



Софийски Университет „Св. Климент Охридски“

Факултет по математика и информатика

Курсов проект

по Обектно-ориентирано програмиране с Java

Криптиране на банкови карти с RMI

Изготвил: Невена Гаджева, фак.№: 61938, курс 3, Софтуерно инженерство

Ръководител: д-р Евгений Кръстев

Дата: 16.02.2018г.

1. Цел на проекта

Целта на проекта е да се реализира потребителско клиент-сътвър приложение за криптиране на банкови карти. Потребителите на системата имат възможност да криптират номер на банкова карта и да извличат номер на карта по криптограма, като за целта трябва да имат съответните права на достъп и да са се вписали в системата. Сървърната част реализира съответно частта с валидирането и верифицирането на данните на потребителите, извършва необходимите изчисления и извежда получените резултати на потребителя. Допълнително позволява да се съхранява информация за потребители и техните права в XML файл, както и да извежда текстов файл с таблица на криптираните карти, сортирани по определен признак.

2. Архитектура

- **Сървърна част**

- *JavaFXML пакет ServerGUI* – съдържа графичния интерфейс на сървъра. Състои се от следните класове и файлове:

- *ServerGUIFXMLDocument.fxml* – fxml файл, съдържащ XML описание на графичния интерфейс на сървърната част на приложението.
- *ServerGUIController.java* – класът *ServerInterfaceController* представлява контролер на графичния интерфейс на сървъра. Предоставя интерактивен интерфейс, позволяващ извеждане на log-ове за системата, записване на данните на потребителите в XML файл и извеждане на текстов файл с таблица на криптираните номера и съответните им банкови карти, сортирана по криптираните номера или по банковите карти.
- *Interface ServerInterface.java* - ...
- *ServerInterfaceImpl.java* - ...
- *ServerGUI.java* – класът *ServerInterface* зарежда графичния интерфейс на сървъра. Това е изпълнимият файл, от който се стартира сървъра.

- **Клиентска част**

- *JavaFXML пакет Interface* – графичен потребителски интерфейс, който се използва многократно като Jar файл.
- *JavaXML пакет UserInterface* – използва за основа за изграждане на потребителския интерфейс пакета *Interface*. Състои се от следните класове и файлове:
 - *LoginGUIFXMLDocument.fxml* – fxml файл, съдържащ XML описание на графичния интерфейс на Login формата на приложението (стартовата точка, от която започва да се изпълнява клиентската част на приложението).
 - *AdminGUIFXMLDocument.fxml* – fxml файл, съдържащ XML описание на графичния интерфейс за администратора. Позволява добавяне на нов потребител, заедно с парола и права на достъп.
 - *UserGUIFXMLDocument.fxml* – fxml файл, съдържащ XML описание на графичния интерфейс за клиента. Позволява криптиране на карта и извличане на номер на карта.

- *UserGUIController.java* – класът *UserInterfaceController* представлява контролер на графичния интерфейс на клиентската част на приложението, която включва Login формата и формата за администратора. Предоставя интерактивност на графичния интерфейс и коректно визуализира резултати и грешки.
- *UserController.java* – класът *UserInterfaceController* представлява контролер на графичния интерфейс на клиентската част на приложението, която включва формата за криптиране/декриптиране на банкови карти. Предоставя интерактивност на графичния интерфейс и коректно визуализира резултати и грешки.
- *UserGUI.java* – класът *UserInterface* е изпълнимият файл, от който се стартира клиентската част на приложението. Първоначално се зарежда Login формата, т.е. файлът *LoginInterfaceFXMLDocument.fxml*.

