International Institute of Information Technology, Bangalore

Project Elective Report

OCR based document validation - Custom Stage

Under the Guidance of
Prof. B. Thangaraju & Sanath & Ramesh

August, 2021 to November, 2021

Bharath Kumar Joshi
IMT2018016

M. SriChandan Teja
MT2020015

K. Sambhavi Apurva
MT2020011

1. Problem Statement

MOSIP is Identity Management Platform for creating digital and foundational IDs. During the creation of ID for a person, he/she needs to provide valid proof documents supporting their claims.

Currently **MOSIP** don't have any steps to validate documents at server side. We propose an OCR based validation stage for the same. Validation may use a document type template and OCR extraction techniques and each country may have different template, so templates are fixed for a country. In this we focused on Indian and Indonesian Passports.

2. Tools and Technologies

1. Eclipse Vert.x for building reactive applications in Java.
2. Spring Boot.
3. Tesseract for OCR library.
4. Apache Commons library for Levenshtein Distance.
5. Mockito and Power Mock for Testing.

3. Work Done

We need to create a stage for recognizing the textual data from proof documents uploaded and we need to validate the contents extracted with the data he provided. The stages are implemented as Verticle using Eclipse Vert.x. So, we implemented the data extraction and validation as part of a stage.

We created a wrapper class called **OCRDocumentClassifierStage** and this is the first class which will be invoked when the verticle is executed. The execution proceeds from this class to **OCRDocumentClassificationProcessor** where the business logic is implemented. This class contains all the logistics need to be taken care like fetching the data the user entered using registration, from **Packet**. The document that is extracted from the packet is sent to **TesseractOCR** class for further processing. In this class the document is given as input to **doOCR()** function of the tesseract. The method extracts all the textual information from the document and returns as a string. And this data will be matched with the data to be validated from the packet. If the data matches then the packet will be accepted else error is returned. The passports, world wide are of 3 types, TD-1, TD-2, and TD-3. We are working on passports of type TD-3. Every passport has a MRZ (Machine Readable Zone) part at the bottom of the passport. As per the type of the passport, the length of MRZ lines varies. The following are MRZ portions of different passport types:

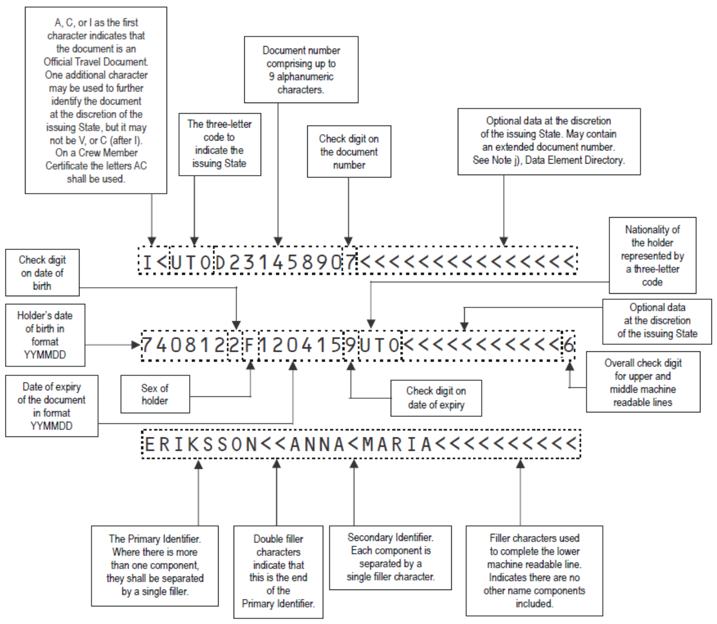

A, C, or I as the first character indicates that the document is an Official Travel Document. One additional character may be used to further identify the document at the discretion of the issuing State, but it may not be V, or C (after I). On a Crew Member Certificate the letters AC shall be used.
The three-letter code to indicate the issuing State
Document number comprising up to 9 alphanumeric characters.
Check digit on the document number
Optional data at the discretion of the issuing State. May contain an extended document number. See Note j), Data Element Directory.
I<UTOD231458907<<<<<<<<<<<<<<<
Nationality of the holder represented by a three-letter code
Check digit on date of birth
Holder's date of birth in format YYMMDD
Optional data at the discretion of the issuing State
7408122F1204159UTO<<<<<<<<<<<6
Overall check digit for upper and middle machine readable lines
Date of expiry of the document in format YYMMDD
Sex of holder
Check digit on date of expiry
ERIKSSON<<ANNA<MARIA<<<<<<<<<<
The Primary Identifier. Where there is more than one component, they shall be separated by a single filler.
Double filler characters indicate that this is the end of the Primary Identifier.
Secondary Identifier. Each component is separated by a single filler character.
Filler characters used to complete the lower machine readable line. Indicates there are no other name components included.

