

*PREDMET: CS322 C# programski jezik*

Projekat: System Vault

Ime i prezime: Aleksa Radonjić

Broj indeksa: 5651

Asistent: Aleksandar Jovanović

Profesor: Miloš Kostić

1. Uvod

SystemVault je WPF desktop aplikacija razvijena u .NET 8 okruženju, namenjena bezbednom čuvanju konfiguracionih fajlova. Korisnicima omogućava enkripciju osetljivih podataka i njihovo skladištenje na odabranoj lokaciji. Pored toga, aplikacija podržava organizaciju fajlova putem kategorija i pruža jednostavan interfejs za upravljanje.

1. Struktura projekta

Aplikacija je organizovana u skladu sa troslojnom arhitekturom radi boljeg održavanja i skalabilnosti:

* **Presentation sloj**

Korisnički interfejs razvijen u WPF-u. Komunicira sa poslovnom logikom preko servisa i prikazuje relevantne podatke korisniku.

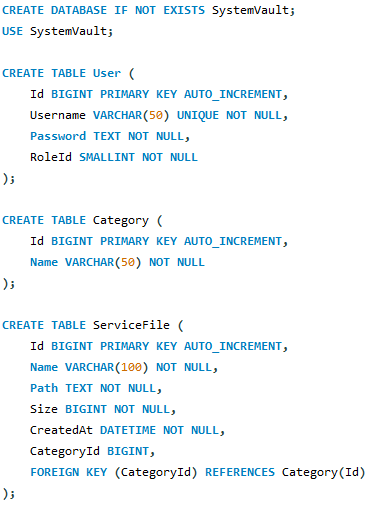
* **BLL (Business Logic Layer)**

Sadrži logiku aplikacije, obradu korisničkih zahteva, validaciju, kriptovanje fajlova, i druge poslovne operacije.

* **DAL (Data Access Layer)**

Ovaj sloj je zadužen za komunikaciju sa MySQL bazom podataka. Kroz jasno definisane repozitorijume, DAL sadrži metode za pristup i upravljanje entitetima. Sve SQL operacije obavljaju se korišćenjem Entity Framework Core ORM-a.

1. Baza podataka

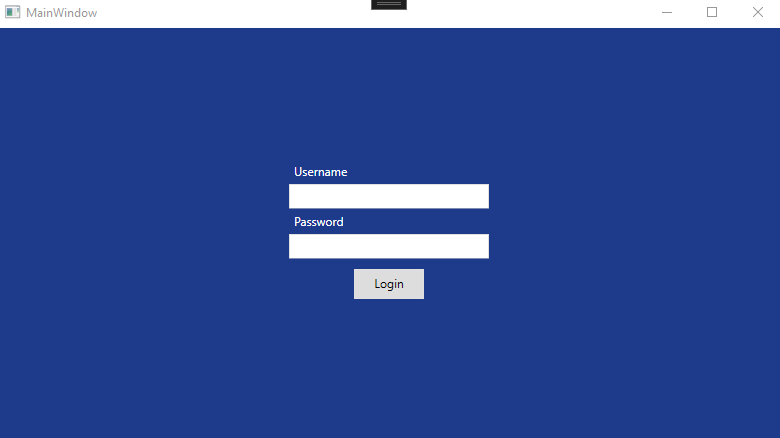


Baza se sastoji od tri tabele:

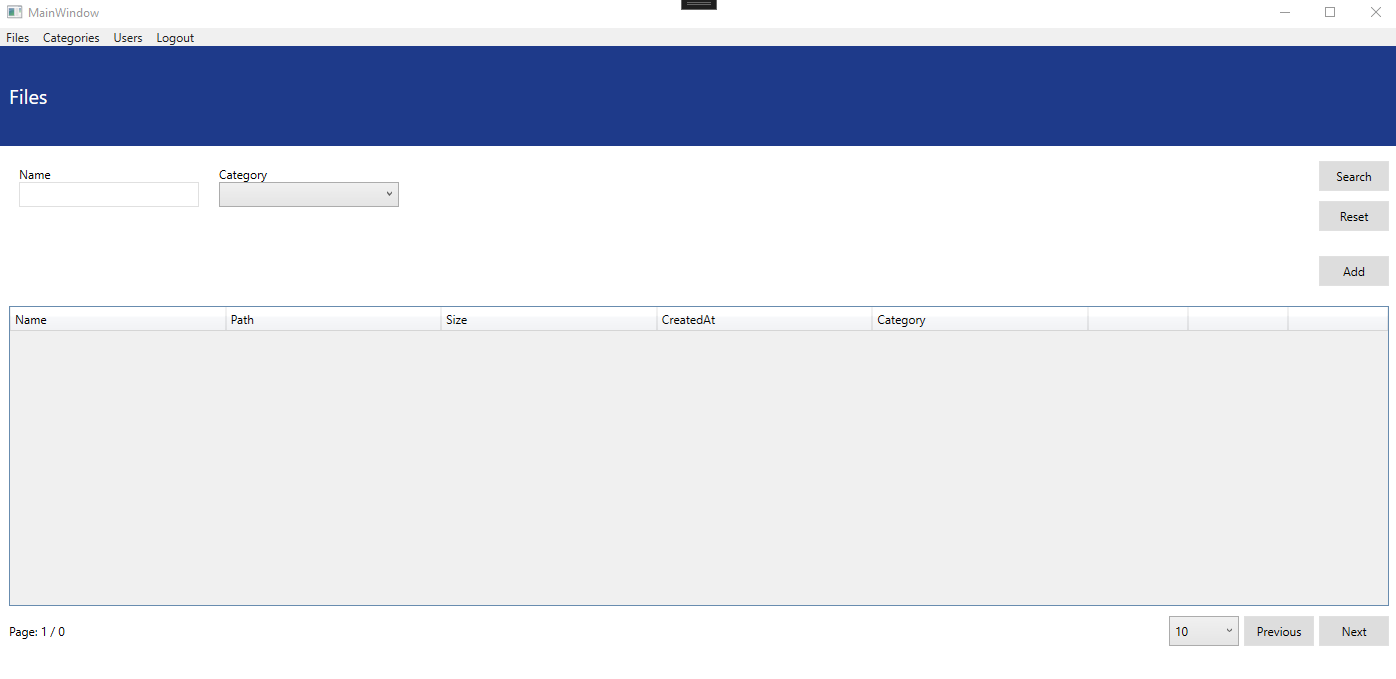
* User – Sadrži korisnike sistema i njihove uloge.
* Category – Definiše kategorije za organizaciju fajlova.
* ServiceFile – Predstavlja fajlove koje korisnici čuvaju, uz vezu ka kategoriji.

1. Implementacija

Kada pokrenemo aplikaciju dobijamo login screen gde je potrebno uneti username i password. Kada popunimo podatke, potrebno je kliknuti „Login“ taster.

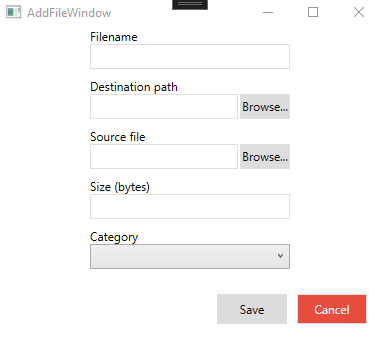


Nakon uspešnog logovanja aplikacija nam otvara stranicu za prikaz fajlova.

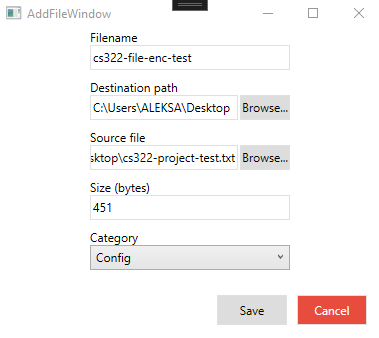


Ovde imamo tabelu za prikaz fajlova, imam pretragu po parametrima i filtriranje. Takođe možemo dodati fajl.

