The beauty of PUVGbot

1.1 Import module

```
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.chrome.options import Options
```

Webdriver modules being import from selenium as we are going to use it later when create a room for our lonely friends.

```
from telegram import (InlineKeyboardButton, InlineKeyboardMarkup, ReplyKeyboardMarkup, ReplyKeyboardRemove)
from telegram.ext import (Updater, CommandHandler, RegexHandler, CallbackQueryHandler, ConversationHandler, MessageHandler, Filters)
```

Necessary keyboard modules and handlers being import from telegram for our telegram bot.

```
import logging
import sqlite3
import datetime
import time
import threading
import os
```

Import sqlite3, datetime, time and os for our db preparation. Import logging to debug the program.

1.2 The secret of PUVGbot

```
278  if __name__=='__main__':
279      main()
```

Execute the code and call function main()

```
def main():
    initDB()
    getCategoryDict()

    updater = Updater(token = '462315757:AAEI7Rj_efTi4nCZxT2qGY029btSmkKB2Yo')

    create_conversation=ConversationHandler(
        entry_points=[CommandHandler('createroom',create_room)],

        states={
            ROOMNAME:[MessageHandler(Filters.text,room_name)],
            CATEGORY:[CallbackQueryHandler(category)],
            SHOWTIME:[RegexHandler('^((Today)|(Tomorrow)) ([0-9]|0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]$',showtime)]
        },

        fallbacks=[CommandHandler('cancel',cancel)]
    )

    view_conversation=ConversationHandler(
        entry_points=[CommandHandler('viewroom',view_room)],

        states={
            CATEGORY:[CallbackQueryHandler(view_category)]
        },

        fallbacks=[CommandHandler('cancel',cancel)]
    )

    dispatcher = updater.dispatcher
    dispatcher.add_handler(CommandHandler('start', start))
    dispatcher.add_handler(create_conversation)
    dispatcher.add_handler(view_conversation)

    updater.start_polling()
```

Let's have a look on our main() before go though other functions in detail. I'll explain the work flow in main() and giving further explanation in each of the functions later. First of all, main() will initialize our db by calling initDB(). getCategoryDict() will look for the existing categories which stored in our current db and then stored it into a global variable name as categoryDict.

In PUVGbot, we will have two different types of conversation which are create_conversation and view_conversation. As we talked about earlier, we have used the handlers which are import

from telegram to control the conversation between bot and user. For create_conversation aka /createroom, PUVGbot will go through the following functions in sequences , create_room(), room_name(), category() and showtime() in order to obtain the url for the room and display it for our lonely user. For view_conversation aka /viewroom, PUVGbot will go through view_room() and view_category() to obtain a list of url for the room and display it for our lonely user.

```
def initDB():
    try:
        conn=sqlite3.connect("watch2gether.db")
        cursor=conn.cursor()
        cursor.execute('''
            CREATE TABLE IF NOT EXISTS TB_ROOM
            (
            PK_ROOMID INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
            ROOMNAME TEXT NOT NULL,
            CODE_CATEGORY TEXT NOT NULL,
            SHOWTIME DATE NOT NULL,
            URL TEXT NOT NULL
            )
            ''')

        cursor.execute('''CREATE TABLE IF NOT EXISTS TB_CODE_CATEGORY
            (CODE PRIMARY KEY NOT NULL,
            DESCRIPTION TEXT NOT NULL)
            ''')

        cursor.execute('''INSERT INTO TB_CODE_CATEGORY(CODE,DESCRIPTION)
            SELECT 'CAT0001','Drama' WHERE NOT EXISTS(SELECT 1 FROM TB_CODE_CATEGORY WHERE CODE='CAT0001')
            ''')

        cursor.execute('''INSERT INTO TB_CODE_CATEGORY(CODE,DESCRIPTION)
            SELECT 'CAT0002','Horror' WHERE NOT EXISTS(SELECT 1 FROM TB_CODE_CATEGORY WHERE CODE='CAT0002')
            ''')

        cursor.execute('''INSERT INTO TB_CODE_CATEGORY(CODE,DESCRIPTION)
            SELECT 'CAT0003','Romantic' WHERE NOT EXISTS(SELECT 1 FROM TB_CODE_CATEGORY WHERE CODE='CAT0003')
            ''')

        conn.commit()
    except Exception as e:
        print(e)
    finally:
        conn.close()
```

initDB() helps to initilize our db. So far we have 3 categories in our db but we would like to have more for our future expansion. It can be simply add on in initDB().

```
def getCategoryDict():
    global categoryDict
    try:
        conn=sqlite3.connect('watch2gether.db')
        cursor=conn.cursor()
        resultSet=cursor.execute(
            '''
            SELECT CODE,DESCRIPTION FROM TB_CODE_CATEGORY
            '''
        )

        for row in resultSet:
            categoryDict[row[0]]=row[1]
    except Exception as e:
        print(e)
    finally:
        conn.close()
```