Figure 3.1: MRZ portion of passport Type TD-1

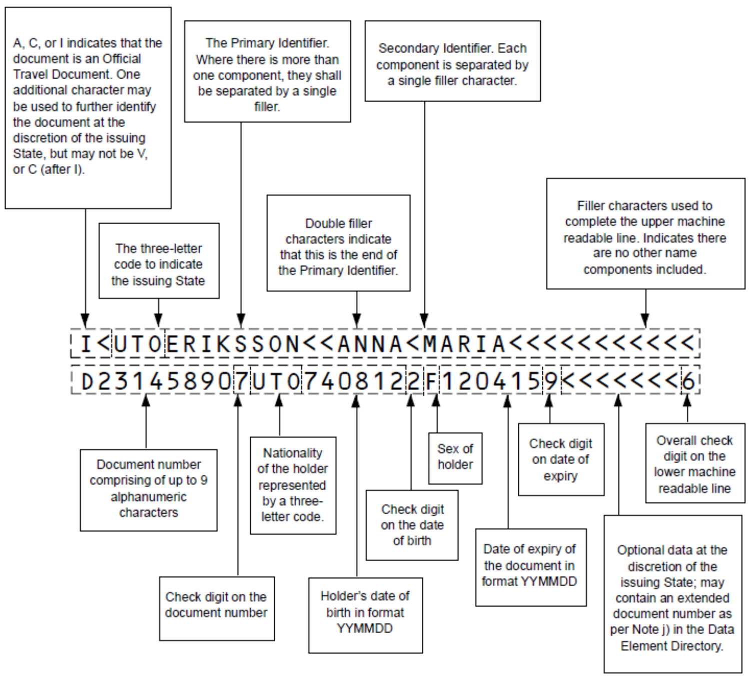

A, C, or I indicates that the document is an Official Travel Document. One additional character may be used to further identify the document at the discretion of the issuing State, but may not be V, or C (after I).
The Primary Identifier. Where there is more than one component, they shall be separated by a single filler.
Secondary Identifier. Each component is separated by a single filler character.
The three-letter code to indicate the issuing State
Double filler characters indicate that this is the end of the Primary Identifier.
Filler characters used to complete the upper machine readable line. Indicates there are no other name components included.
I<UTOERIKSSON<<ANNA<MARIA<<<<<<<<<<<
D231458907UTO7408122F1204159<<<<<<<6
Document number comprising of up to 9 alphanumeric characters
Nationality of the holder represented by a three-letter code.
Sex of holder
Check digit on date of expiry
Overall check digit on the lower machine readable line
Check digit on the date of birth
Date of expiry of the document in format YYMMDD
Check digit on the document number
Holder's date of birth in format YYMMDD
Optional data at the discretion of the issuing State; may contain an extended document number as per Note j) in the Data Element Directory.

Figure 3.2: MRZ portion of passport Type TD-2

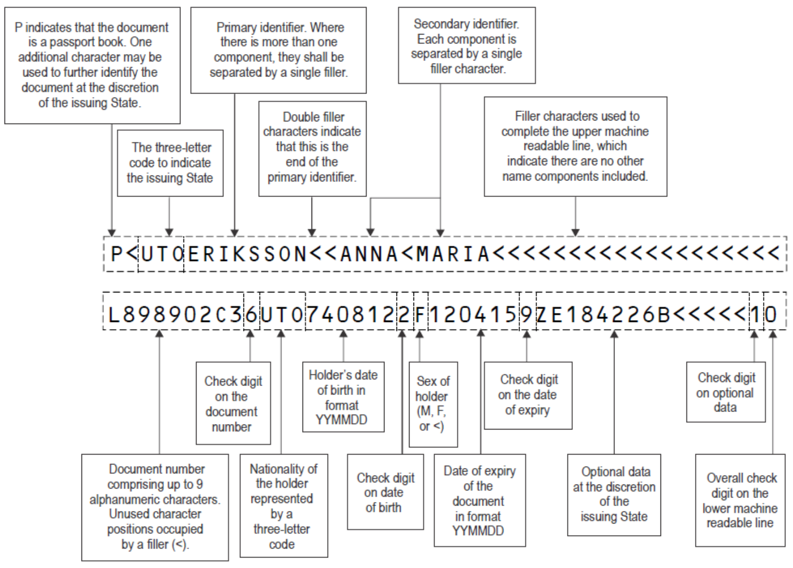

P indicates that the document is a passport book. One additional character may be used to further identify the document at the discretion of the issuing State.
Primary identifier. Where there is more than one component, they shall be separated by a single filler.
Secondary identifier. Each component is separated by a single filler character.
The three-letter code to indicate the issuing State
Double filler characters indicate that this is the end of the primary identifier.
Filler characters used to complete the upper machine readable line, which indicate there are no other name components included.
P<UTOERIKSSON<<ANNA<MARIA<<<<<<<<<<<<<<<<<<<
L898902C36UTO7408122F1204159ZE184226B<<<<<10
Check digit on the document number
Holder's date of birth in format YYMMDD
Sex of holder (M, F, or <)
Check digit on the date of expiry
Check digit on optional data
Document number comprising up to 9 alphanumeric characters. Unused character positions occupied by a filler (<).
Nationality of the holder represented by a three-letter code
Check digit on date of birth
Date of expiry of the document in format YYMMDD
Optional data at the discretion of the issuing State
Overall check digit on the lower machine readable line

