

Design Doc - Mobile Number Validation Feature

This document is meant for mobile number validation feature.

constants.py

Added 3 new constants

```
# ---Mobile num validation ----
DATA = {"records": []}
VALID_COUNTRY_LIST = ["91", "45", "67", "56"]
EXCLUDED_NUMBERS = [9898989898, 9999999999, 8888888888]
```

user_utils.py

Required constants are imported

```
# import constants
from constants import *
```

is_excluded function is defined for excluding the specific mobile numbers for example look in constants file for EXCLUDED_NUMBERS variable which has pre-defined numbers [9898989898, 9999999999, 8888888888] these numbers are excluded for validation as customer wants to excluded these numbers based on the requirement. If the function receives an argument which is available in the excluded list then function return True otherwise False.

NEW function

```
def is_excluded(mobile_num):
    if mobile_num in EXCLUDED_NUMBERS:
        print(f"{mobile_num} in exluded list")
        print("Mobile verification is successfull")
        return True
    return False
```

is_valida_country function is designed to validate the country code. If the received mobile str first 2 chars in the VALID_COUNTRY_LIST then function returns true otherwise since it is unknown number function raises the Value error.

NEW function

```
def is_valid_country(converted_str):
    if converted_str[:2] in VALID_COUNTRY_LIST:
        print("Mobile verification is successfull")
        return True
    else:
        raise ValueError(f"Invalid country code - {converted_str[:2]}")
        valid country codes are {VALID_COUNTRY_LIST}")
```

is_mobile_length function checks the input mobile number whose length should be 12 then function returns True otherwise it

raises the exception (ValueError).

NEW function

```
def is_mobile_length_valid(converted_str):  
    # Mobile length must be 12 digits  
    if len(converted_str) == 12:  
        return True  
    else:  
        raise ValueError(f"Invalid Mobile number length {len(converted_str)}  
        and valid length is 12")
```

is_valid_type function checks the mobile number type - If the provided number is int then function returns True value otherwise it raises the exception as mobile number should be of int type.

NEW function

```
def is_valid_type(mobile):  
    if isinstance(mobile, int):  
        return True  
    else:  
        raise ValueError(f"Invalid mobile number type - {type(mobile)}")
```

app.py

Importing the necessary functions from user_utils module

```
from user_utils import is_valid_type, is_mobile_length_valid, \  
    is_valid_country, is_excluded
```

check_mobile_num function is the main function which takes the input data(mobile num) from the real -users. And validation performs in the user_utils module based on the validation if it is successful then it returns True otherwise false.

NEW function

```
def check_the_mobile_num(mobile):  
    converted_str = str(mobile) # "674545454545"  
    mobile_num = int(converted_str[2:]) # int("4545454545") => 4545454545  
  
    if is_valid_type(mobile) and is_mobile_length_valid(converted_str):  
        # Excluded persons no further validation required  
        if is_excluded(mobile_num):  
            return True  
        if is_valid_country(converted_str):  
            return True  
        return False
```