Grab categories in our db, for selection purpose.

```
def start(bot,update):
    update.message.reply_text(
        'Hi! I am PUVGbot, feel lonely when you watch video alone? We are here for you :)\n'
                'By creating a room, you can send a link to your friends who would like to watch a video with you \n'
                'No friends? No worries , you can join other rooms and share your ideas with them :) \n'
                'Please select the following commands \n \n'
        '<a>/createroom</a> - Create a room and set showtime\n'
        '<a>/viewroom</a>   - View available room\n',
        parse_mode='html',
        reply_markup=ReplyKeyboardRemove()
    )
```

Yong Seng 3:17:23 PM

/start

PU

PUVGbot 3:17:24 PM

Hi! I am PUVGbot, feel lonely when you watch video alone? We are here for you :)
By creating a room, you can send a link to your friends who would like to watch a video with you
No friends? No worries :), you can join other rooms and share your ideas with them :)Please select the following commands

/createroom - Create a room and set showtime
/viewroom - View available room

As the screenshot above, start() was triggered when /start command being entered. A brief introduction and available commands will be given to our new lonely user.

1.3 Create Room

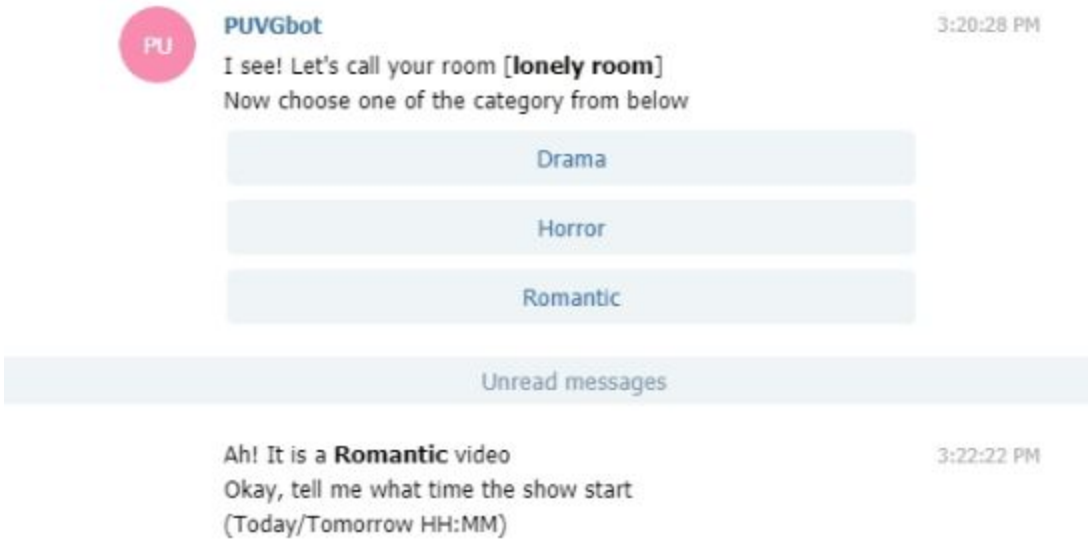
```
def create_room(bot,update):
    global inputRoomName
    global inputCategory
    global inputShowTime
    inputRoomName=''
    inputCategory=''
    inputShowTime=''

    update.message.reply_text(
        'Please tell me your room name?'
    )

    return ROOMNAME
```

From the main(), you must be familiar with create_room now, Yes!, this is the entry point where lonely user input /createroom. As the screenshot above, lonely user need to enter their desired roomname. After that, it'll direct our lonely user to the next function room_name().

```
def room_name(bot,update):
    global inputRoomName
    global inputCategory
    global inputShowTime
    global categoryDict

    inputRoomName=update.message.text

    keyboard = []

    for i in categoryDict.keys():
        keyboard.append([InlineKeyboardButton(categoryDict[i],callback_data=i)])
    reply_markup = InlineKeyboardMarkup(keyboard)
    update.message.reply_text(
        'I see! Let\'s call your room [<b>'+inputRoomName+'</b>]\n'
        'Now choose one of the category from below',
        parse_mode='html',
        reply_markup = reply_markup
    )
    return CATEGORY
```

Now, we have the room name which entered by our lonely user, but this is definitely not the end of PUVGbot, it requires our lonely user to pick the type of the categories from the video which makes our lonely viewers easier to look for the video they wanted to watch. After we have selected the the category above, it will direct lonely user to the next function category().