Figure 3.3: MRZ portion of passport Type TD-3

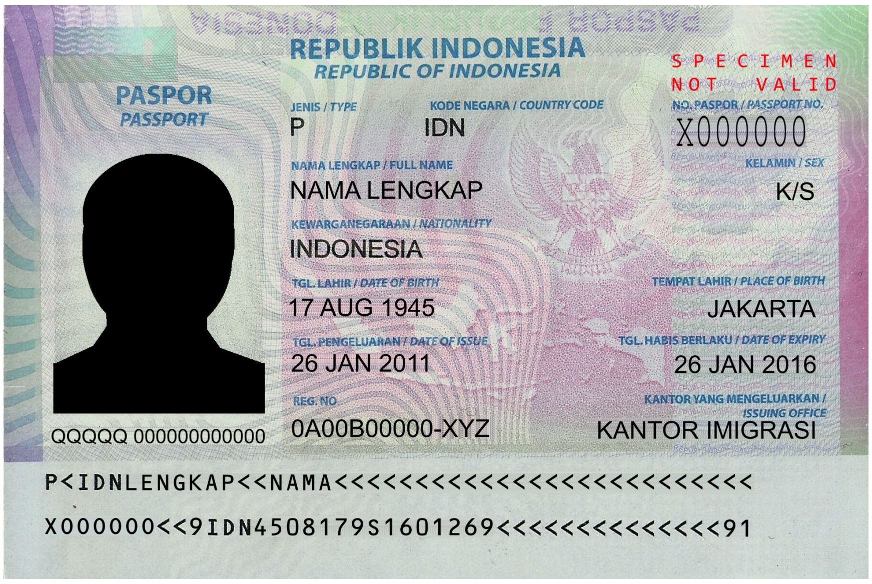
There are 44 characters in each line of MRZ portion of TD-3 type passport. And as shown in the above image, in the first line, first 2 characters represent document type. Next 3 characters represent country code. All the characters following the country code is the name. Each field is separated with "»" character and space in the name is represented by ">" character. In the 2nd line, first 9 characters represent document number, then comes the check digit for the document number. next 3 characters indicate the nationality code for the document holder. Next 6 digits are the date of birth followed by check digit for date of birth. Next 6 digits are the date of expiry of the document followed by check digit for date of expiry.

Now as we want to validate name of the user, the name part is extracted from the output of OCR. We extracted the required fields from output of OCR in two ways. One is from MRZ, another way is from the main portion of passport where the user details can be seen. We have extracted the NAME part in the ways mentioned above. We have used Levenshtein distance as the metric to evaluate the similarity between the input data and extracted data. It returns the number of characters different between the 2 strings. When the name is extracted from the document using 2nd way i.e. from the central portion of the passport, then the input name is checked in each line of the output data. This is done because the image format of all the passports is not the same.

4. Results

4.1. Passports Used

The following passports are used for the experimentation.



