



# Banking system

Да се реализира система имплементираща основни банкови функционалности. Системата ще поддържа следните типове потребители:

## 1. Клиент

Клиентът е потребител на приложението, който може да има различни сметки, отворени в различни банки.

За потребител, който е клиент, системата поддържа следния интерфейс:

- `check_avl [bank_name] [account_number]` - Извежда наличността в сметка с номер `[account_number]` в банка с име `[bank_name]`. Ако такава сметка или такава банка не съществуват, да се изведе подходяща грешка.
- `open [bank_name]` - Изпраща заявка към банката за отваряне на нова сметка. В заявката, която се изпраща към банката, автоматично се изпращат и неговите имена, ЕГН и възраст (потребителя не ги въвежда ръчно).
- `close [bank_name] [account_number]` - Изпраща заявка към банката за затваряне на сметка. При невалидни данни да се изведе подходяща грешка.
- `redeem [bank_name] [account_number] [verification_code]` - Осребрява чек, получен от трето лице чрез банката. Парите от чека се прехвърлят в подадената сметка при валидиран код. Ако такъв чек не съществува се извежда подходящо съобщение за грешка.
- `change [new_bank_name] [current_bank_name] [account_number]` - Изпраща заявка към банката за преместване на подадена сметка към друга банка.
- `list [bank_name]` - Извежда всички сметки, които потребителя притежава в банката.
- `messages` - Показва всички съобщения в хронологичен ред.

## 2. Служител на банка

Служителят работи по задачи, които са му дадени от банката. Банката генерира задачи на служителите си, когато потребител изпрати заявка за отваряне на сметка, преместване на сметка или затваряне на сметка. Банката знае за своите служители и при получаване на заявка от потребителя създава нова задача на най-малко заетия си служител.

Служителя изпълнява задачата си и въвежда резултата в банковата система. Задачите могат да имат следните резултати:

- `Open заявка` - Служителят може да прегледа личните данни на потребителя, който иска да създаде сметка. Служителят може да реши дали да позволи създаването или не. Ако реши да не го позволи, той трябва да изпрати съобщение към клиента.





- Close заявка - Аналогично на Open заявката, служителят може да я приеме или откаже. При отказ се изпраща съобщение от служителя.
- Change заявка - Служителят трябва да провери дали наистина такава сметка съществува в старата банка на клиента и дали данните съответстват. Ако служителят не направи такава проверка, той не може да одобри смяната. Ако направи такава проверка и валидацията е успешна, служителят може да продължи с одобряване на заявката. Дори валидацията да е успешна служителят, все пак може да не разреши преместването.

При завършена задача от служителя потребителят, който чака резултат от тази задача, получава съобщение. Ако задачата е успешна, съобщението е по подразбиране. Ако е неуспешна, съобщението изпратено от служителя, се изпраща на потребителя. Във всеки един случай потребителят знае името на служителя, който е разгледал задачата му (изпраща се като част от отговора)

За потребител, който е служител, системата поддържа следния интерфейс:

- tasks - показва задачите, които трябва да свърши. Всяка задача има тип, име и индекс.
- view [task\_id] - Показва всички подробности около задачата.
- approve [task\_id] - Служителят приключва задачата и одобрява заявката.
- disapprove [task\_id] [message] - Служителят приключва задачата и отказва заявката.
- validate [task\_id] - Служителят от новата банка прави заявка към старата банка, очакваща валидация дали потребителя е въвел коректни данни. Тази валидация не минава през служител на старата банка, а се връща на момента. Могат да се валидират единствено Change задачи.

### 3. Служител на външна фирма.

Служителят на фирма е потребител на системата, който е отговорен за изпращане на чекове към клиенти на банката. Чекът трябва да съдържа:

- Уникален таен код от три символа състоящ се от цифри, малки и главни латински букви. (пример: agD, SIY, b5L).
- Сума за превод.

Както бе споменато при разглеждане на интерфейса на клиента, чековете могат да се осребряват с командата redeem. Клиентите трябва да получат съобщение с уникалния си код.

За потребител, който е служител на външна фирма, системата поддържа следния интерфейс:

- send\_check [sum] [verification\_code] [egn] - Създава чек за потребител с посочено егн.

