

Szakképesítés megnevezése: **Webfejlesztő**

Képzés azonosítója: **06135001**

VIZSGAREMEK

X Bank - netbank alkalmazás

Készítette: Verovszki Tamás

Konzulensek: Budaházi Bence
Czinkóczy Tamás
Halász Gábor

Budapest, 2022

NYILATKOZAT

Alulírott Verovszki Tamás kijelentem, hogy ez a vizsgaremek saját tudásom, önálló munkám terméke.

Budapest, 2022. május 2.



.....

aláírás

Tartalomjegyzék

1. Bevezetés.....	5
2. A témaválasztás indoklása.....	6
3. Fejlesztői dokumentáció.....	7
3.1. Backend oldal	7
3.1.1 Az alkalmazott fejlesztői eszközök	7
3.1.2 Az adatmodellek leírása	8
3.1.3 Részletes feladat-specifikáció, algoritmusok.....	11
3.1.4 A program telepítése.....	23
3.1.5 Tesztelési dokumentáció	24
3.1.6 Továbbfejlesztési lehetőségek	26
3.2. Frontend oldal	26
3.2.1 Az alkalmazott fejlesztői eszközök	26
3.2.2 Részletes feladat-specifikáció, algoritmusok.....	27
3.2.3 A program telepítése.....	30
3.2.4 Tesztelési dokumentáció	31
3.2.5 Továbbfejlesztési lehetőségek	34
4. Felhasználói dokumentáció	35
4.1 A program általános specifikációja.....	35
4.2 Rendszerkövetelmények	35
4.3 A program telepítése és indítása.....	36
4.4 A program használata.....	36
4.4.1 Bejelentkezés nélkül használható rész	36
4.4.2 Bejelentkezéshez kötött rész.....	38
5. Összegzés.....	46
Irodalomjegyzék, forrásjegyzék.....	47
Ábrajegyzék	48

1. Bevezetés

A vizsgaremekem egy képzeletbeli bank - az X Bank Limited - netbank alkalmazása. Az alkalmazás teljes funkcionalitással rendelkezik, de értelemszerűen nem valós adatokat tartalmaz, így értelemszerűen nem használható valódi pénz küldésére vagy fogadására.

Az alkalmazás két fő részből áll. Az egyik része bejelentkezés nélkül is használható. Itt olyan általános adatokat és információkat találhatunk meg, melyek függetlenek az ügyfelektől. Ilyenek például a Bank bankfiókjainak és bankautomatáinak az elérhetőségei, a valuta-és deviza árfolyamok, vagy a bankkal történő kapcsolatba lépési lehetőségek. Az alkalmazás másik része bejelentkezéshez kötött, ugyanis itt már olyan személyes információk válnak hozzáférhetővé, mint a számlaegyenleg, a számlatörténet vagy a bankszámla kivonatok. Itt lehet továbbá banki műveleteket is végezni, itt kezelhetjük bankkártyáinkat és itt tudunk jelszót változtatni.

A kezdetek során az volt az elképzelés, hogy az alkalmazásom mindent tudjon, amit egy valamire való netbank alkalmazás tudhat. Azonban hamar rájöttem, hogy ez egy akkora méretű feladat lenne, ami sem időben, sem terjedelemben nem férne bele a vizsgaremek követelményei által megszabott keretbe. Így kompromisszumokat kötnöm és kihagytam olyan programrészeket és szolgáltatásokat, melyekkel a mindennapi használat során ritkán találkozik egy átlagfelhasználó. Ezek a kihagyott részek azonban talonban maradnak, és a későbbiek folyamán tegy esetleges továbbfejlesztés alapját fogják képezni.

Az alkalmazás fejlesztése során volt egy fontos szempont, nevezetesen az, hogy az elkészült alkalmazás a lehető legnagyobb mértékben felhasználóbarát legyen. Meggyőződésem, hogy egy alkalmazás tervezése és elkészítése során ez egy olyan fontos kritérium, melyre fokozottan oda kell figyelni. Manapság hatalmas választék van a szolgáltatások minden területén, és a felhasználók könnyen tovább állnak, ha az általuk használt alkalmazás vagy szolgáltatás nem elégíti ki igényüket a gyors és kényelmes használhatóság terén.

2. A témaválasztás indoklása

A bankok sokféle szolgáltatással és kedvezményekkel csábítják magukhoz az ügyfeleket, azonban ami az egyik bank kínálatában szerepel, az nem biztos, hogy a másikonál is megtalálható. Vannak testre szabható lakossági banki szolgáltatások is, de azok sem tartalmazznak minden lehetséges funkciót. Ezek miatt életem folyamán számtalan banknál volt folyószámlám, sokszor több banknál is egyszerre. Igyekeztem kihasználni a különböző banki szolgáltatások közül a számomra előnyösebbeket. Mivel napjainkban az összes bank nyújt internetes bankolási felületet az ügyfelek részére, így sikerült az évek során sokféle netbank és mobilbank alkalmazással megismerkednem. Aktív bankszámlahasználatom gyorsan elvezetett odáig, hogy egy átlag felhasználónál jóval mélyebben megismertem az bankok elektronikus felületű szolgáltatásait. Észrevettem hiányosságokat és hibákat is, valamint születtek olyan ötleteim, melyek megítésem szerint könnyebbé tehetnék mind a bankok, mind az ügyfelek életét. Azonban hiába jeleztem ezeket a bankok felé, a nagy méretük és összetett hierarchiájuk miatt az ilyen „apróságokkal” az esetek többségében nem foglalkoztak.