REPUBLIK INDONESIA
REPUBLIC OF INDONESIA

PASPOR
PASSPORT

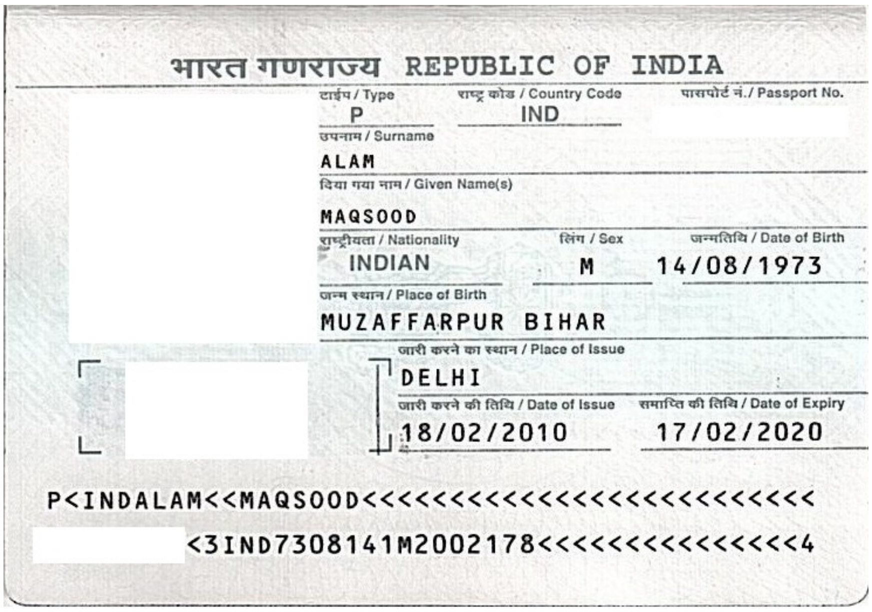
SPECIMEN
NOT VALID

JENIS / TYPE: P	KODE NEGARA / COUNTRY CODE: IDN	NO. PASPOR / PASSPORT NO.: X000000
NAMA LENGKAP / FULL NAME: NAMA LENGKAP		KELAMIN / SEX: K/S
KEWARGANEGARAAN / NATIONALITY: INDONESIA		
TGL. LAHIR / DATE OF BIRTH: 17 AUG 1945		TEMPAT LAHIR / PLACE OF BIRTH: JAKARTA
TGL. PENGELUARAN / DATE OF ISSUE: 26 JAN 2011		TGL. HABIS BERLAKU / DATE OF EXPIRY: 26 JAN 2016
REG. NO: 0A00B00000-XYZ		KANTOR YANG MENGELUARKAN / ISSUING OFFICE: KANTOR IMIGRASI

QQQQQ 000000000000

```
P<IDNLENGKAP<<NAMA<<<<<<<<<<<<<<<<<<<<<<<<<<
X000000<<9IDN4508179S1601269<<<<<<<<<<<<<<91
```

Figure 4.1: Indonesian Passport

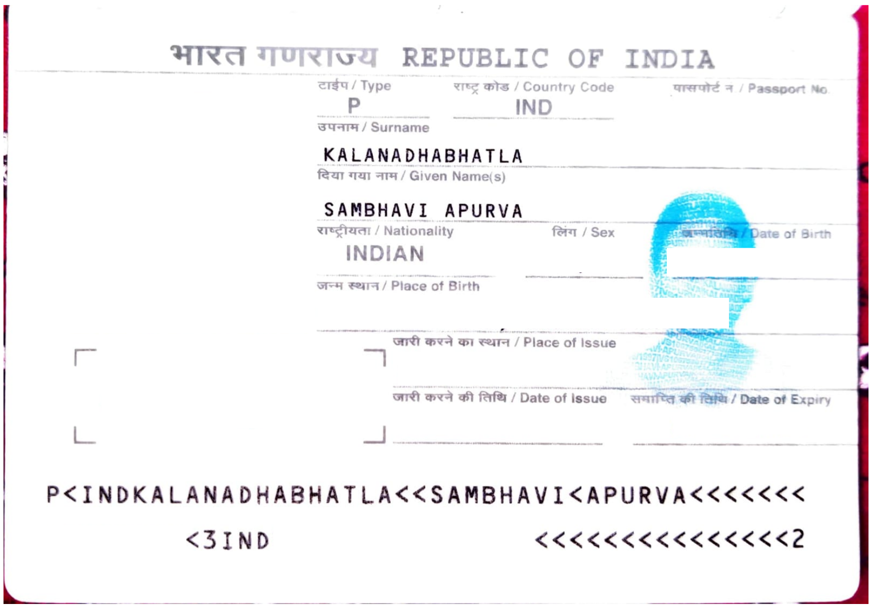


भारत गणराज्य REPUBLIC OF INDIA

टाईप / Type: P	राष्ट्र कोड / Country Code: IND	पासपोर्ट नं. / Passport No.
उपनाम / Surname: ALAM		
दिया गया नाम / Given Name(s): MAQSOOD		
राष्ट्रीयता / Nationality: INDIAN	लिंग / Sex: M	जन्मतिथि / Date of Birth: 14/08/1973
जन्म स्थान / Place of Birth: MUZAFFARPUR BIHAR		
जारी करने का स्थान / Place of Issue: DELHI		
जारी करने की तिथि / Date of Issue: 18/02/2010	समाप्ति की तिथि / Date of Expiry: 17/02/2020	

```
P<INDALAM<<MAQSOOD<<<<<<<<<<<<<<<<<<<<<<<<<<
         <3IND7308141M2002178<<<<<<<<<<<<<<4
```

Figure 4.2: Indian Passport 1



भारत गणराज्य REPUBLIC OF INDIA

टाईप / Type
P

राष्ट्र कोड / Country Code
IND

पासपोर्ट न. / Passport No.