- **Помощен пакет**

- *Пакет CourseProjectNo4* – съдържа класовете за валидиране, криптиране и декриптиране на банкови карти, както и клас, представящ информация за отделния потребител.
 - *Enumeration class UserType* – изборим тип, представящ видовете потребители и техните нива на достъп – администратор(1), обикновен потребител(0) и гост(-1).
 - *Клас User* – представя информация за отделния потребител – потребителско име, парола, права (ниво на достъп), банкови карти, заедно с техните криптограми.
 - *Enumeration class CardPrefixes* – изброим тип, представящ валидни префикси на банкови карти – 4 за Visa, 5 за Master card, 6 за Discover card и 37 за American Express.
 - *Клас ValidateCard* – служи за валидиране на банкова карта по формулата на Luhn.
 - *Клас CardEncryption* – реализира криптиране на банкова карта по алгоритъма Route Cipher.
 - *BankCardsSortedByCardNumber.java* – класът записва банковите карти и техните криптограми във файл „sortedByCardNumber.txt“ под формата на таблица, сортирана по номерата на банковите карти, след което ги прочита.
 - *BankCardsSortedByEncryption.java* – класът записва банковите карти и техните криптограми във файл „sortedByEncryption.txt“ под формата на таблица, сортирана по криптограмите, след което ги прочита.
 - *UsersInfoXML.java* – класът създава XML файл и записва информацията за потребителите в него (потребителско име, парола и права на достъп).

3. Използвани структури данни и алгоритми

- **Алгоритъм Route cipher**

- *Криптиране*

Номерът на банковата карта се записва в двумерна матрица. Ключът за криптиране е 5, като при всяко следващо криптиране на банковата карта се инкрементира с 1, след което се дели по модул 16. Той определя колко колони ще има матрицата. Редовете се определят, като дължината на номера на банковата карта се раздели на броя на получените колони. След като се създаде матрицата, всяка цифра от номера на банковата карта се записва в получената матрица, спираловидно, започвайки от горния ляв ъгъл обратно на часовниковата стрелка.

- **Алгоритъм на Luhn**

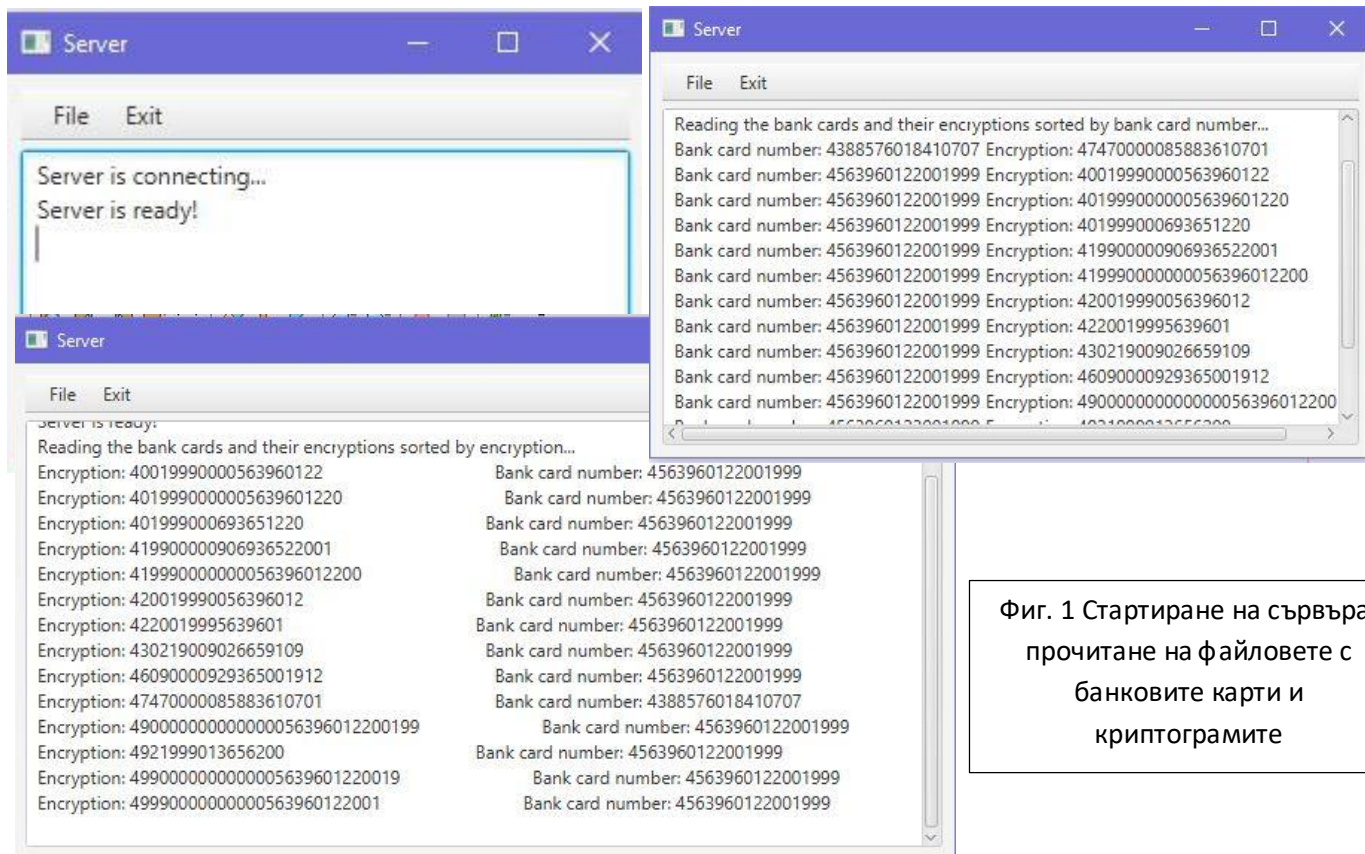
- От дясно на ляво всяка втора цифра на банковата карта се удвоява. Ако резултатът от удвояването на всяка цифра е по-голям от 9, то се изважда 9 от полученото число.
- Събират се всички цифри на банковата карта, включително и новополучените.
- Ако полученият резултат, разделен по модул 10, е равен на 0, то номерът на банковата карта е валиден спрямо формулата на Лу. В противен случай не е.