Az előbb említettek következményeként jött az az ötletem, hogy a vizsgamunkám keretein belül elkészítsek egy olyan webes alkalmazást, amely az alapvető szolgáltatások mellett olyan funkciókkal is rendelkezik, melyek nem, vagy csak részben találhatók meg a magyarországi bankoknál. Más szóval egy olyan netbank szolgáltatást szerettem volna megalkotni, amely az általam ismert bankok szolgáltatásiból összegyűjti a hasznos szolgáltatásokat és emellett kiküszöböli az esetleges hiányosságokat, hibákat.

Az új dolgok iránti kíváncsiságom és az analitikus gondolkodásmódom sokat segít abban, hogy mind szolgáltatói, mind pedig felhasználói szemmel is alaposan átlássak egy-egy rendszert. Mivel gyerekkorom óta vonzódok a matematikához és a számokhoz, valamint több munkahelyen is dolgoztam olyan pozícióban, ahol pénzügyi és banki területeket is érintett a munkaköröm, így kézenfekvő volt számomra, hogy olyan témát válasszak a vizsgaremekhez, amiben a számok kiemelt szerepet játszanak.

3. Fejlesztői dokumentáció

Az alkalmazás fejlesztése során olyan rendszer-összetevők kerültek létrehozásra, melyek: kliens oldalon mindössze egy átlagos böngésző használatát igénylik, szerver oldalon pedig webservert, MariaDB vagy MySQL adatbázis-kezelő rendszer és PHP interpreter jelenlétét követelik meg.

3.1. Backend oldal

3.1.1 Az alkalmazott fejlesztői eszközök

Az X Bank alkalmazás backend oldalának a fejlesztése Windows 10 operációs rendszer alatt történt. Munkám során az alábbi szoftvereket, rendszereket és technológiákat használtam:

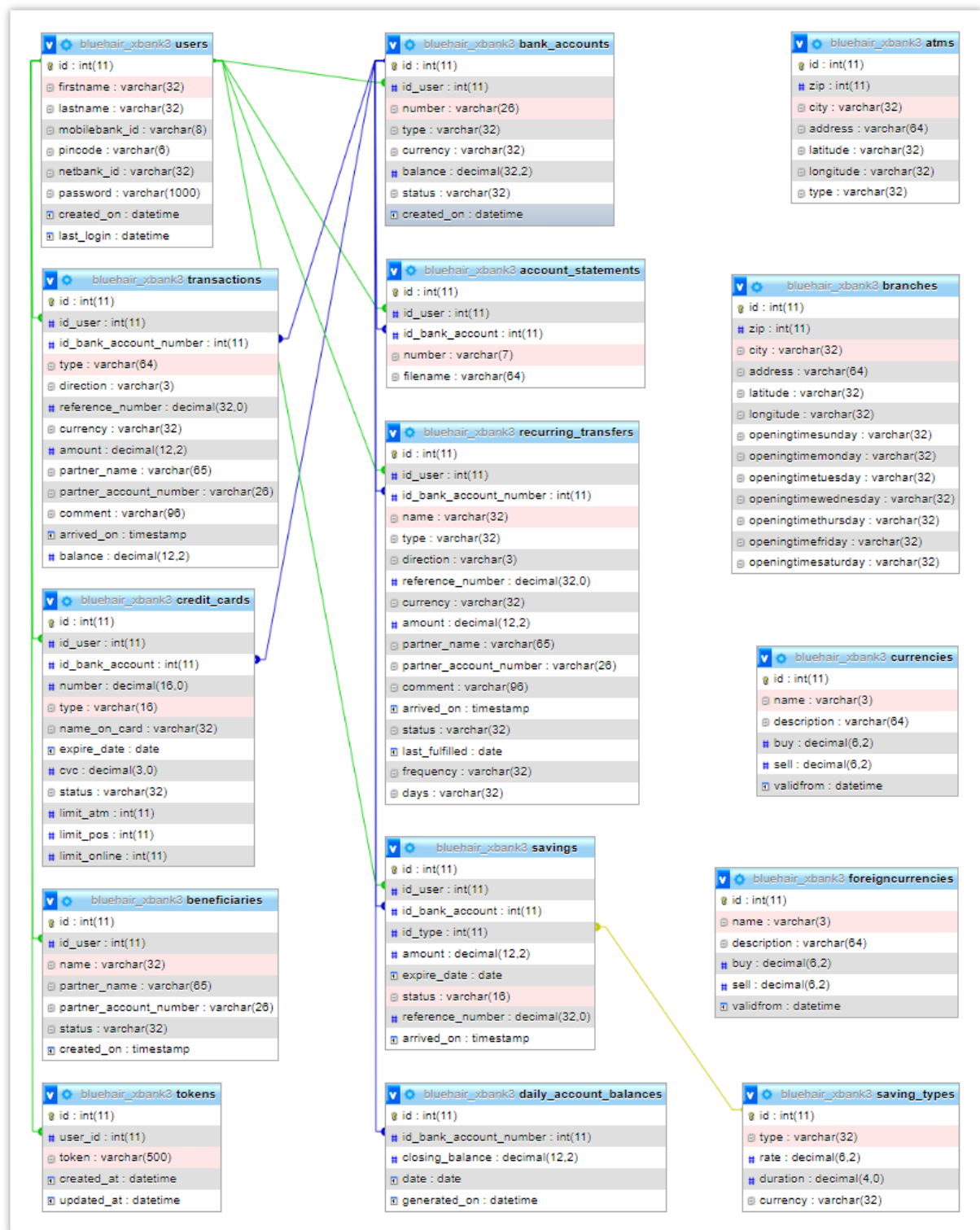
- alkalmazott programozási nyelvek, keretrendszerek és technológiák:
 - Slim v4.9
 - PSR7 v1.5
 - Illuminate Database v8.73
 - Slim Twig View v3.3
 - PHP 8
 - FPDF Library
- a szerveroldali forráskód elkészítése:
 - Visual Studio Code 1.66.1
- webservert, adatbázisok létrehozása és kezelése:
 - XAMPP 8.1.4 webservert csomag
- adatbázis-kezelő rendszer:
 - MariaDB
- böngésző:
 - Google Chrome v100.0.4896.127
 - Microsoft Edge v100.0.1185.50
- verziókövető rendszer:
 - Git
 - GitHub
- tesztelés:
 - PHP Debug Adapter for Visual Studio Code
 - Xdebug
 - Thunder Client v1.16.0

