Štúdia k programu na elektronické podpisy

Autori: Martin Rosa (230648), Vojtech Schiller (231279), Vladimír Peňaz (227802), Michal Stejskal (231282)
Skupina: Skupina č. 9

Rok: 2021

Obsah:

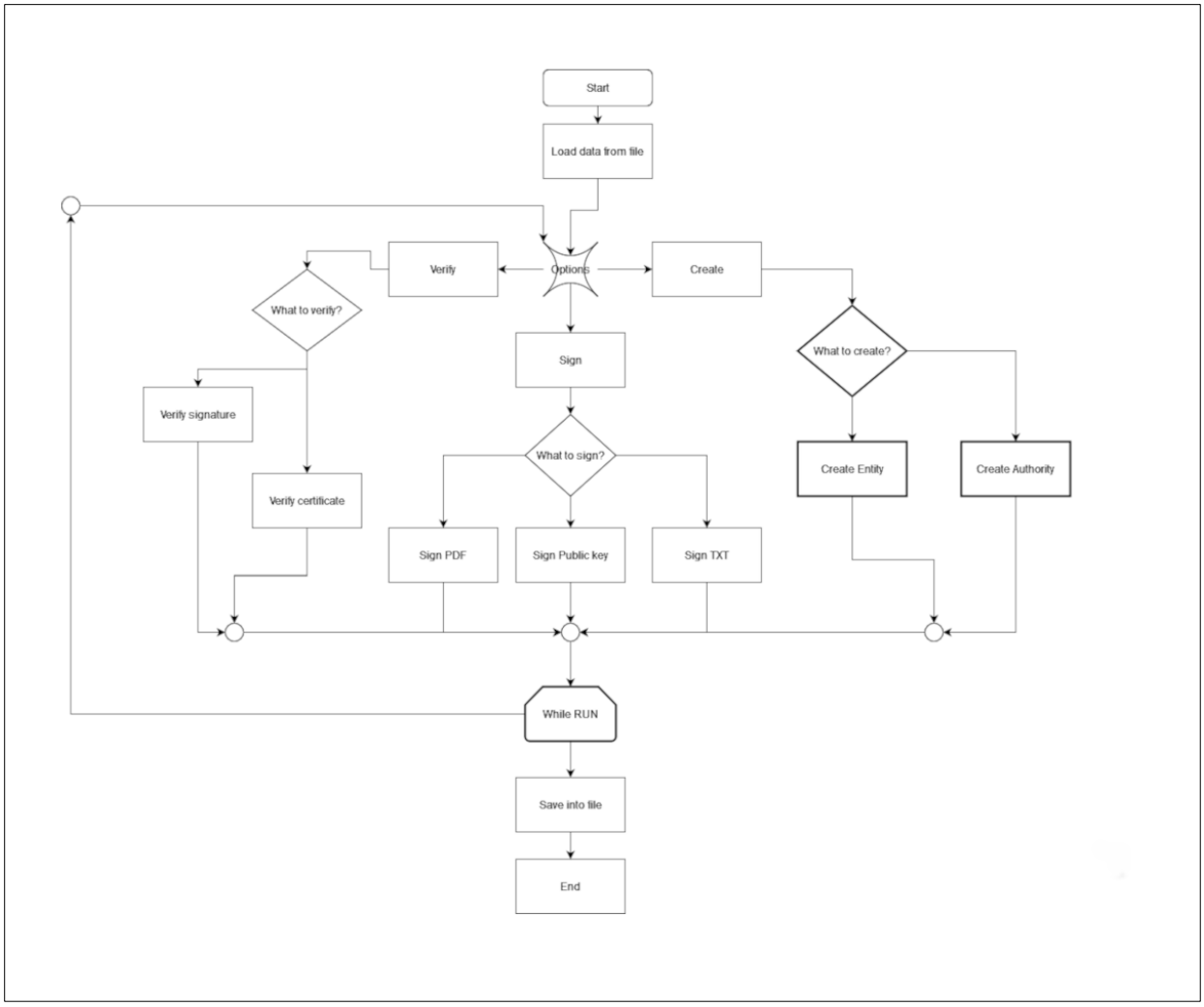
1. Úvod

Projekt je program na podpisovanie dokumentov elektronickým podpisom. Projekt je naprogramovaný v programovacom jazyku menom Python. Zvolili sme si tento programovací jazyk hlavne kvôli jeho jednoduchosti. Python zároveň má veľa užitočných knižníc, ktoré nám prácu na projekte výrazne uľahčili.

2. Popis programu

Program sa skladá z 5 súborov s koncovkou .py, súboru s koncovkou .json a zopár súborov na podpisovanie s koncovkou .txt alebo .pdf. Súbor s koncovkou .json slúži na ukladanie certifikačných autorít a entít pre podpisovanie. Súbory s koncovkou .py obsahujú samotný kód programu. Na spustenie musíme spustiť súbor main.py (či už pomocou IDE alebo pomocou konzole príkazom *python main.py*, resp. *python3 main.py* ak používame Linux ako operačný systém). Následne sa otvorí grafické okno v ktorom môžeme podpísať súbor, skontrolovať podpis, vytvoriť autoritu/entitu alebo pridať certifikát. Grafické rozhranie je vytvorené za pomoci knižnice *tkinter*.
Na podpis sa používajú kľúče vygenerované pomocou RSA algoritmu. Na generáciu kľúčov používame knižnicu *PyCryptodome.* Používame kľúče dĺžky 1024 bitov.
Program vyčíta dáta zo súboru, zahešuje ich a podpíše ich. Následne sú dáta uložené naspäť do súboru.
Pri súboroch s koncovkou .pdf je podpis uložený do metadát. To že je súbor podpísaný následne vidieť aj ak si súbor otvoríme, napr. v Adobe Acrobat Reader. Na prístup k metadatám používame knižnicu *PyPDF2.*
Pri dokumente s koncovkou .txt sme nemohli vstúpiť do metadát, keďže .txt súbor takúto funkcionalitu nemá. Miesto toho ukladáme podpis priamo do súboru a oddeľujeme ho pomocou špeciálneho znaku (znak u2557 na začiatku a znak u255d). Podpis je uložený za samotný text v súbore.
Program si vyčíta dáta pomocou knižnice *tika.*
Pri verifikácii podpisu v podpísanom .txt súbore program dokáže vyhľadať podpis podľa zmienených znakov a overiť ho. Program samozrejme musí podpis oddeliť od samotných dát,(čo je ďalší dôvod prečo používame dané znaky) pretože inak by sa zmenila hodnota hešu a tým by bolo nemožné overiť podpísaný dokument. V .pdf dokumente program opäť vstúpi do metadát, načíta si podpis a overí ho.
Ak vytvoríme entitu, entita si vygeneruje RSA kľúče. Pri autorite, si autorita zoberie entitu a na základe ich si vytvorí certifikát, čím sa stane jej autoritou.
Oba obejkty sa po výtvore zapíšu do súboru s koncovkou .json. Zápis prebieha tak, že najprv si prevedieme základné údaje o entite/autorite do dictionary a následne ich pomocou knižnice *json* prevedieme do json stringu. Z .json súboru si nahráme už vopred vytvorený array menom *objects* a k nemu pripojíme json string. Celý array sa potom naspäť nanovo vloží do súboru. Aby sme rozlíšili autority od entít, json string obsahuje premennú *id* ktorá má buď hodnotu *autority* alebo *entity.*

3. Vývojový diagram



Obrázok 1 – vývojový diagram