4. Възникнали и текущи проблеми

- По време на имплементацията на приложението възникна проблем със свързването на клиента със сървъра и предаването на данни от една JavaFX FXML сцена на друга.

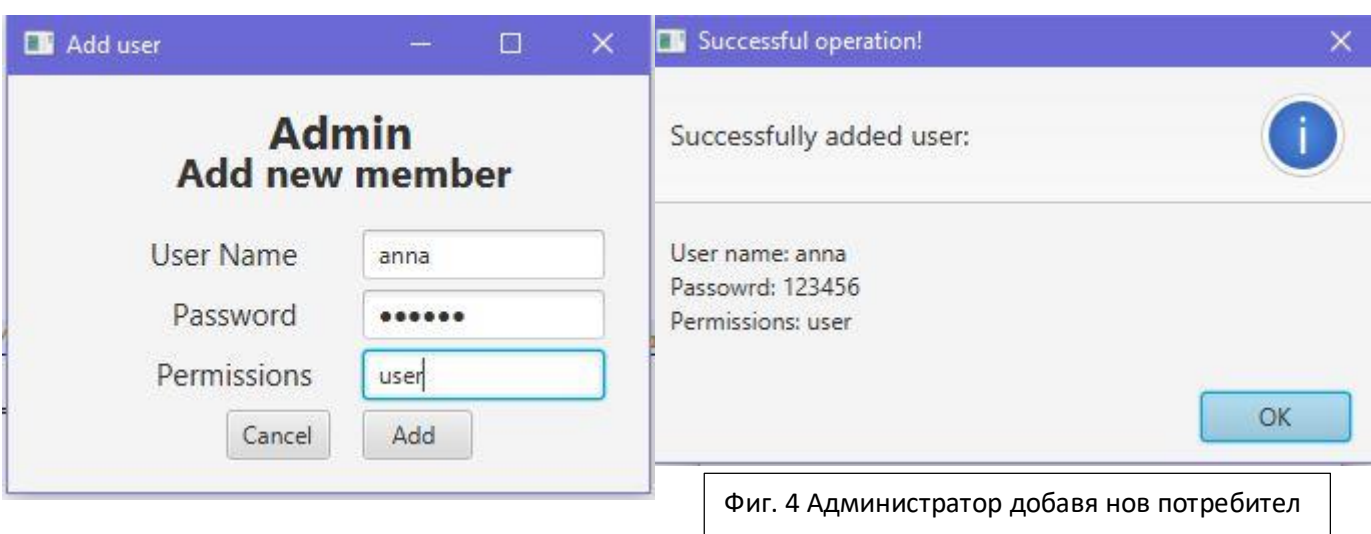
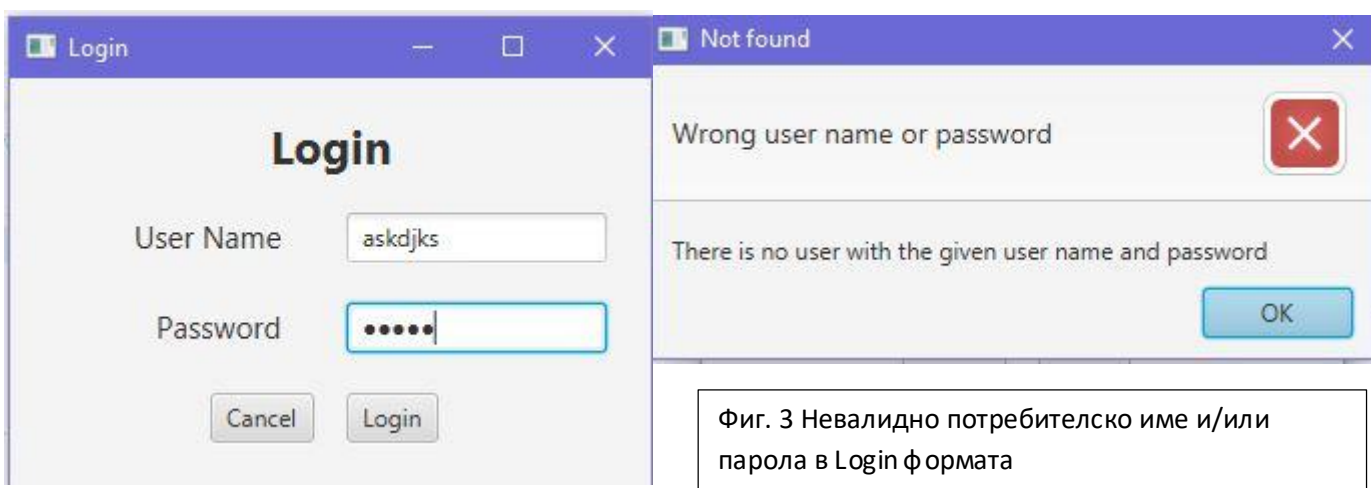
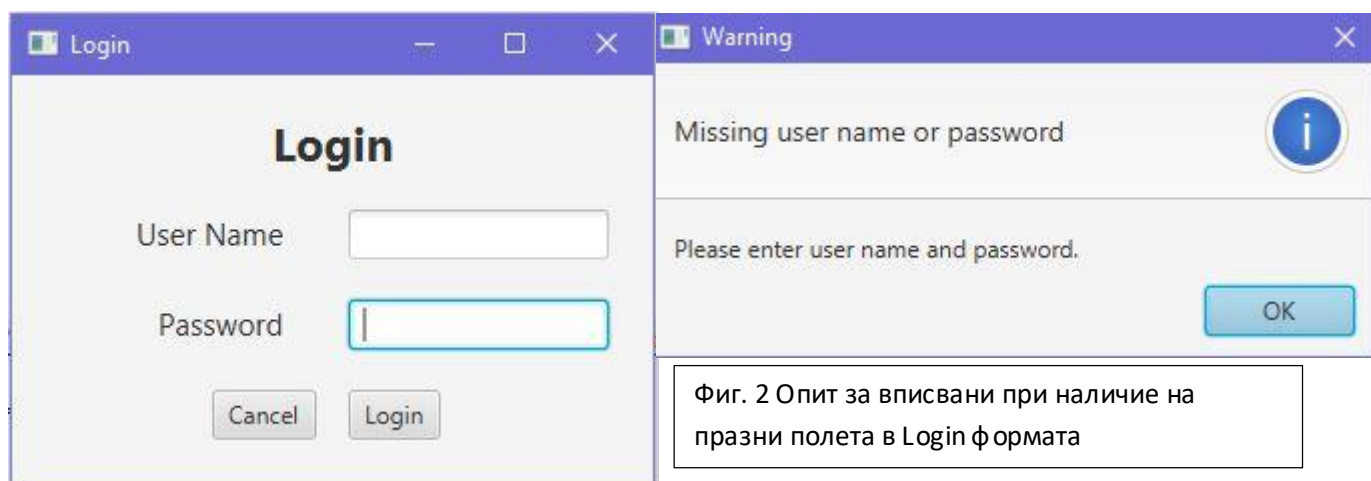
5. Графичен интерфейс