उपनाम / Surname
KALANADHABHATLA

दिया गया नाम / Given Name(s)
SAMBHAVI APURVA

राष्ट्रीयता / Nationality
INDIAN

लिंग / Sex

जन्मतिथि / Date of Birth

जन्म स्थान / Place of Birth

जारी करने का स्थान / Place of Issue

जारी करने की तिथि / Date of Issue

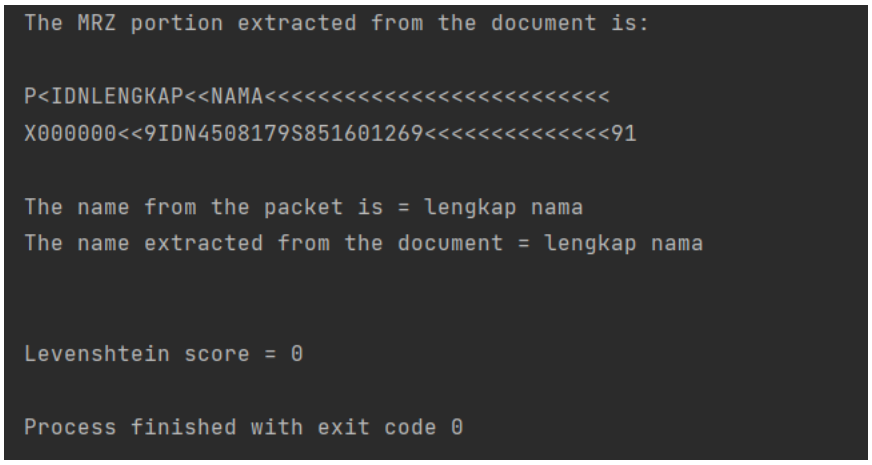
समाप्ति की तिथि / Date of Expiry

```
P<INDKALANADHABHATLA<<SAMBHAVI<APURVA<<<<<<<
     <3IND                  <<<<<<<<<<<<<<<<2
```

Figure 4.3: Indian Passport 2

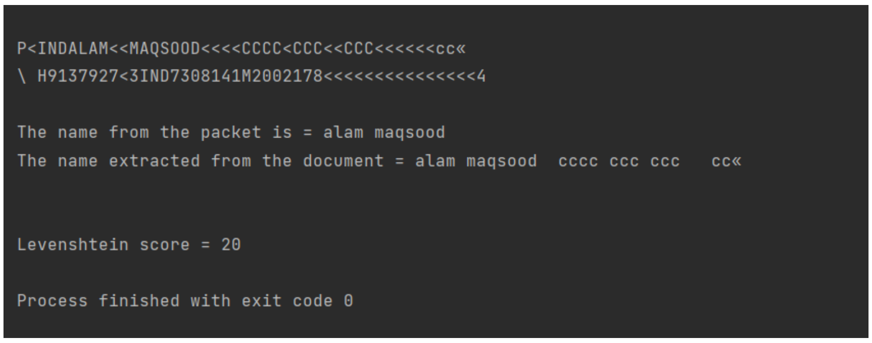
4.2. outputs

Initially we considered the full image and in that we took MRZ part and parsed it and extracted the name from it. The results are as follows:



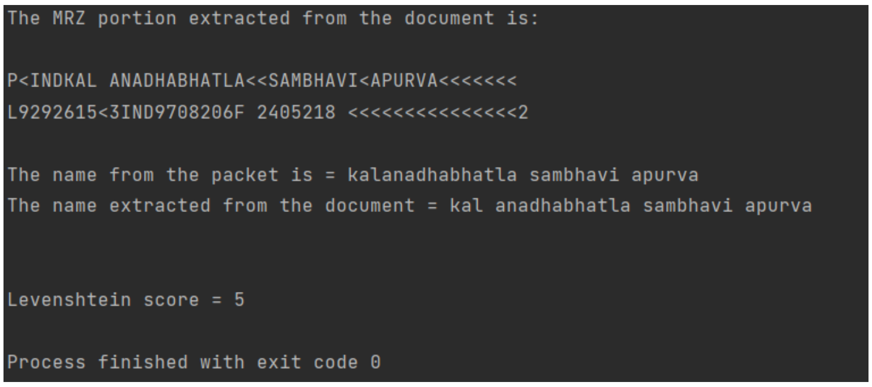
```
The MRZ portion extracted from the document is:

P<IDNLENGKAP<<NAMA<<<<<<<<<<<<<<<<<<<<<<<<<<
X000000<<9IDN4508179S851601269<<<<<<<<<<<<91

The name from the packet is = lengkap nama
The name extracted from the document = lengkap nama


Levenshtein score = 0

Process finished with exit code 0
```

In this passport, the name of the user is **lengkap nama** and the OCR also correctly detected the name. So the levenshtein distance between these two is 0.



```
P<INDALAM<<MAQSOOD<<<<CCCC<CCC<<CCC<<<<<<cc«
\ H9137927<3IND7308141M2002178<<<<<<<<<<<<<<<4

The name from the packet is = alam maqsood
The name extracted from the document = alam maqsood  cccc ccc ccc   cc«


Levenshtein score = 20

Process finished with exit code 0
```

