**КУРСОВ ПРОЕКТ**

**„Токенизация на банкови карти с многонишков сървър“**

**Изготвил: Стоян Стоянов; ф.н: 25016**

1.Описание на сървърното приложение – Системата има за цел да позволи на потребител да генерира токен за банкова карта, подавайки нейният номер или да получи номер след като бъде предоставен съответния токен. Системата се състои от две части. Първата част е приложение, коете се вписва в регистрите на компютъра на определен адрес и порт, на който втората част на системата или потребителското приложение ще изпраща заявки за обработка. След като се стратира сървъра, той автоматично регистрира три различни потребителя с различни права на достъп, запазвайки ги в xml файл с име users.xml, който се генерира в директорията на приложението. Потребителите са както следва:

* име: **admin**; парола: **12345** – има всички права
* име: **writer**; парола: **12345** – има право да генерира токен
* имe: **reader**; парола: **12345** – има право да получава номера на карти

След като бива стартиран, сървърът вече може да получава и да обработва заявки от клиентите, свързани към него.

Сървърат има три публични метода, които могат да бъдат извиквани:

* **login** – за вход в системата и оторизация на потребител, по заданеи username и password като параметри. След като бива извикан, този метод прочита xml файла –users и проверява дали има съществуващ потребител с това име и парола. Ако съществува –записва текущия потребител в паметта на сървъра и връща потребителското име към клиентското приложение или съответно съобщение за грешка, ако такова не е намерено.
* **saveToken** – приема като параметър номера на картата. Първо проверява дали текущия оторизиран потребител в системата има необходимите права ( ако те липсват връща подходяща греша на клиента), след което процедира към проверка на самия номер за валидност. Ако номера е валиден, генерира токен за него и записва двойката номер – токен в xml файл с име cards.xml в директорията на приложението. Накрая връща генерирания токен на клиента.
* **getCardNumber** – който приема подадения токен, след което проверява за необходими права. Ако няма необходимите права, уведомява клиентското приложение, в противен случай, прочита xml файла card.xml, проверява за наличнието на карта с такъв токен и ако намери връща този номер на клиента, а ако не - съобщение, че карта с такъв токен не е намерена.

Методите на това приложение се извикват посредством интерфейса, който наследява, но освен това съдържа и main метод, който при извикването си генерира swing компоненти, които служат като административен модул за генерирането на справки. Този модул съдържа два бутона за генериране на информация на банкови карти. Единия записва информацията във файл, като я сортира във възходящ ред по номер на карта, а другия – по токен.

2. Описание на потребителския интерфейс – втората част от сисемата е

потребителския интерфейс, който изпраща заявки към сървъра. При стартиране на приложението, то проверява в регистрите за сървър, към който да изпраща заявки на съответен адрес и порт. След като го намери, го зарежда в паметта, като вече има достъп до неговите методи посредством интерфейса на сървъра.

Първоначално се появява форма, която изисква оторизация чрез въвеждане на потребителско име и парола. След като се въведат и се натисне Enter или бутон **Log in.** Въведените данни се изпращат на сървъра за валидация. Ако данните са коректни, се зарежда вторни панел, който осигуравя достъп до услугите, които приложенито предоставя, ако името или паролата са грешни, изкача нов прозорец със съобщението за грешка.

След като се зареди втория панел, потребителят вижда голямо текстово поле, в чиято горна част се появява съобщението “Hello, ” и оторизираното потребителско име. Отдолу има две полета – едното за въвеждане на номер на карта чрез който да се генерира токен от съръва, а другото за въвеждане на токен и за получаване на записан номер на карта.

След като се въведене номера на карта в първото поле и се натисне Enter докато полето е маркирано или се кликне върку бутона “**Generate**”, се прихваща съответния евент, който проверява дали номерът се състой само от цифри и дали е подходящата дължина, използвайки регулярен израз. Ако валидацията е успешна, номерът се изпраща на сървъра. След като се получи отговор, се проврява неговото съдържание и ако то съдържа токен, то той се визуализира в текстовото поле отгоре, ако ли пък е грешка, изкача прозорче, което я визуализира (за невалиден номер на карта, за липса на права и пр).

Аналогично, при второто поле се изпраща токен, валидира се и се връща номер на карта или грешка от сървъра.

3. Използвани структури и алгорими

* **Card** – структура, за запазване на номера и токена на банковата карта, чрез която се сериализират/десериализират обектите при записване/четене на банкове карти от xml файловете
* **User** – аналогично като структурата на картата, се използва съхранение на името и паролата на потребителите
* UserRights – енумерация, която съдържа различните права за достъп
* **SortBy** – енумерация за определяне на полето по което ще се сортират банковите карти при извеждането на справки от административния модул – по номер или токен
* За сериализация и десериализация на xml файловете се използва **XStream-1.4.8**

4.Тестване - Данните използване за тестване на приложението са всички **16-цифрени** номера от този [адрес](https://www.paypalobjects.com/en_US/vhelp/paypalmanager_help/credit_card_numbers.htm) и съответно получените токени за получаване на номерата обратно.