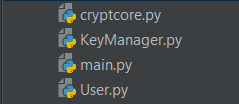
# Exercitiul 3

1. Problema a fost rezolvata in limbajul Python ver 3.5
2. Drept librarie criprografica a fost folosita “cryptography”
   1. pip install cryptography //on windows
   2. sau <https://pypi.python.org/pypi/cryptography> pentru surse
   3. <https://cryptography.io/en/latest/> documentatia librariei
3. Exercitiu proriiu zis

Avem urmatoarele fisiere python

1. **cryptocore.py**

* Fisierul contine clasa CryptCore: aceasta ofera functionalitatile urmatoare:
  + Cripta blocuri de plaintext
  + Decripta bloruri de cryptotext
  + modul cheia si iv-ul specificat
* modul de operarea a criptarii si decriptarii plain/crypt textului se afla in acest fisier
  + encrypt\_blocks(args)
  + decrypt\_blocks(args)
  + details there ☺

1. **KeyManager.py**

* Fisierul contine Clasa KeyManager care contine toate cele 3 key
  + Key1 pentru comunicarea ECB
  + Key2 pentru comunicarea CFB
  + Key3 pentru criptarea celor 2 chei
* Acesta are functia get\_key prin care User-ii pot cere cheia necesara pentru comunicare
* Key1 si Key2 vor fi criptate si trimise clientului cu modul ECB pe 128 biti

1. **User.py**

* Fisierul contine clasa User care are urmatoarele atribute:
  + id unic
  + key\_manager pentru a cere cheia de comunicare
  + key3 pentru a putea decripta cheia de comunicare
  + iv initial vector pentru modul de comunicare CFB
  + users o lista cu utilizatorii cu care poate comunica
  + path calea catre sendbox-ul personal
  + crypt\_helper instant a clasei CryptCore la care apeleaza pentru criptare/decriptare a fisierului.

1. **main.py**

* In acest fisier este reprezentat un demo care reprezinta flow-ul general al exercitiului
  + Se genereaza mai intai key1, key2, key3, iv,
  + Se creeaza o instanta KeyManager
  + Se creeaza 3 utilizatori cu datele necesare
  + In Folder-ul exercitiului sunt pregatite deja niste fisiere pentru exemplu de transimtere a acestora intre utilizatori
  + Demo: rulati main.py (detalii in fisierul sursa)