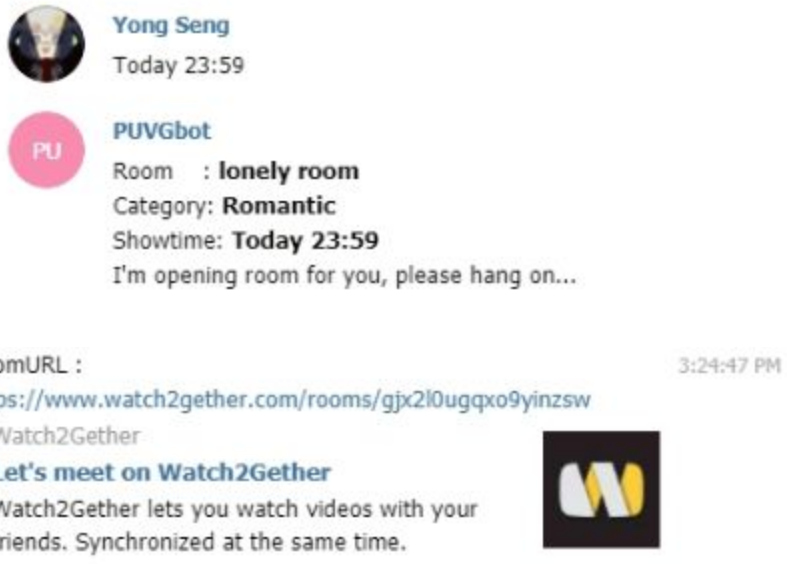
```
def category(bot,update):
    global inputRoomName
    global inputCategory
    global inputShowTime
    global categoryDict

    query=update.callback_query
    inputCategory=query.data

    update.callback_query.message.reply_text(
        'Ah! It is a <b>'+categoryDict.get(inputCategory)+'</b> video\n'
        'Okay, tell me what time the show start\n'
        '(Today/Tomorrow HH:MM)',
        parse_mode='html'
    )

    return SHOWTIME
```



Category that entered by lonely user has been passed down from the previous function. Now lonely user need to set the show time in order to inform other lonely friends when the show will be started.

```
def showtime(bot,update):
    global inputRoomName
    global inputCategory
    global inputShowTime
    global categoryDict

    datetimeString=update.message.text

    day, time=datetimeString.split(' ')

    if(day=='Tomorrow'):
        inputShowTime=datetime.datetime.today() + datetime.timedelta(days=1)
    elif(day=='Today'):
        inputShowTime=datetime.datetime.today()

    hr, mi=time.split(':')

    inputShowTime=inputShowTime.replace(hour=int(hr), minute=int(mi))

    update.message.reply_text(
        'Room     : <b>'+inputRoomName+'</b>\n'
        'Category: <b>'+categoryDict.get(inputCategory)+'</b>\n'
        'Showtime: <b>'+datetimeString+'</b>\n'
        'I\'m opening room for you, please hang on...',
        parse_mode='html'
    )

    chrome_options = Options()
    chrome_options.add_argument("--headless")
    if os.name == "nt":
            driver = webdriver.Chrome(executable_path="chromedriver.exe", chrome_options = chrome_options
    elif os.name == "posix":
            driver = webdriver.Chrome("./chromedriver" , chrome_options = chrome_options)
    driver.wait = WebDriverWait(driver, 5)

    driver.get('https://www.watch2gether.com') #going to site
    driver.find_element_by_css_selector('.ui.primary.button').click()
    room_url = driver.current_url
    driver.quit()

    update.message.reply_text(
        'RoomURL : <a>'+room_url+'</a>\n',
        parse_mode='html'
    )

    data_store(inputRoomName,inputCategory,inputShowTime,room_url)

    return ConversationHandler.END
```

```
def data_store(inputRoomName,inputCategory,inputShowTime,room_url):
    try:
        conn = sqlite3.connect("watch2gether.db")
        cursor = conn.cursor()
        cursor.execute("INSERT INTO TB_ROOM(ROOMNAME,CODE_CATEGORY,SHOWTIME,URL) VALUES (?,?,?,?);",(inputRoomName,inputCategory,inputShowT
        conn.commit()
        conn.close()
    except Exception as e:
        print(e)
```