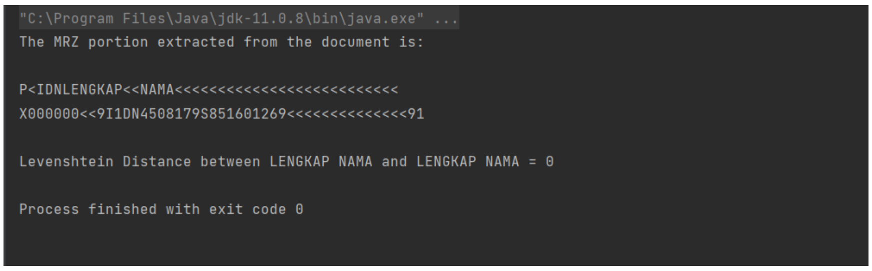
But in this image, the name of the user is **alam maqsood**, but the text detected by the OCR is **alam maqsood cccc ccc ccc cc«**. This is one of the issue with OCR, it detects extra characters. In case if we try to remove the c's present then the c's present in the name will also be replaced.



```
The MRZ portion extracted from the document is:

P<INDKAL ANADHABHATLA<<SAMBHAVI<APURVA<<<<<<<
L9292615<3IND9708206F 2405218 <<<<<<<<<<<<<<<2

The name from the packet is = kalanadhabhatla sambhavi apurva
The name extracted from the document = kal anadhabhatla sambhavi apurva


Levenshtein score = 5

Process finished with exit code 0
```

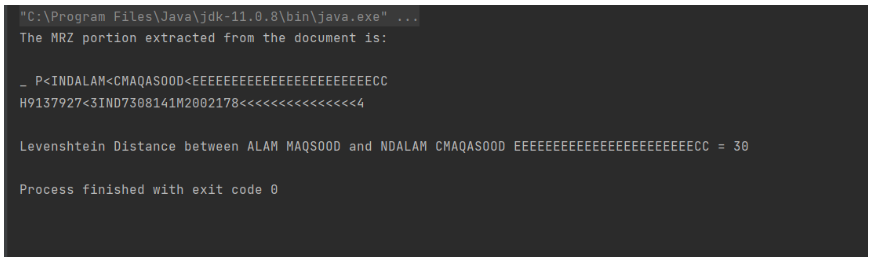
In this image, the name of the user if **kalanadhabhtla sambhavi apurva**, but the text detected by OCR is **kal anadhabhtla sambhavi apurva**. In most of the cases OCR detects extra characters or detects them wrong.

Next we tried cropping the image to MRZ part, and then performing OCR on it. It gave us the following results:



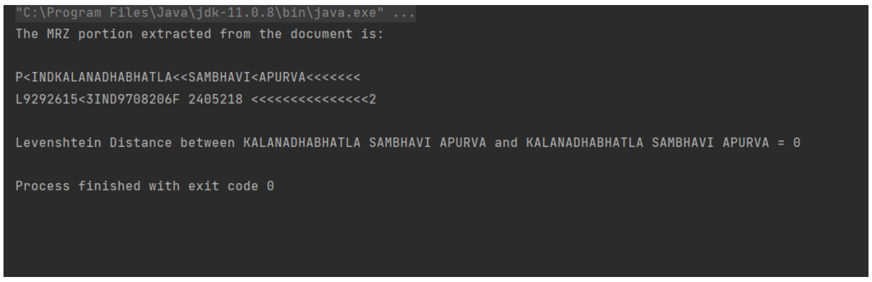
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
The MRZ portion extracted from the document is:

P<IDNLENGKAP<<NAMA<<<<<<<<<<<<<<<<<<<<<<<<<<<
X000000<<9I1DN4508179S851601269<<<<<<<<<<<<<<91

Levenshtein Distance between LENGKAP NAMA and LENGKAP NAMA = 0

Process finished with exit code 0
```

In this the name of the user is **lengkap nama** and the OCR detected text is also **lengkap nama**.



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
The MRZ portion extracted from the document is:

_ P<INDALAM<CMAQASOOD<EEEEEEEEEEEEEEEEEEEEEEECC
H9137927<3IND7308141M2002178<<<<<<<<<<<<<<<4

Levenshtein Distance between ALAM MAQSOOD and NDALAM CMAQASOOD EEEEEEEEEEEEEEEEEEEEEEECC = 30

Process finished with exit code 0
```

Here the name of the user is **Alam Maqsood** but OCR detected text is **NDALAM CMAQASOOD EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEC** which is completely irrelevant to the name of the user. The levenshtein score obtained is 30.