- dokumentációk elkészítése:
 - Microsoft Word for Office 365
 - Libre Office 6.3.5.2

3.1.2 Az adatmodellek leírása

Az adatbázis neve: ***xbank3***. A bankfiókok, bankautomaták és valuta-deviza árfolyamok adatai mellett a bankszámlaműveletekhez szükséges törzsadatokat és tranzakciós adatokat tartalmazza. Az alkalmazás HTTP kapcsolatot használ az adatbázissal történő kommunikáció során, REST API -n keresztül. A küldött és a fogadott adatok JSON formátumot használnak. A táblák szerkezete és a köztük lévő kapcsolatok az 1. ábrán láthatók. Az adatbázis az alábbi eseményindítókat tartalmazza:

- **savings_insert_after**: új megtakarítás rögzítése után létrehozza a megtakarításnak megfelelő tranzakciót a *transactions* táblában;
- **savings_update_after**: egy megtakarítás idő előtti feltörése esetén létrehozza a megfelelő tranzakciót a *transactions* táblában;
- **transactions_insert_after**: kimenő utalás esetén csökkenti, bejövő utalás esetén megnöveli az érintett bankszámla egyenlegét a megfelelő összeggel a *bank_accounts* táblában.



1. ábra - A MariaDB adatbázis felépítése és a táblák közötti kapcsolatok

Az adatbázis táblái:

- **account_statements:** bankszámla kivonatok adatai
- **atms:** bankautomaták adatai
- **bank_accounts:** bankszámlák adatai
- **beneficiaries:** kedvezményezettek (megbízási sablonok)
- **branches:** a Bank fiókjai
- **credit_cards:** bankkártyák adatai
- **currencies:** valutaárfolyamok
- **daily_account_balances:** napi bankszámlaegyenlegek
- **foreign_currencies:** devizaárfolyamok
- **recurring_transfers:** állandó megbízások
- **savings:** megtakarítások
- **saving_types:** megtakarítások fajtái
- **tokens:** hitelesítéshez használt tokenek
- **transactions:** banki tranzakciók adatai
- **users:** a Bank ügyfeleinek az adatai

Az táblák közötti kapcsolatok:

Alap tábla	Kulcs	Kapcsolt tábla	Kapcsolómező
bank_accounts	id	account_statements	id_bank_account
bank_accounts	id	credit_cards	id_bank_account
bank_accounts	id	savings	id_bank_account
bank_accounts	id	daily_account_balances	id_bank_account_number
bank_accounts	id	recurring_transfers	id_bank_account_number
bank_accounts	id	transactions	id_bank_account_number
saving_types	id	savings	id_type
users	id	account_statements	id_user
users	id	bank_accounts	id_user
users	id	beneficiaries	id_user
users	id	credit_cards	id_user
users	id	recurring_transfers	id_user
users	id	savings	id_user
users	id	transactions	id_user
users	id	tokens	user_id

3.1.3 Részletes feladatpecifikáció, algoritmusok

A szerveroldali API fogadja a beérkező kéréseket, melyek – amennyiben megfelelően vannak paraméterezve – végrehajtásra kerülnek, a végrehajtás eredménye pedig visszaküldésre kerül válaszként az őket meghívó metódusoknak. Szintén itt található azok a metódusok, melyek a banki tranzakciókhoz kapcsolódó banki költségeket számolják ki és létrehozzák az átutalási díjaknak megfelelő tranzakciókat. Ezeknek a PHP scripteknek a megfelelő működését az adatbázisban létrehozott eseményindítók egészítik ki. A felhasználók jelszavai titkosítottan vannak tárolva!!

A program főbb részei:

- **routes.php:** az API végpontokat tartalmazza. Ha az adott művelet sikeres volt, akkor a lenti felsorolásban található visszatérési értékeket adja vissza. Sikertelen művelet esetén hibaüzenetet a vissza válaszként.

- **bejelentkezéshez nem kötött végpontok:**

- **/:** üzenetet küld vissza, mely szerint nincs jogosultságunk az oldal megtekintéséhez

- státuszkód: 401

- **/getatms:** a bankautomaták adatait adja vissza

- metódus típusa: GET

- paraméterek: -

- visszatérési érték: az adatbázis *atms* táblájának a tartalma egy többelemű tömb formájában, az alábbi adatszerkezettel:

address:	"Szent István krt. 10."
city:	"Budapest"
id:	3
latitude:	"47,5363719"
longitude:	"19,047132"
type:	"Euro"
zip:	1137

- státuszkód: 200

- **/getbranches:** a bankfiókok adatait adja vissza

- metódus típusa: GET

- paraméterek: -

- visszatérési érték: az adatbázis *branches* táblájának a tartalma egy többelemű tömb formájában, az alábbi adatszerkezettel:

address:	"Fóti út 4."
city:	"Budapest"
id:	3
latitude:	"47,5820813"

```

longitude:           "19,0820029"
openingtimefriday:   "08:00-15:00"
openingtimemonday:   "08:00-17:00"
openingtimesaturday: "00:00-00:00"
openingtimesunday:   "00:00-00:00"
openingtimethursday: "08:00-17:00"
openingtimetuesday:  "08:00-17:00"
openifngtimewednesday: "08:00-17:00"
zip:                 1044
    
```