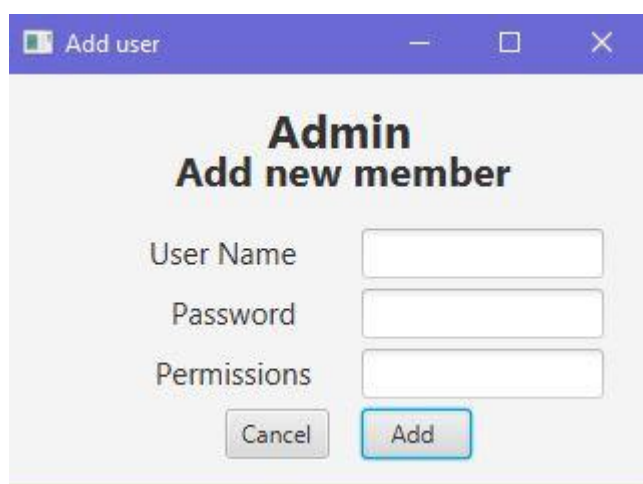
- **Сървър**



Фиг. 1 Стартиране на сървъра, прочитане на файловете с банковите карти и криптограмите

- **Клиент**






Add user

Admin
Add new member

User Name

Password

Permissions

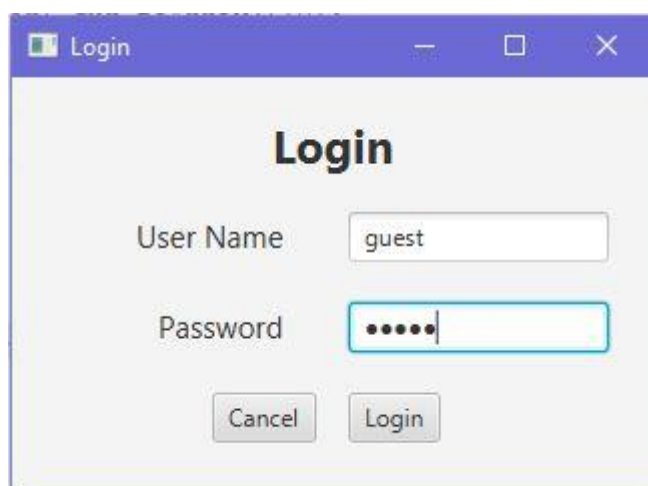


Warning

Missing user name, password or permissions or wrong permissions.

Please input user name, password and permissions. The permissions should be one of the following: admin, user or guest.

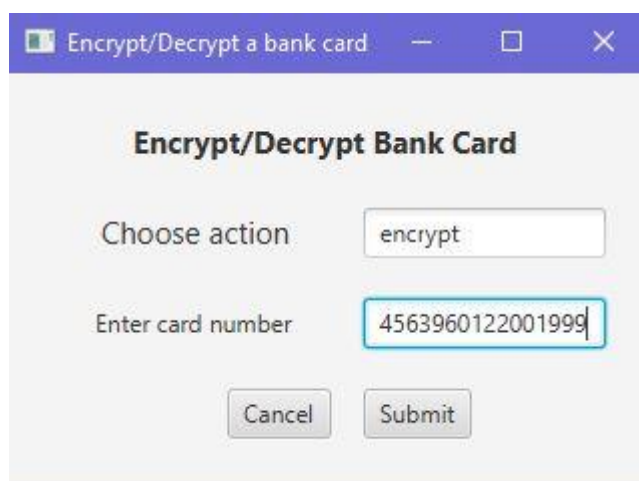
Фиг. 5 Опит за добавяне на нов потребител при празни полета