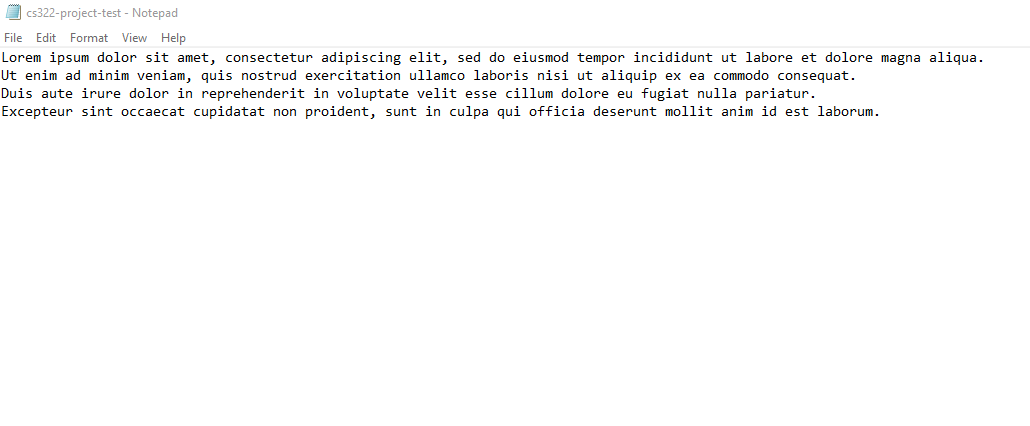
Klikom na taster „Add“ otvara nam se novi prozor gde je potrebno da popunimo podatke.



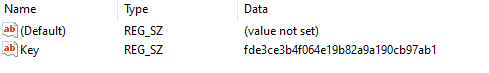
Potrebno je uneti ime fajla koje želimo, putanju gde želimo da se sačuva, putanju do fajla koji želimo da enkriptujemo i kategoriju.



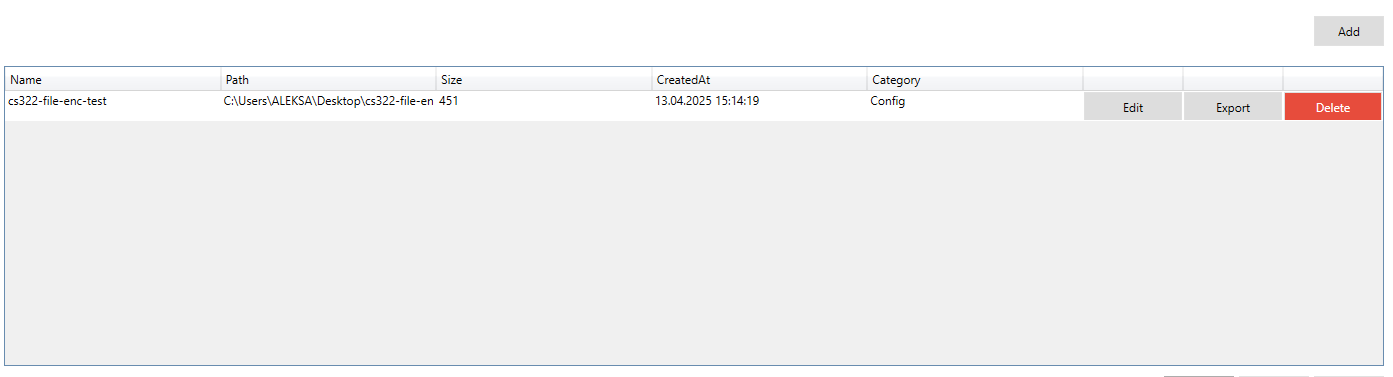
Ovo je sadržaj fajla koji želimo da enkriptujemo:



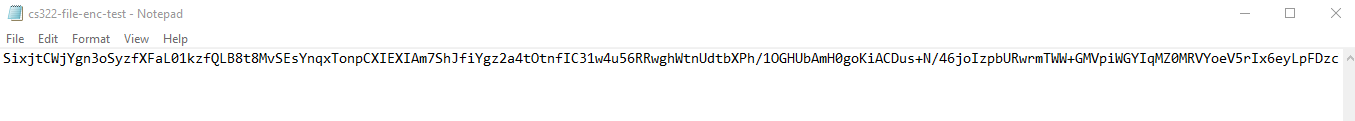
Prilikom prvog čuvanja fajla, generiše se random enkripcioni ključ koji se čuva u registry operativnog sistema „Windows“ na putanji „HKEY\_CURRENT\_USER\SOFTWARE\SystemVault“. Ključ se koristi za dalje enkriptovanje fajlova.



Nakon klika na taster „Save“, dobijamo poruku da je fajl uspešno enkriptovan i sada vidimo da se nalazi u tabeli.