▪ **/getcurrencies:** valuta árfolyamok lekérdezése

- metódus típusa: GET
- paraméterek: -
- visszatérési értékek: az adatbázis *currencies* táblájának a tartalma egy több-elemű tömb formájában, az alábbi adatszerkezettel:

```

buy:                 "338.79"
description:         "Amerikai dollár"
id:                  2
name:                "USD"
sell:                "359.75"
validfrom:           "2022-04-23 08:04:03"
    
```

- státuszkód: 200

▪ **/getforeigncurrencies:** devizaárfolyamok lekérdezése. A metódus típusa, a paraméterek és a visszatérési értékek megegyeznek az előző végpontnál leírtakkal, de az adatok a *foreigncurrencies* táblából érkeznek

▪ **/login:** a felhasználó bejelentkezését végzi el. Hibás hitelesítési adatok esetén hibaüzenetet ad vissza, sikeres azonosítás esetén a felhasználói adatokat és egy tokent

- metódus típusa: POST
- paraméterek: felhasználói azonosítási adatok az alábbi adatszerkezet szerint:

```

netbankId:           "verotomi123"
password:             "8p56bZrK"
    
```

- példa egy lehetséges visszatérési értékre:

```

created_on:           "2018-12-01 12:00:00"
firstname:            "Tamás"
id:                   1
last_login:           "2022-04-23 19:57:19"
lastname:             "Verovszki"
netbank_id:           "verotomi123"
token:                "5fe9917a93369fea915a7e2cc3331f13880f35d5d8020bd670e03eeee95142c6c37105cb60cbe85ed801f3f0da8ce0f1948a36c6a63590e4b6236b43f13b986a"
    
```

- státuszkód: 200

▪ **/logout:** a felhasználó bejelentkezését végzi el

- metódus típusa: POST
- paraméterek: felhasználói azonosító adatok az alábbi adatszerkezet szerint:

```
id_user: 1
netbank_id: "verotomi123"
```

- visszatérési érték: logikai igaz vagy hamis (true, false)
- státuszkód: 200

- **bejelentkezéshez kötött végpontok:** a megfelelő Bearer tokent el kell küldeni a HTTP kérés során!

- **/breakdeposit:** lekötött pénz idő előtti feltörését végzi

- metódus típusa: PUT
- paraméterek: felhasználói azonosító és a megtakarítás azonosítója az alábbi adatszerkezet szerint:

```
id: 108
id_user: 1
```

- visszatérési érték: a felhasználó összes megtakarítását tartalmazó tömb az alábbi adatszerkezet szerint:

```
amount: "100000.00"
arrived_on: "2019-03-26 20:38:10"
balance: "21095847.00"
created_on: "2018-12-01 12:00:00"
currency: "Forint"
duration: "365"
expire_date: "2020-04-25"
id: 1
id_bank_account: 1
id_type: 3
id_user: 1
number: "99999999-11111111-11111111"
rate: "1.15"
reference_number: "544825064"
saving_id: 1
saving_status: "Expired"
status: "Active"
type: "Yearly saving"
```

- státuszkód: 200

- **/changepassword:** jelszóváltoztatást végez

- metódus típusa: POST
- paraméterek: felhasználói azonosítói adatok az alábbi adatszerkezet szerint:

```
id_user: 1
netbank_id: "verotomi123"
old_password: "8p56bZrK"
password: "12121212"
```

- visszatérési érték: logikai igaz érték (true)

- státuszkód: 200

▪ **/createbeneficiary:** új kedvezményezett rögzítése

- metódus típusa: POST
- paraméterek: a felhasználó azonosítója és az új kedvezményezett adatai az alábbi adatszerkezet szerint:

```
id_user: 7
name: "Kovács Erika"
partner_account_number: "12121212-54545454"
partner_name: "Kovács Erika"
```

- visszatérési érték: logikai igaz érték (true)
- státuszkód: 201

▪ **/createrecurringtransfer:** új állandó átutalási megbízás rögzítése

- metódus típusa: POST
- paraméterek: a felhasználó azonosítója és az állandó megbízás adatai az alábbi adatszerkezet szerint:

```
amount: "83000"
comment: "Szerződésszám: LH5120-34152556"
currency: "Forint"
days: "5. napján"
frequency: "Every month"
id_bank_account_number: 1
id_user: 1
name: "Lakáshitel"
partner_account_number: "11773131-25254017"
partner_name: "OTP Bank Zrt."
```

- visszatérési érték: a felhasználó összes állandó megbízását tartalmazó tömb az alábbi adatszerkezet szerint:

```
amount: "83000.00"
arrived_on: "2022-04-27 14:23:45"
balance: "22103577.00"
comment: "Szerződésszám: LH5120-34152556"
created_on: "2018-12-01 12:00:00"
currency: "Forint"
days: "5. napján"
direction: "out"
frequency: "Every month"
id: 1
id_bank_account_number: 1
id_user: 1
last_fulfilled: null
name: "Lakáshitel"
number: "99999999-11111111-11111111"
partner_account_number: "11773131-25254017"
partner_name: "OTP Bank Zrt."
recurring_transfer_id: 33
recurring_transfer_status: "Active"
reference_number: "181264404"
```

```
status: "Active"
type: "Retail bank account"
```

- státuszkód: 201

▪ **/createtransferonetime:** egyszeri átutalási megbízás rögzítése

- metódus típusa: POST
- paraméterek: a felhasználó azonosítója és az átutalási megbízás adatai az alábbi adatszerkezet szerint:

```
amount: "15600"
comment: "Virágállvány ára"
currency: "Forint"
id_bank_account_number: 3
id_user: 1
partner_account_number: "99999999-22222222-33333333"
partner_name: "Balassy Melinda"
```

