**Обектно Ориентирано Програмиране(OOP with C#.NET)**

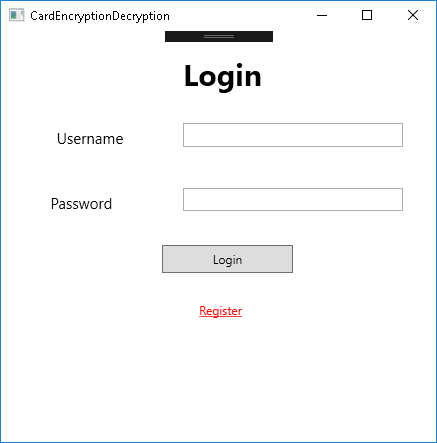
**Криптиране на банкови карти с SOAP Web service**

Изготвил: Даниел Светославов Пенчев СИ Курс2 62114

Приел: др. Евгений Кръстев

1. **Цел на проекта** – Разработване на система която използва SOAP уеб-услуга за криптиране , съответно декриптиране на банкови карти и на която потребителите могат да се доверят.
2. **Основни функционалности които системата предлага**
   1. **Влизане –** функционалност която позволява на регистриран потребител чрез въвеждане на личните си данни да влезе в системата, за да използва функционалностите към които има достъп

Фиг 1 – UI за вход в системата

****

* Потребителят въвежда потребителското име и парола
* При наличие на проблем с формата на потребителското име се появява tooltip със стандартите които трябва да бъдат спазени още докато го изписва потребителят .
* Потребителското име трябва :

-да започва с буква

- да се състои само от a-z, A-Z, 0-9, ‘-‘, ‘\_’ , ‘.’

- дължината му да е 8-25 символа

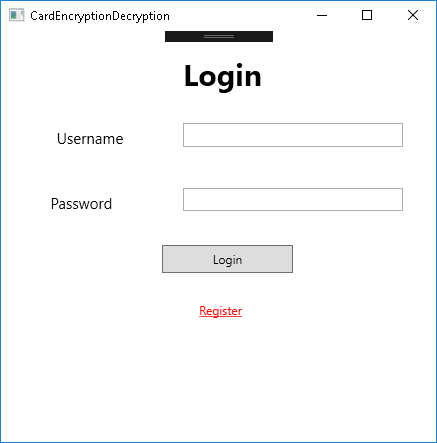
**- проблем** с PasswordBox - не може да се приложи ValidationRules

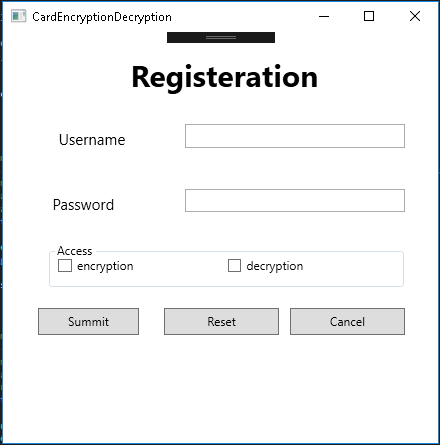
- След като бъде натиснат бутона Login – първо се проверява пак формата на потребителското име и на паролата, чак след това системата извиква услугата за вход в системата, където се проверява дали съществува такъв потребител с това потребителско име и парола, ако има такъв то услугата връща ид на текущата сесия, иначе се показва съобщение за неуспешно влизане. Системата качва на специално заделена памет всички детайли на потребителя от DB, и я освобождава при излизане от акаунта, като обновява детайлите на потребителя в базата от данни.

-**Проблем** с повече от едно устройства да използват един и същ акаунт – синхронизация

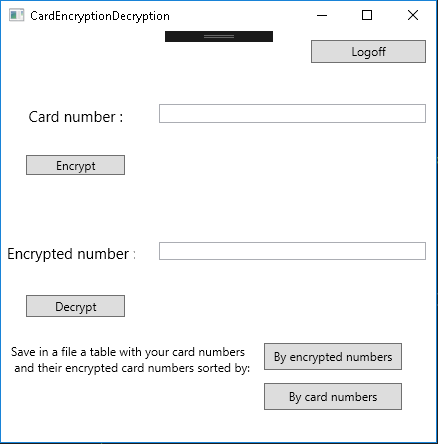
-**Решение**-да не може от няколко клиента да влизат с едно и също потребителско име по едно и също време

**2.2 Регистриране –** функционалност която позволява на нерегистриран потребител да се регистрира в системата, което ще му позволи да използва функционалностите за които има достъп.