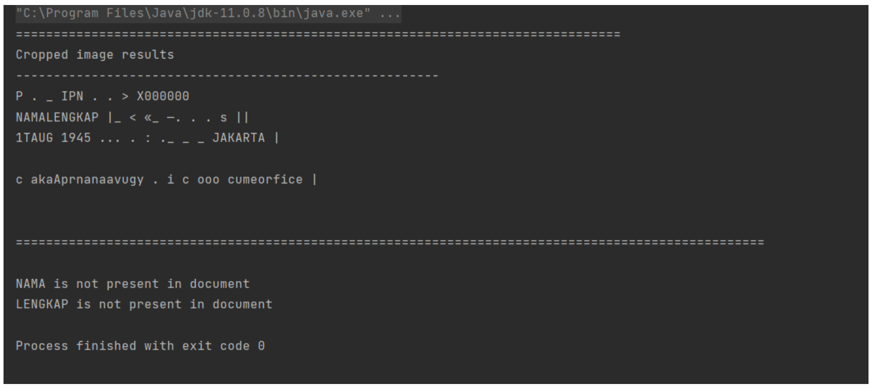


```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
The MRZ portion extracted from the document is:

P<INDKALANADHABHATLA<<SAMBHAVI<APURVA<<<<<<
L9292615<3IND9708206F 2405218 <<<<<<<<<<<<<<2

Levenshtein Distance between KALANADHABHATLA SAMBHAVI APURVA and KALANADHABHATLA SAMBHAVI APURVA = 0

Process finished with exit code 0
```

In this the name of the user is **kalanadhabhtla sambhavi apurva** and the OCR detected text is also **kalanadhabhtla sambhavi apurva**.

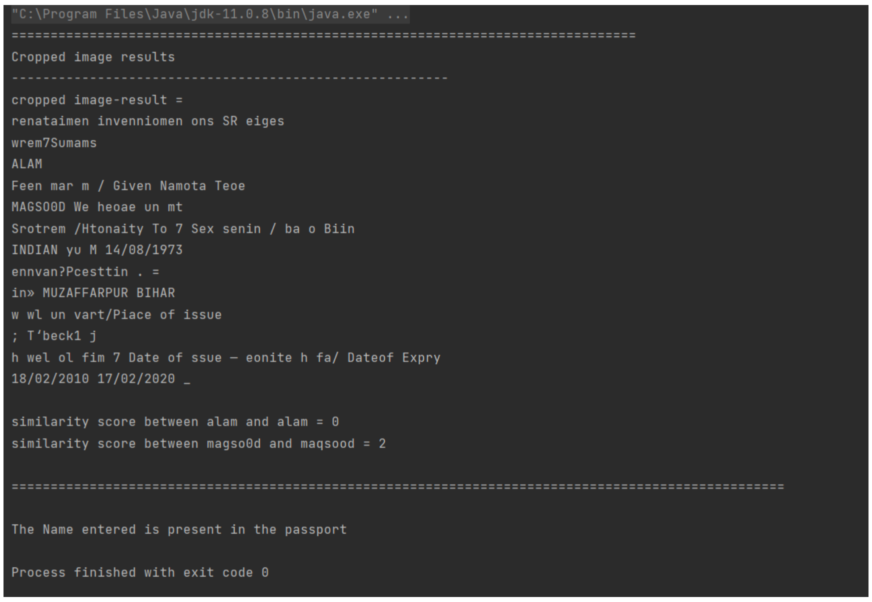
Next, with the assumption that we would get better results by cropping the center part, we proceeded by cropping the center part in the passport where all the fields like Name, DOB etc of the user will be present.

Below are the results we obtained:



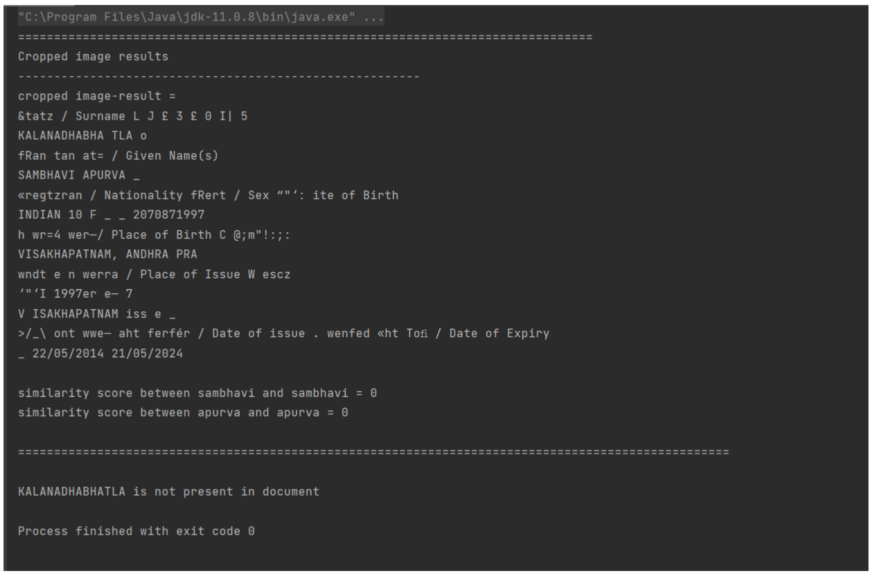
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
===============================================================================
Cropped image results
-------------------------------------------------------
P . _ IPN . . > X000000
NAMALENGKAP |_ < «_ —. . . s ||
1TAUG 1945 ... . : ._ _ _ JAKARTA |

c akaAprnanaavugy . i c ooo cumeorfice |


==================================================================================================