- visszatérési érték: a létrejött tranzakció adatai az alábbi adatszerkezet szerint:

```
amount: "15600"
arrived_on: "2022/04/27 14:45:18"
comment: "Virágállvány ára"
currency: "Forint"
direction: "out"
id: 40342
id_bank_account_number: 3
id_user: 1
partner_account_number: "99999999-22222222-33333333"
partner_name: "Balassy Melinda"
reference_number: 767630281
type: "outgoing transfer"
```

- státuszkód: 201

▪ **/deletebeneficiary:** meglévő kedvezményezett adatainak a törlése

- metódus típusa: DELETE
- paraméterek: a törlendő kedvezményezett azonosítója az alábbi adatszerkezet szerint:

```
id: "66"
```

- visszatérési érték: logikai igaz érték (true)
- státuszkód: 200

▪ **/deleterecurringtransfer:** meglévő állandó átutalási megbízás törlése

- metódus típusa: DELETE
- paraméterek: a törlendő kedvezményezett azonosítója az alábbi adatszerkezet szerint:

```
id: "66"
id_user: 2
```

- visszatérési érték: logikai igaz érték (true)

```

amount:                "10000.00"
arrived_on:            "2022-03-23 09:29:57"
balance:               "22123636.00"
comment:               "Kölcsön"
created_on:            "2018-12-01 12:00:00"
currency:              "Forint"
days:                 "19"
direction:             "out"
frequency:             "Every month"
id:                    1
id_bank_account_number: 1
id_user:               1
last_fulfilled:        null
name:                  "Antinak kölcsön"
number:                "99999999-11111111-11111111"
partner_account_number: "00000000-12345678-55220011"
partner_name:          "Nagy Antal"
recurring_transfer_id: 14
recurring_transfer_status: "Active"
reference_number:       "513819134"
status:                "Active"
type:                  "Retail bank account"

```

- státuszkód: 200

▪ **/getaccounts:** a felhasználó bankszámla-adatait adja vissza

- metódus típusa: POST
- paraméterek: felhasználói adatok az alábbi adatszerkezet szerint:

```
id_user: 1
```

- visszatérési érték: a felhasználó bankszámla-adatait tartalmazó tömb az alábbi adatszerkezet szerint:

```

balance:                "21095747.00"
created_on:             "2018-12-01 12:00:00"
currency:               "Forint"
id:                     1
id_user:                1
number:                 "99999999-11111111-11111111"
status:                 "Active"
type:                   "Retail bank account"

```

- státuszkód: 200

▪ **/getaccounthistory:** a felhasználó egyik bankszámlájának a számlatörténetét kéri le a megadott paraméterek alapján

- metódus típusa: POST
- paraméterek: a felhasználó azonosítója és a lekérdezés paraméterei az alábbi adatszerkezet szerint:

```

direction:              "all"
from:                   "2022-03-27 00:00:00"
id_bank_account_number: 1
id_user:                1
to:                     "2022-04-27 23:59:59"

```

- visszatérési érték: a szóban forgó bankszámlának a lekérdezési paraméterek alapján leszűrt tranzakciós adatait tartalmazó tömb az alábbi adatszerkezettel:

```

amount:                "120.00"
arrived_on:            "2022-04-04 00:02:05"
balance:               "22103577.00"
comment:               "Oltai Emese 5000.00 Forint"
created_on:            "2018-12-01 12:00:00"
currency:              "Forint"
direction:             "out"
id:                    1
id_bank_account_number: 1
id_user:               1
number:                "99999999-11111111-11111111"
partner_account_number: null
partner_name:          "X Bank Limited"
reference_number:      "957128420"
status:                "Active"
transaction_id:        38810
transaction_type:      "recurring transfer fee"
type:                  "Retail bank account"

```

- státuszkód: 200

▪ **/getbeneficiaries:** a kedvezményezettek listájának a lekérdezése

- metódus típusa: POST
- paraméterek: a felhasználó azonosítója az alábbi adatszerkezet szerint:

```

id_user:                1

```

- visszatérési érték: a felhasználó kedvezményezettjeinek az adatait tartalmazó tömb az alábbi adatszerkezet szerint:

```

created_on:             "2021-11-17 20:16:48"
id:                     9
id_user:                1
name:                   "Fenyvesi Erika lakossági"
partner_account_number: "99999999-44444444-11111111"
partner_name:           "Fenyvesi Erika"
status:                 "Active"

```

- státuszkód: 200

▪ **/getcreditcards:** a felhasználó bankkártya-adatait kérdezi le

- metódus típusa: POST
- paraméterek: a felhasználó azonosítója az alábbi adatszerkezet szerint:

```

id_user:                1

```

- visszatérési érték: a felhasználó bankkártya-adatait tartalmazó tömb az alábbi adatszerkezet szerint:

```

balance:                "29282285.00"
created_on:             "2018-12-01 12:00:00"
credit_card_status:     "Active"
creditcard_id:          2

```



```

creditcard_number: "2805630076646928"
creditcard_type: "Visa"
currency: "Forint"
cvc: "234"
expire_date: "2024-12-31"
id: 3
id_bank_account: 3
id_user: 1
limit_atm: 150000
limit_online: 200000
limit_pos: 180000
name_on_card: "Verovszki Tamás"
number: "99999999-11111111-33333333"
status: "Active"
type: "Saving account"

```

- státuszkód: 200

▪ **/getrecurringtransfers:** a felhasználó állandó átutalási megbízásait kérdezi le

- metódus típusa: POST

- paraméterek: a felhasználó azonosítója az alábbi adatszerkezet szerint:

```
id_user: 1
```