Login

User Name

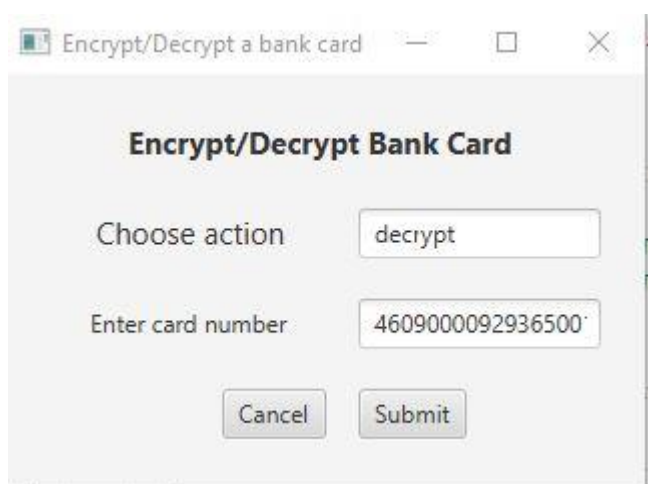
Password



Encrypt/Decrypt a bank card

Choose action

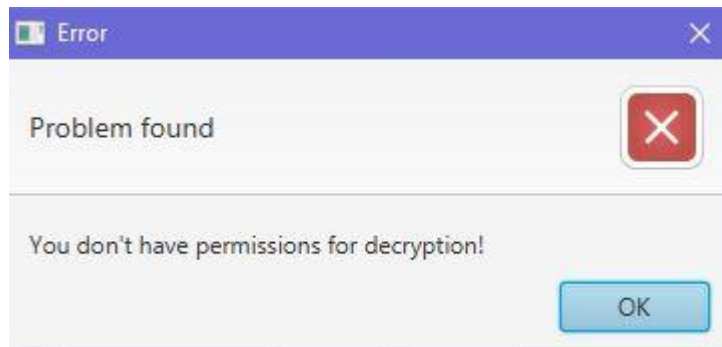
Enter card number



Encrypt/Decrypt Bank Card

Choose action

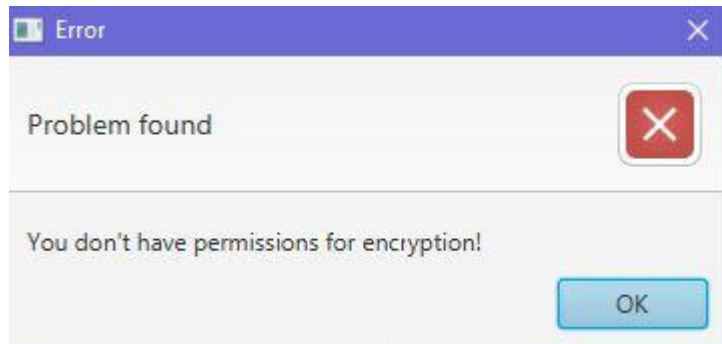
Enter card number



Error

Problem found

You don't have permissions for decryption!



Error

Problem found

You don't have permissions for encryption!

Фиг. 6 Опит за криптиране/декриптиране на гост

Encrypt/Decrypt Bank Card

Choose action

Enter card number

Wrong operation

Problem occurred.

The operation you entered is incorrect!
Please, try again!