****

****

* Потребителят както въвежда потребителско име и парола , така избира и ниво на достъп.
* Докато потребителят въвежда потребителското име както при формата за влизане така и тук програмата проверява за формата на му.
* Бутон “Reset” – изчиства формата за регистриране
* Бутон “Cancel”- връщане към формата за вход
* Бутон “Summit” – входните данни първо минават през проверка на формата. После се изпращат на сървъра предлагащ услугата на регистриране, където се проверява дали съществува вече регистриран потребител с това потребителско име.Ако не, тогава се създава нов акаунт на сървъра и се връщаме на формата за вход, ако съществува потребителят ще получи съобщение за грешка.

****

* 1. **Криптиране –** фунционалност , която позволява на регистрирания потребител да криптира номера на своята банкова карта

- потребителят въвежда номера на картата, прави се проверка на формата(правило на Luhn), ако отговаря на формата се изпраща на сървъра използвайки услугата за криптиране. Проверява се веднага достъпът до тази услуга(Понеже при вход в системата потребителят получава само ид на текущата сесия, и никаква друга допълнителна информация), ако има достъп сървърът проверява дали карта вече е била криптирана или е нова за системата

\*ако е нова, то сървъра запазва номера й в системата и криптира картата само чрез двойно транспониране с произволни 2 ключа от масив с ключове

\*ако пък не е , то картата не само че се криптира пак с двойно транспониране пак с различни кл. но и в зависимост от текущия номер на криптиране на картата, се използва substitution cipher

- Ако операцията не е успешна, ще се покаже съобщение за грешка

* 1. **Декриптиране –** Аналогично на процеса на криптиране, но в обратен ред
  2. **Създаване на файл, съдържащ таблица с банкови карти и техните криптирани номера –** функционалност която позволя на потребителя да си изведе таблица на банк. номера и техните крип. номера. Вече логнатият потребител може да избере между две опции, таблицата да бъде сортира по банк. номера или по крип. номера.Взимат се криптираните номерата на всички банкови карти и се декриптират като се конструира таблица от двойки банк карта – криптиран номер, и вече в зависимост от начинът на сортиране се записват във посочен от потребителя файл.
  3. **Излизане от системата –** функционалност която дава възможност потребителят да излезе от системата, с цел влизане отново в някои акаунт или създаване на такъв.(от гледна точка на производителността няма смисъл при всяка услуга да се взима съдържимото в DB и да се ъпдейтва, за това само при излизане от акаунта се прави това)

- Потребителят праща на сървъра ид на сесията чрез услугата за излизане от системата, като се проверява дали наистина има логнат такъв потребител.

- ако има то DB се ъпдейтва , ако ли не се извежда съобщение за грешка

**3. Проблеми от общ характер**

**-** За всеки нов хост трябва да се промени пътя до база данните, защото услугата работи с IIS express , а не като local host.

- LogOffBtn – трябва да е част от формата за влизане и излизане

- Хиперлинка за регистриране е труден за активиране

- Не се адаптира към промени на размера на формата

4. **Съхранение на информация**

- Използва се xml file с име DB,който се намира в папката CardEncryptionDecryptionService

**5. Използвана литература**

**-** How Security Can Help Grow Your Business: The Marketing Side of Tokenization

-H. M. Deitel, P. J. Deitel, “Visual C# 2012 How to Program”, 5th ed., Prentice Hall 2013, ISBN 0-13-337933-7 ISBN-13: 978-0-13-337933-4 (основна)

- Daniel Solis, “Illustrated C# 2012”, 4th ed. APress 2012, ISBN-13 (pbk): 978-1-4302-4278-9

- John Sharp, “Microsoft Visual C# 2012 Step By Step”, Microsoft Press, 2013 ISBN-13: 978-0735668010

Алгоритъми за криптиране :

* Двойна транспосизия - <http://en.wikipedia.org/wiki/Transposition_cipher>
* Проста субституция - <http://en.wikipedia.org/wiki/Substitution_cipher>

Проверка на формата на банковата карта :

* Алгоритъм на Luhn- <https://en.wikipedia.org/wiki/Luhn_algorithm>

Изисквания за потребителско имена и пароли :

* <https://support.kraken.com/hc/en-us/articles/201396497-What-are-the-Password-and-Username-format-requirements->