Enkriptovan fajl koji smo kreirali:

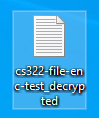


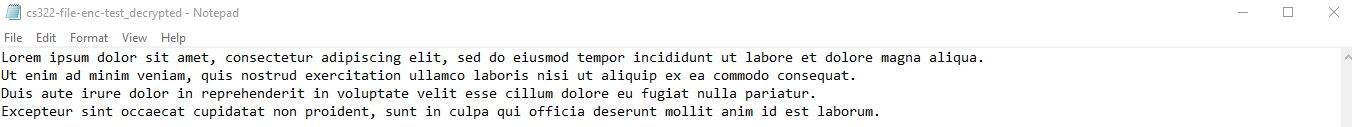
Dostupne su nam sledeće opcije:  
 - Edit – služi za izmenu podataka o fajlu

- Export – dekriptovanje fajla na desktop-u sa imenom [ime\_fajla]\_decrypted.

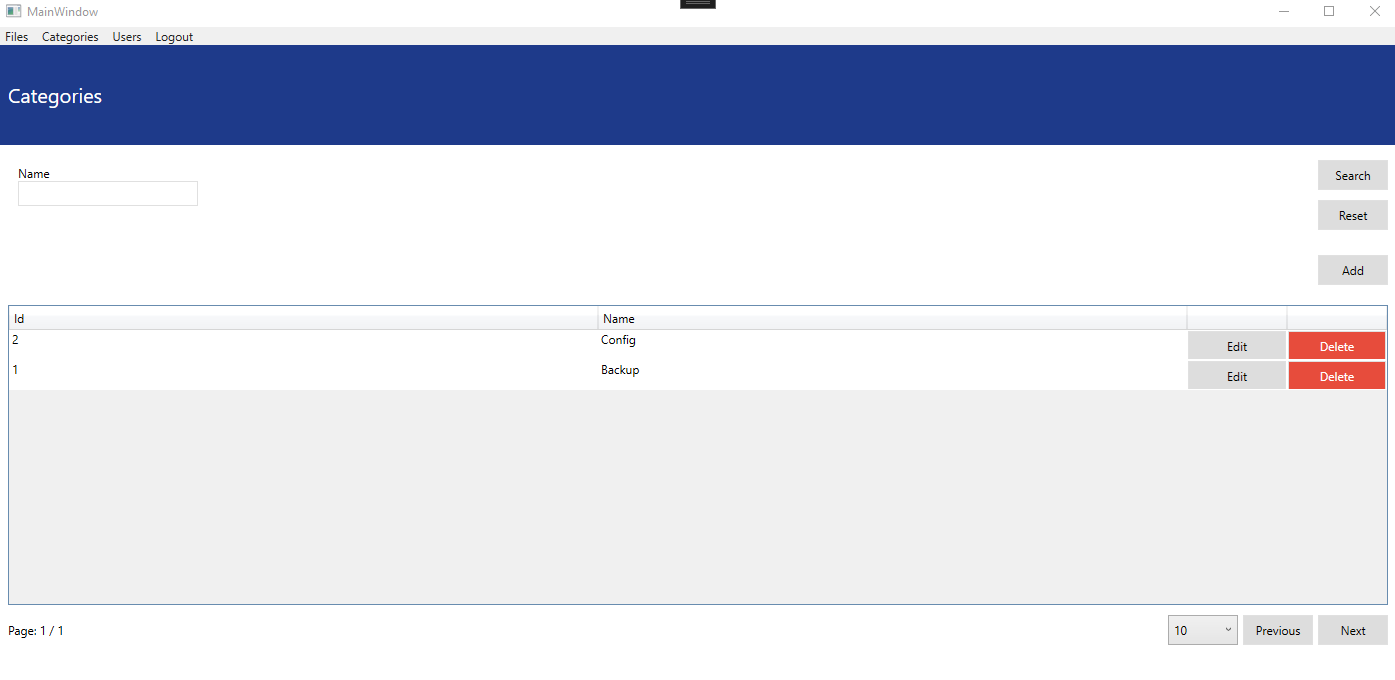
- Delete – brisanje fajla

Kada kliknemo na taster „Export“ dobijamo na desktop-u dekriptovan fajl koji ima isti sadržaj kao prvobitna verzija fajla:



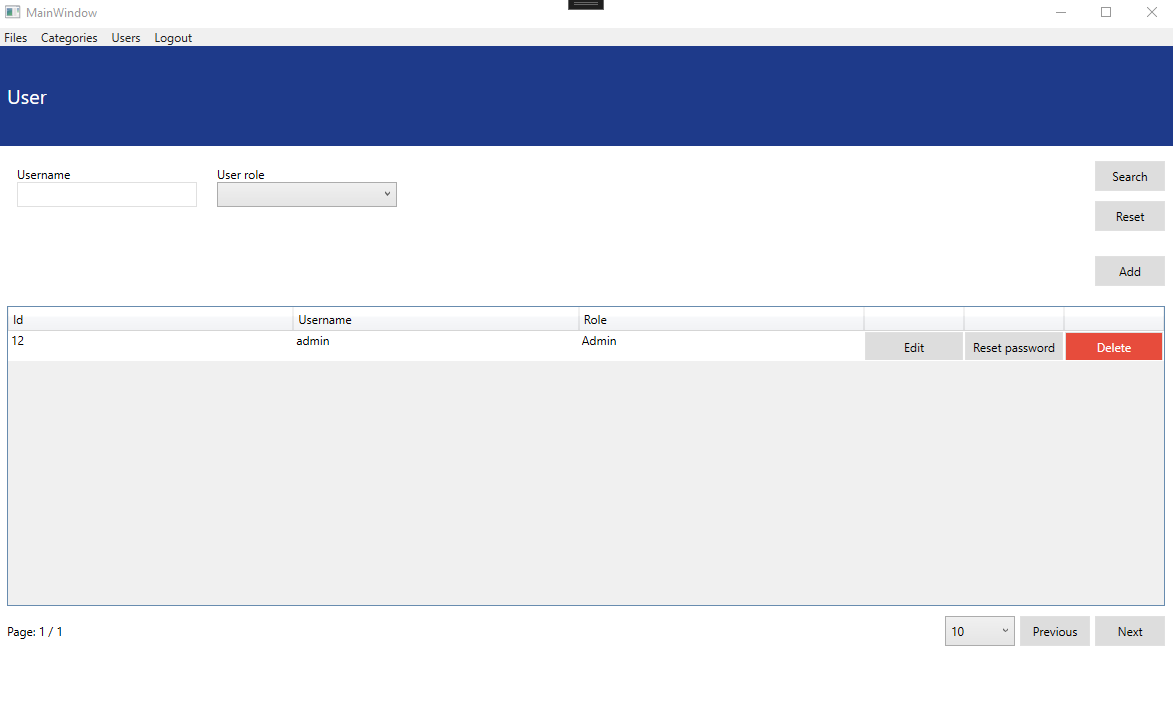


Na stranici „Categories“ imamo takođe prikaz tabele, pretraga po parametrima, dodavanje, izmena i brisanje kategorija.



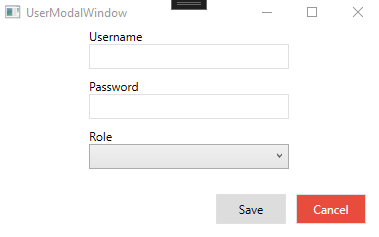
Ako želimo da dodamo kategoriju dovoljno je kliknuti na taster „Add“ i samo uneti ime kategorije koju želimo, takođe se ovo odnosi i na „Edit“ izmenu.

Admin uloga ima jednu dodatnu stranicu „Users“ gde moze upravljati korisnicima aplikacije. Imamo dve dostupne uloge u aplikaciji: Admin i Operator.

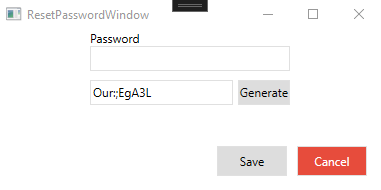


Može pretraživati, dodavati, menjati, resetovati lozinku, i brisati korisnike.

Klikom na taster „Add“ otvara nam se novi prozor za dodavanje korisnika:



Ako želimo da resetujemo lozinku korisniku, mozemo kliknuti na taster „Reset password“ gde nam se otvara novi prozor.



U input password mu unosimo lozinku, a ako ne želimo sami da smišljamo lozinku, mozemo je programski generisati.

Klikom na „Logout“ u menu bar-u, odjavljujemo se iz aplikacije.

Eksterne biblioteke koje aplikacija koristi:

- Entity Framework Core

- BCrypt.Net-Next

1. Zaključak

SystemVault pruža jednostavno i efikasno rešenje za lokalno skladištenje i zaštitu konfiguracionih fajlova. Implementacijom troslojne arhitekture, omogućeno je jasno razdvajanje odgovornosti i lakše održavanje koda. U budućnosti, aplikacija se može proširiti dodatnim bezbednosnim funkcionalnostima kao i sinkronizacijom preko mreže.

1. Literatura

<http://lams.metropolitan.ac.rs>

<https://www.google.com>

<https://stackoverflow.com>