Фиг. 7 Опит за криптиране/декриптиране при наличие на празни полета

Encrypt/Decrypt Bank Card

Choose action

Enter card number

Encryption

The encryption code for your bank card is:

Bank card number: 4563960122001999
Encryption code: 46090000929365001912

Фиг. 8 Криптиране на банкова карта

Encrypt/Decrypt Bank Card

Choose action

Enter card number

Фиг. 9 Опит за криптиране на невалидна банкова карта

Warning

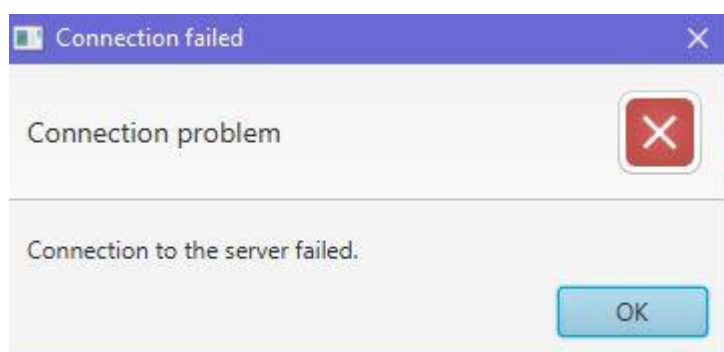
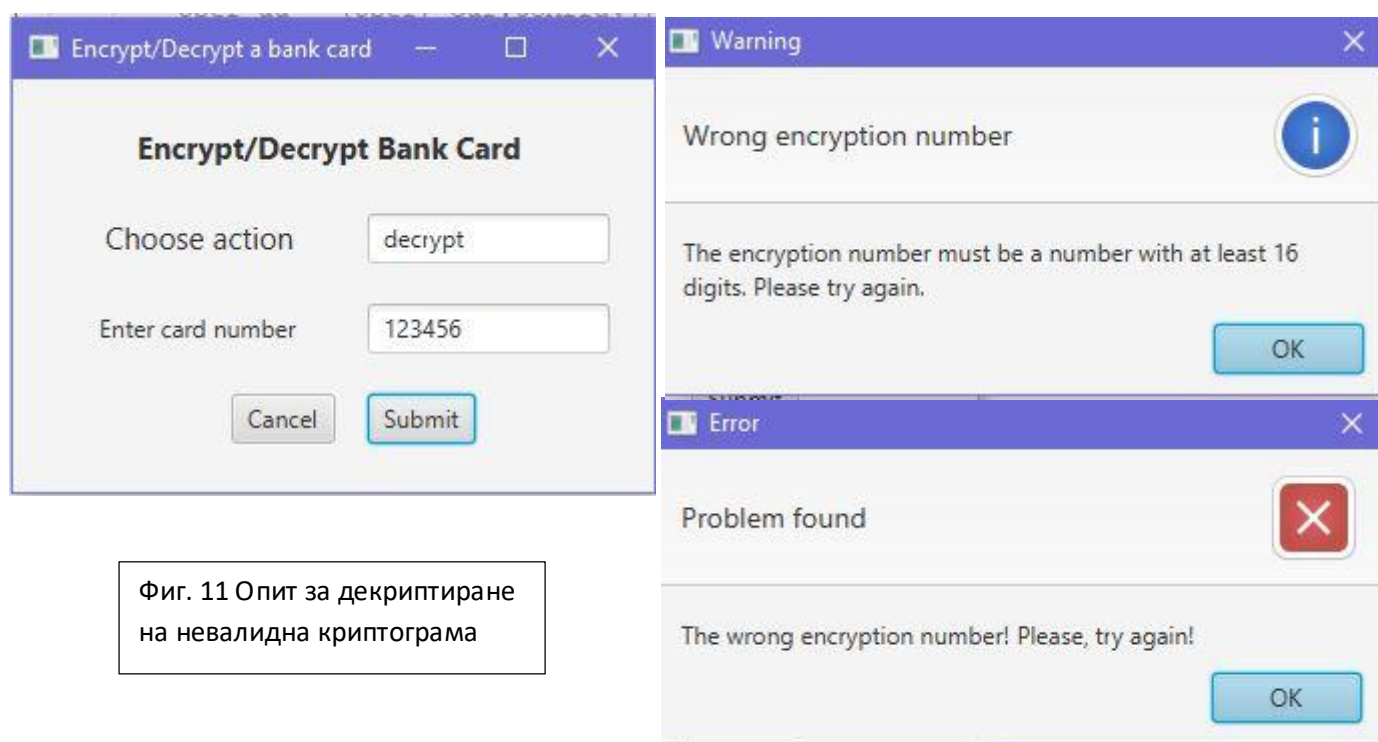
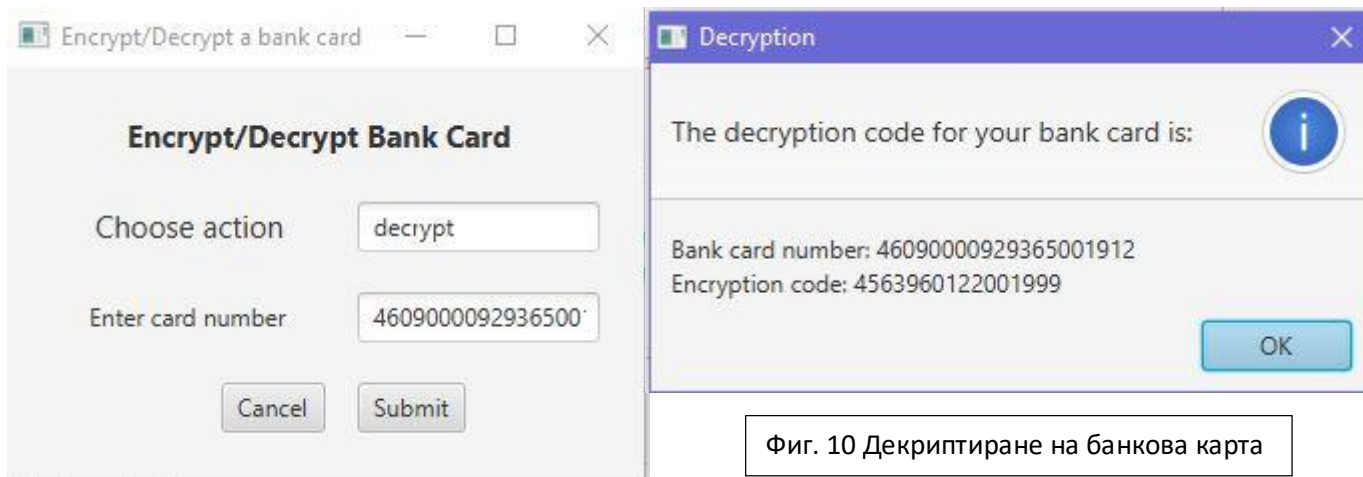
Wrong card number