- visszatérési érték: a felhasználó állandó átutalási megbízásait tartalmazó tömb az alábbi adatszerkezet szerint:

```

amount: "35000.00"
arrived_on: "2019-04-12 05:14:29"
balance: "29282285.00"
comment: "Lakbér"
created_on: "2018-12-01 12:00:00"
currency: "Forint"
days: "10"
direction: "out"
frequency: "Every month"
id: 3
id_bank_account_number: 3
id_user: 1
last_fulfilled: "2022-04-10"
name: "Lakbér"
number: "99999999-11111111-33333333"
partner_account_number: "99999999-44444444-11111111"
partner_name: "Fenyvesi Erika"
recurring_transfer_id: 11
recurring_transfer_status: "Active"
reference_number: "289303111"
status: "Active"
type: "Saving account"

```

- státuszkód: 200

▪ **/getsavings:** a felhasználó megtakarításait kérdezi le

- metódus típusa: POST

- paraméterek: a felhasználó azonosítója az alábbi adatszerkezet szerint:

```
id_user: 1
```

- visszatérési érték: a felhasználó megtakarításait tartalmazó tömb az alábbi adatszerkezet szerint:

```

amount:                "1000000.00"
arrived_on:            "2019-04-02 06:42:50"
balance:               "22103577.00"
bank_account_type:     "Retail bank account"
created_on:            "2018-12-01 12:00:00"
currency:              "Forint"
duration:              "365"
expire_date:           "2020-04-01"
id:                    1
id_bank_account:       1
id_type:               1
id_user:               1
number:                "99999999-11111111-11111111"
rate:                  "4.80"
reference_number:      "727541320"
saving_id:             16
saving_status:         "Expired"
status:                "Active"
type:                  "Yearly saving"

```

- státuszkód: 200

- **/getsavingtypes:** új megtakarítás rögzítésekor kerül meghívásra, a lehetséges pénzlekötési módokat és azok tulajdonságait adja vissza

- metódus típusa: GET
- paraméterek: -
- visszatérési érték: az adatbázis *saving_types* táblájának a tartalma az alábbi adatszerkezet formájában:

```

currency:              "Forint"
duration:              "30"
id:                    2
rate:                  "3.60"
type:                  "Monthly saving"

```

- státuszkód: 200

- **/getstatements:** a felhasználó bankszámlakivonatainak a listáját adja vissza

- metódus típusa: POST
- paraméterek: a felhasználó azonosítója az alábbi adatszerkezet szerint:

```

id_user:               1

```

- visszatérési érték: a bankszámlakivonat-listát tartalmazó tömb az alábbi adatszerkezet szerint:

```

account_statements_id: 174
account_statements_number: "2021/07"
balance:               "29282285.00"
bank_accounts_number:  "99999999-11111111-33333333"
created_on:            "2018-12-01 12:00:00"

```

```

currency: "Forint"
filename: "Xbank-statement-2021-07-99999999-11111111-33333333.pdf"
id: 3
id_bank_account: 3
id_user: 1
number: "99999999-11111111-33333333"
status: "Active"
type: "Saving account"

```

- státuszkód: 200

▪ **/insertsaving:** új megtakarítást rögzít

- metódus típusa: POST
- paraméterek: a felhasználó azonosítója és a megtakarítás paraméterei az alábbi adatszerkezet szerint

```

amount: "4000"
duration: 365
id_bank_account: 2
id_type: 1
id_user: 1

```

- visszatérési érték: a rögzített megtakarítás adatai az alábbi adatszerkezet szerint:

```

amount: "4000"
arrived_on: "2022/04/27 16:43:44"
expire_date: "2023/04/27"
id: 111
id_bank_account: 2
id_type: 1
id_user: 1
reference_number: 670174204
status: "Active"

```

- státuszkód: 201

▪ **/statistics:** banki pénzügyi statisztika megjelenítése, megtekintéséhez admin jogosultság szükséges

- metódus típusa: GET
- paraméterek: -
- visszatérési érték: szerver oldalon renderelt HTTP kód
- státuszkód: 200

▪ **/updatebeneficiary:** egy kedvezményezett adatainak a módosítása

- metódus típusa: PUT
- paraméterek: a felhasználó azonosítója és a módosítandó kedvezményezett adatai az alábbi szerkezet szerint:

```

id: 54
id_user: 7
name: "Dönci"

```

```
partner_account_number: "12125454-88554455"
partner_name: "Baradlay Ödön"
```

- visszatérési érték: a módosított rekord az alábbi adatszerkezet szerint:

```
created_on: "2022-04-27 15:01:17"
id: 54
id_user: 7
name: "Dönci"
partner_account_number: "12125454-88554455"
partner_name: "Baradlay Ödön"
status: "Active"
```

- státuszkód: 201

▪ **/updatecreditcard:** egy bankkártya adatainak a módosítása

- metódus típusa: PUT
- paraméterek: a felhasználó azonosítója és a módosítandó kedvezményezett adatai az alábbi szerkezet szerint:

```
id: 7
id_bank_account: 12
id_user: 7
limit_atm: "650"
limit_online: "1650"
limit_pos: "1200"
status: "Active"
```

- visszatérési érték: a felhasználó bankkártyáinak a listája az alábbi adatszerkezet szerint:

```
balance: "50976.00"
created_on: "2022-04-09 10:29:40"
credit_card_status: "Active"
creditcard_id: 7
creditcard_number: "2805123446465210"
creditcard_type: "Visa"
currency: "Euro"
cvc: "311"
expire_date: "2024-12-31"
id: 12
id_bank_account: 12
id_user: 7
limit_atm: 650
limit_online: 1650
limit_pos: 1200
name_on_card: "Bakó Dániel"
number: "99999999-77777777-22222222"
status: "Active"
type: "Foreign currency account"
```