NAMA is not present in document
LENGKAP is not present in document

Process finished with exit code 0
```

If we see here, there is no space between the first name and last name of the user and it is detected as **NAMALENGKAP** instead of **LENGKAP** and **NAMA**. This is because the OCR doesn't recognize the text properly.



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
================================================================================
Cropped image results
--------------------------------------------------------
cropped image-result =
renataimen invenniomen ons SR eiges
wrem7Sumams
ALAM
Feen mar m / Given Namota Teoe
MAGSO0D We heoae un mt
Srotrem /Htonaity To 7 Sex senin / ba o Biin
INDIAN yu M 14/08/1973
ennvan?Pcesttin . =
in» MUZAFFARPUR BIHAR
w wl un vart/Piace of issue
; T‘beck1 j
h wel ol fim 7 Date of ssue — eonite h fa/ Dateof Expry
18/02/2010 17/02/2020 _

similarity score between alam and alam = 0
similarity score between magso0d and maqsood = 2

===================================================================================================

The Name entered is present in the passport

Process finished with exit code 0
```

If we observe the above image, **Tesseract** is unable to recognize most of the data in the central part of the passport and it outputted some garbage data in place of data present in passport. It is able to recognize only the bold text which can be seen clearly.



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
================================================================================
Cropped image results
--------------------------------------------------------
cropped image-result =
&tatz / Surname L J £ 3 £ 0 I| 5
KALANADHABHA TLA o
fRan tan at= / Given Name(s)
SAMBHAVI APURVA _
«regtzran / Nationality fRert / Sex “"‘: ite of Birth
INDIAN 10 F _ _ 2070871997
h wr=4 wer—/ Place of Birth C @;m"!:;:
VISAKHAPATNAM, ANDHRA PRA
wndt e n werra / Place of Issue W escz
‘"‘I 1997er e— 7
V ISAKHAPATNAM iss e _
>/_\ ont wwe— aht ferfér / Date of issue . wenfed «ht Toñ / Date of Expiry
_ 22/05/2014 21/05/2024

similarity score between sambhavi and sambhavi = 0
similarity score between apurva and apurva = 0

===================================================================================================

KALANADHABHATLA is not present in document

Process finished with exit code 0
```

Here, the actual name of the user is **KALANADHABHATLA SAMBHAVI APURVA** but the text detected by OCR is **KALANADHABHA TLA o SAMBHAVI APURVA**. Because of this poor detection by OCR, we are unable to process the input properly, if we combine the space between **KALANADHABHA** and **TLA o** then the word becomes **KALANADHABHATLAo** then we have to remove last "o". Another issue is that this is specific to this particular passport. OCR adds extra characters to the text at different positions in each passport, because of this generating a common pattern for all passports is becoming highly difficult.

5. Challenges Faced

1. When using power mockito to mock the functions of classes during testing, if **anyString()** in parameters of function, then mocking did not work properly. So instead of **anyString()**, **any()** must be used. This is because anyString() overwrites any().

2. Documents like passports are not accepted in PDF, JPG, JPEG, PNG or other image and PDF formats. The documents must be sent in Byte Array format i.e. **Byte[]**. So when the document is received in this format then the same format document is sent to OCR also. But the OCR doesn't take input in Byte[] format. It takes input in **BufferedImage** format. So first Byte[] must be converted to **ByteArrayInputStream** then **ByteArrayInputStream** must be converted to **BufferedImage** format. Then this will be given as input to **doOCR()** function of Tesseract.

3. **Tesseract Text Detection Issues**:

    (a) Adds extra characters in the word or at the end of the word or at the beginning of the word. These extra characters may be integers or alphabets (lower or upper case) or special characters.

    (b) It doesn't detect properly when the background is not white or when the image is a little disoriented i.e. tilted.

    (c) Due to the above issues, « are detected as CC, O is detected as 0. It adds few extra '<'s in the middle. Because of all these, we are unable to form a generalized pattern for processing of passport. Each passport has a different format.

    (d) Sometimes, extra lines were also added by OCR at the end of the actual text, which makes it difficult to make one pattern for all passports.

6. Summary

So, after experimenting with various methods, we have followed the method 1 i.e. full image is given as input to Tesseract and it converts text for the full document, in that we take the last two lines which has MRZ. Then we process this MRZ part and extract name. Then we compare this extracted name with the actual name from the packet. In this way Document validation has been done.

7. References

1. https://vertx.io/docs/

2. https://docs.mosip.io/platform/

3. https://www.geeksforgeeks.org/tesseract-ocr-with-java-with-examples/

4. https://github.com/tesseract-ocr/tesseract