

The module may not be flexible enough to accommodate different document types or structures. Future enhancements should allow the module to be more adaptable to different document types and structures.

## 8.2 Future Enhancements

To enhance the Intelligent Document Extraction and Classification module, future developments could include:

- The use of machine learning algorithms to improve accuracy and increase efficiency in processing large volumes of data.
- The inclusion of a natural language processing (NLP) module to enable the module to better interpret unstructured data and extract relevant information.
- The incorporation of more advanced data visualization techniques to enable better interpretation of data.
- The use of coordinates that of data extracted can improve user expression to easily navigate user to see data from where extracted.
- The development of a more user-friendly interface to enable easier navigation and use of the module.

## **CHAPTER 9**

### **References**

- [tabula-py: Read tables in a PDF into DataFrame](#)
- [A User Guide of Pandas.pydata](#)
- [PyMuPDF User Guide](#)
- Kaur, H., & Gupta, A. (2020). Intelligent document extraction using machine learning techniques. 2020 IEEE International Conference on Computing, Electronics & Communications Engineering (IEEE iCCECE), 1-6.
- Kulkarni, S. P., & Ramaiah, C. K. (2018). Intelligent document extraction and classification using machine learning. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 1378-1382.
- [Using Python Flask and Ajax to Pass Information between the Client and Server](#)
- [Flask web development, one drop at a time](#)
- [Scrape and Extract Data from PDFs Using Python and tabula-py](#)
- [PDF Processing with Python](#)
- Aslam, Fankar & Mohammed, Hawa & Lokhande, Prashant. (2015). Efficient Way Of Web Development Using Python And Flask. International Journal of Advanced Research in Computer Science. 6.
- [Data Extraction from Unstructured PDFs](#)
- [Advanced Text Manipulation Using PyMuPDF](#)
- [Techniques for Detecting and Extracting Tabular Data from PDFs and Scanned Documents: A Survey](#)