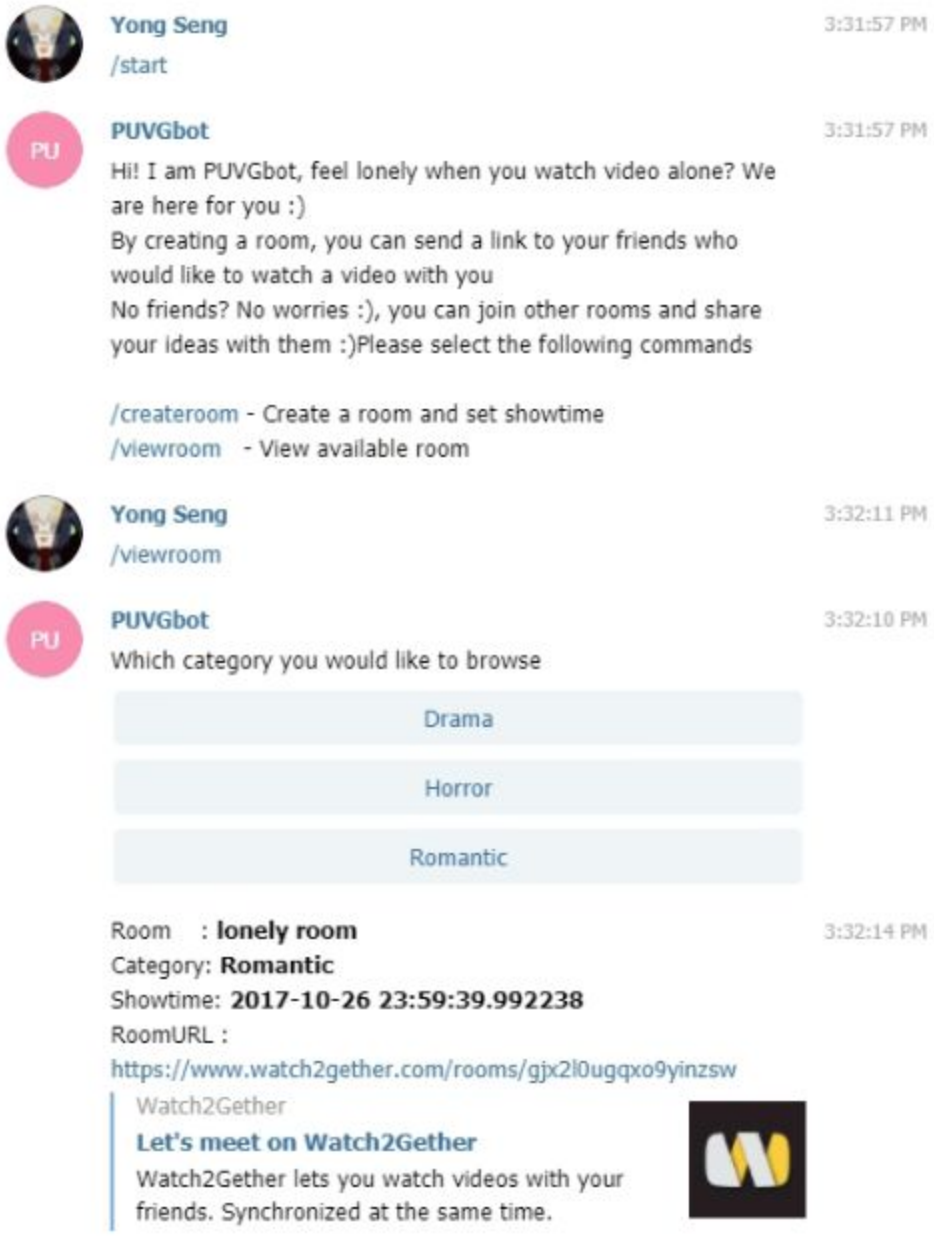


The lonely user choose midnight today, PUVGbot execute it immediately to open a room for him. It may take a few seconds. Your new room will be created. At the same time, the room has been stored in our db by using data_store(). You can share the link to your friends if you wanted to otherwise other lonely friends who are using PUVGbot might join to your room as well.

1.4 View Room

```
def view_room(bot,update):
    global categoryDict

    reply_keyboard=[]
    for i in categoryDict.keys():
        reply_keyboard.append([InlineKeyboardButton(categoryDict[i],callback_data=i)])

    update.message.reply_text(
        'Which category you would like to browse',
        reply_markup=InlineKeyboardMarkup(reply_keyboard)
    )
    return CATEGORY
```

```
def view_category(bot,update):
    global categoryDict

    text=''

    query=update.callback_query
    inputCategory=query.data
    try:
        conn = sqlite3.connect("watch2gether.db")
        cursor = conn.cursor()

        for row in cursor.execute('SELECT * FROM TB_ROOM WHERE CODE_CATEGORY=?',(inputCategory,)):
            text+='Room    : <b>'+row[1]+'</b>\n'+'Category: <b>'+categoryDict.get(row[2])+'</b>\n'+'Showtime: <b>'+row[3]+'</b>\n'+'RoomUR
            text+='\n'
    except Exception as e:
        print(e)

    update.callback_query.message.reply_text(
        text,
        parse_mode='html'
    )

    return ConversationHandler.END
```



The lonely user would like to ensure that his `lonely room` has been created by our PUVGbot. He entered /viewroom which will call out view_room() and a list of categories will show up. He selected the same category as he created previously, view_category() was triggered and his room show up immediately.