- státuszkód: 201

▪ **/updaterecurringtransfer:** egy állandó átutalási megbízás adatainak a módosítása

- metódus típusa: PUT

- paraméterek: a felhasználó azonosítója és a módosítandó megbízás adatai az alábbi szerkezet szerint:

```

amount:                "100.00"
comment:               "Szerződés: 2022/08645PD"
currency:              "Forint"
days:                 "Wednesday"
frequency:             "Every week"
id:                    16
id_bank_account_number: 1
id_user:               7
name:                  "Motor törlesztőrészlete"
partner_account_number: "50505050-80808080"
partner_name:          "Honda Magyarország Kft."
status:                "Active"
    
```

- visszatérési érték: a felhasználó bankkártyáinak a listája az alábbi adatszerkezet szerint:

```

balance:                "50976.00"
created_on:             "2022-04-09 10:29:40"
credit_card_status:     "Active"
creditcard_id:          7
creditcard_number:      "2805123446465210"
creditcard_type:        "Visa"
currency:               "Euro"
cvc:                    "311"
expire_date:            "2024-12-31"
id:                      12
id_bank_account:        12
id_user:                 7
limit_atm:               650
limit_online:            1650
limit_pos:               1200
name_on_card:           "Bakó Dániel"
number:                  "99999999-77777777-22222222"
status:                  "Active"
type:                    "Foreign currency account"
    
```

- státuszkód: 201
- **models mappa:** az adatbázis tábláihoz való hozzáférést kezelő PHP fájlokat tartalmazza
- **scripts.php:** többször, több helyen használt függvények gyűjteménye
- **AuthMiddleware osztály:** a bejelentkezéshez szükséges token kezelését és ellenőrzését végzi
- **statistics.twig:** backend oldali renderelt oldal, a banki pénzügyi statisztika megjelenítéséért felelős

A fentiekben megemlített adatszerkezetekben szereplő mezők részletes specifikációi megtalálhatók az 1. ábrán! A backend oldal részét képezik továbbá az alábbi CRON feladatok is, melyek meghatározott időközönként futnak le:

- **DailyTaskDailyAccountBalances:** minden nap elején elmenti az adatbázisba a bankszámlák előző napi záróegyenlegeit.
- **GenerateRandomCurrencyRates:** új valuta- és devizaárfolyamokat generál véletlenszerű adatokkal. Szintén a program működésének a demonstrálásához van rá szükség, éles működés esetén kikapcsolandó.
- **GenerateRandomTransactions:** véletlenszerűen generált bejövő utalásokat hoz létre annak érdekében, hogy folyamatosan legyenek olyan adatok az adatbázisban, melyek szükségesek a program működésének a bemutatásához. Enélkül ugyanis – ha a felhasználó nem használná aktívan az alkalmazást – nem lenne elegendő megjeleníthető adat a programban. Ha éles használatra kerül az alkalmazás, akkor ez a script kikapcsolásra kerül.
- **DailyTaskRecurringTransfers:** az állandó átutalási megbízásokat dolgozza fel, létrehozza az esedékes utalásoknak megfelelő tranzakciókat a *transactions* táblában
- **DailyTaskSavings:** az adott napon lejáró megtakarításoknak megfelelő tranzakciós adatokat hoz létre a *transactions* táblában. Kiszámítja a lekötés által termelődött kamatot, majd visszavezeti a bankszámlára a tőkét is és a kamatot is egy-egy tranzakció formájában.
- **MonthlyTaskGenerateMonthlyFees:** hó végén létrehozza a Bank által felszámított számlavezetési díjakat, majd rögzíti azokat tranzakció formájában a *transactions* táblában.
- **MonthlyTaskGenerateAccountStatements:** elkészíti a bankszámlákhoz tartozó előző havi számlakivonatokat PDF formátumban és ennek megfelelően aktualizálja az *account_statements* táblát.

3.1.4 A program telepítése

A backend oldal telepítésének a lépései:

1. A Git repository klónozása: git clone <https://github.com/verotomi/X-Bank-Backend.git>
2. A programot tartalmazó mappa megnyitása: cd X-Bank-Frontend
3. Csomagok telepítése: composer install
4. Az adatbázis-szerver elindítása és az *xbank3* adatbázis
5. A szerver elindítása: php -S 0.0.0.0:8888

3.1.5 Tesztelési dokumentáció

A backend oldali program egyszerű struktúráját tekintve a frontend oldalénál kevesebb és egyszerűbb tesztelésekre volt szükségem.

3.1.5.1 Függvények és metódusok tesztelése

Az általánosan használt segédfüggvények megfelelő működését egységtesztekkel vizsgáltam, a HTTP kérések feldolgozását végzőknél pedig funkcionális tesztelést alkalmaztam, vagyis elküldtem egy tesztadatokból álló kérést az API végpont felé, és megvizsgáltam, hogy az általam elvárt válasz jön-e vissza?

Példák a bejelentkezés tesztelésére:

- **sikeres belépés tesztelése:**

- meghívott végpont: /login
- az API kérésben szereplő adatok:

netbankId:	"verotomi123"
password:	"8p56bZrK"

- az elvárt válasz: a felhasználó bejelentkezési adatait és egy tokent tartalmazó JSON objektum

- visszakapott válasz:

created_on:	"2018-12-01 12:00:00"
firstname:	"Tamás"
id:	1
last_login:	"2022-04-29 14:20:55"
lastname:	"Verovszki"
netbank_id:	"verotomi123"
token:	"5ea6f03b1dbec70a85bac8e4ed34f5be61fb7b8bbfd9f2dfdf4a7c0164700f29ed599b2fd745b9625bddb0d325041ec68f83076c8a6372c134cb890fdeae6b44"

- a teszt eredménye sikeres, az elvárt válasz érkezett vissza.