### 4. Представяне на потребителите





Всеки един потребител на системата се описва със следните характеристики:

- Собствено и фамилно име.
- ЕГН
- Възраст

Понеже личната информация на клиентите се разглежда от служителите на банката, клиентите съдържат още адрес и запис на чековете, които е осребрил.

Всички потребители поддържат следния интерфейс:

- exit - излиза от профила.
- whoami - показва пълна информация за текущия потребител.
- help - показва помощно меню за поддържаните команди.

## 5. Системни изисквания

Системата трябва да поддържа следния интерфейс:

- login - Вписва потребител в системата.
- signup - създава нов потребител в системата. При регистрацията можем да се регистрираме като всеки от трите типа служители.
- create\_bank - създава нова банка в системата. За да се създаде банка не ни е нужен акаунт. За да се създаде служител на банка банката трябва да съществува.





## 6. Примерен интерфейс на приложението

`create_bank` EntranceInvest

`create_bank` Health\_Savings\_Accounts

### `signup`

Name: Stancho

EGN: 0xxxxxxxx1

Age: 23

Role: Client

Password: \*

### `signup`

Name: Yoncho

EGN: 0xxxxxxxx2

Age: 23

Role: Employee

Bank associated: EntranceInvest

Password: \*

### `signup`

Name: Plámena

EGN: 0xxxxxxxx3

Age: 19

Role: Third-party employee

Password: \*

### `signup`

Name: Nikola

EGN: 0xxxxxxxx4

Age: 29

Bank associated: Health\_Savings\_Accounts

Role: Employee

Password: \*

### `login`

Name: Plámena

Password: \*





whoami

You are Plámena!

send\_check 150.00 agd 0xxxxxxx1  
exit

login

Name: Stancho

Password: \*

open EntranceInvest  
exit

login

Name: Yoncho

Password: \*

tasks

[1] Open - Stancho wants to create an account.

view 1

Open request from:

Name: Stancho

EGN: 0xxxxxxx1

Age: 23

approve 1  
exit

login:

Name: Stancho

Password: \*

messages

[1] - You have a check assigned to you by Plámena.

[2] - You opened an account in EntranceInvest! Your account id is 777.

list EntranceInvest

\* 777





check\_avl EntranceInvest 777

0\$

redeem EntranceInvest 777 agd

check\_avl EntranceInvest 777

150\$

change Health\_Savings\_Accounts EntranceInvest 777

exit

login

Name: Nikola

Password: \*

tasks

[1] Change - Stancho wants to join Health\_Savings\_Accounts.

view 1

Open request from:

Name: Stancho

EGN: 0xxxxxxxx1

Age: 23

Bank: EntranceInvest

approve 1

Cannot proceed - please make sure Stancho is real user by asking the bank!

validate 1

exit

login

Name: Yoncho

Password: \*

tasks

[2] Change - Health\_Savings\_Accounts asks if Stancho is real user.





view 2

Open request from:

Name: Stancho

EGN: 0xxxxxxxx1

Age: 23

Account number: 777

Balance: 150\$

approve 2

exit

login

Name: Nikola

Password: \*

tasks

[1] Change (approved) - Stancho wants to join Health\_Savings\_Accounts.

approve 1

exit

login

Name: Stancho

Password: \*

messages

[1] - You have a check assigned to you by Plámena.

[2] - You opened an account in EntranceInvest! Your account id is 777.

[3] - You changed your savings account to Health\_Savings\_Accounts. New account id is 2903.

list EntranceInvest

list Health\_Savings\_Accounts

\* 2903

check\_avl 2903

150\$

close Health\_Savings\_Accounts 2903

exit





### login

Name: Nikola

Password: \*

### tasks

[2] Close - Stancho wants to **close** account with id 2903.

**view** 2

Close request from:

Name: Stancho

EGN: 0xxxxxxxx1

Age: 23

Bank: Health\_Savings\_Accounts

Account number: 2903

Balance: 150\$

**disapprove** 2 "Why don't you spend your money?"

**exit**

### login

Name: Stancho

Password: \*

### messages

[1] - You have a check assigned to you by Plámena.

[2] - You opened an account in EntranceInvest! Your account id is 777.

[3] - You changed your savings account to Health\_Savings\_Accounts. New account id is 2903.

[4] - Your Close request was not approved. Reason: Why don't you spend your money?

**exit**

**exit**