The card number must be a number with 16 digits. Please try again.

Error

Problem found

The bank card is invalid! Please, try again



6. Тестване на приложението

Проектът е тестван с първоначално въведени 3 основни потребители, покриващи трите групи потребители в приложението, а именно администратор (admin), обикновен потребител (user) и гост (guest), както и 2 банкови карти със съответстващите им криптограми:

- | Банкова карта | Криптограма при стандартно отместване 5 |
|--------------------|---|
| • 4563960122001999 | – 46090000929365001912 |
| • 4388576018410707 | – 47470000085883610701 |

Всички възможни получени криптограми на банковата карта с номер 4563960122001999 са записани във файловете sortedByCardNumber.txt и sortedByEncryption.txt, съдържащи таблиците със сортираните банкови карти. Добавен е и нов потребител anna с парола 123456 и права user. Данните за потребителите са записани в XML файл usersInfo.xml. Съответните файлове се намират в папката ServerGUI.

7. Използвана литература

- **Евгений Кръстев**, Lecture_10.3-FXShot.pdf, Lecture11c.pdf, Lecture14aFX.pdf, Lecture14bFX.pdf
- **Bruce Eckel**, "Да мислим на Java", SoftPress, 2001
- **Алгоритъм Transposition cipher** - https://en.wikipedia.org/wiki/Transposition_cipher
- **Алгоритъм Route cipher** - <http://crypto.interactive-maths.com/route-cipher.html>
- **Алгоритъм на Luhn** - https://en.wikipedia.org/wiki/Luhn_algorithm