- **sikertelen belépés tesztelése:**

- meghívott végpont: /login
- az API kérésben szereplő adatok:

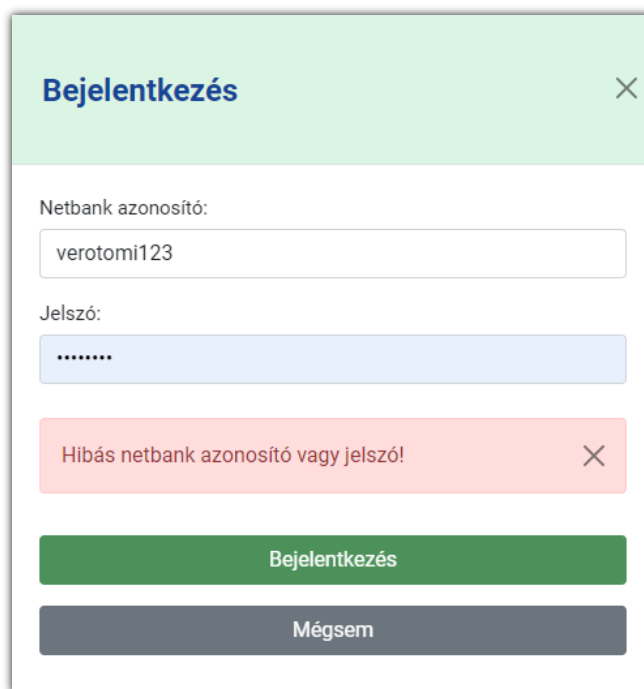
netbankId:	"verotomi123"
password:	"12345678"

- az elvárt válasz: hibaüzenet

- visszakapott válasz:

error: "Hibás netbank azonosító vagy jelszó!"

- a teszt eredménye sikeres, az elvárt válasz érkezett vissza.



2. ábra - Sikertelen bejelentkezési kísérlet

3.1.5.2 Integrációs teszt

Az alkalmazás különböző részeinek az együttműködését vizsgáltam. Egy sok fejtörést okozó hibával találkoztam a teszt során. Amikor elkészítettem a token ellenőrzést végző *Auth-Middleware* osztály azon részét, ami a lejárt tokenről figyelmezteti a felhasználót és tesztelni kezdtem, akkor az nem akart működni. Az API nem küldött semmilyen választ a token meglétét igénylő kérésekre. Hosszas keresgélés után sikerült rájönnöm, hogy a kérdéses osztályban a sikeres token azonosítást kezelő osztály által használt *Response* interfészt akartam használni a hibaüzenet kliens felé történő visszaküldésénél, holott ott nem interfészt, hanem osztályt kell alkalmazni. Az egyező elnevezés tévesztett meg, ráadásul a fejlesztőkörnyezet sem jelzett sem szintaktikai, sem szemantikai hibát.

3.1.5.3 Rendszer teszt

Az API felület és a MariaDB adatbázis közötti kommunikáció kezelését jelentős mértékben elvégzi a Slim4 keretrendszer, így ennek a tesztelésekor csak ritkán kellett a phpMyAdmin meglehetősen szűkszavú hibaüzeneteivel bajlódnom.

Az összes API végpont tesztelése megtörtént a Visual Studio Code-ban használt Thunder Client bővítmény használatával. A kérések egy *Xbank requests* nevű gyűjteménybe lettek elmentve JSON formátumban. A gyűjtemény megtalálható a projektkönyvtár *test* nevű

mappájában, a végpontok tesztelése ezáltal bármikor megismételhető. Arra azonban figyelni kell, hogy a token meglétét igénylő végpontok tesztelése előtt a login kérést hajtsuk végre, és az abban kapott új tokennel helyettesítsük a gyűjteményben található tokeneket. Továbbá arra is figyelmet kell fordítani, hogy a *config.php* fájlban a szerveradatok alapesetben a <https://verovszki.eu/xbankapi> címhez tartoznak. Amennyiben helyi szerveren kívánjuk a tesztet futtatni, úgy a megfelelő szerver adatokat ki kell választani a konfigurációs fájlban.

3.1.5.4 A program ismert hibái

Az alábbi programhibákról van tudomásom, melyek kijavítására időhiány miatt még nem volt lehetőségem:

- a havi bankszámlakivonatok elkészítését végző programrész kizárólag az adott hónap 31-edik napján fut le. Ha a hónap ennél rövidebb, akkor abban a hónapban nem készül kivonat. A program működését és az adatok helyességét ez a hiba nem befolyásolja, a kivonatok utólag is elkészíthetők.

3.1.6 Továbbfejlesztési lehetőségek

A backend oldalon maradtak még nyitott ötletek, melyek a későbbiekben megvalósításra kerülhetnek:

- kétlépcsős azonosítás megvalósítása;
- felhasználói fiók zárolása többszöri sikertelen bejelentkezési próbálkozás után.

Időhiány miatt a szerveroldali validáció nem lett befejezve. Az minden végponton ellenőrzésre kerül, hogy a műveletekhez szükséges paramétereket tartalmazzák-e a HTTP kérések, de annak a vizsgálatát, hogy a paraméterek az előírásoknak, elvárásoknak megfelelőek-e, majd egy későbbi továbbfejlesztés fogja megoldani.

3.2. Frontend oldal

3.2.1 Az alkalmazott fejlesztői eszközök

Az X Bank alkalmazás frontend részének a fejlesztése is Windows 10 operációs rendszer alatt történt. A munka során az alábbi szoftver eszközöket, rendszereket és technológiákat használtam:

- A frontend alkalmazás által használt nyelvek, keretrendszerek és technológiák:
 - React v17.0.2 Javascript könyvtár
 - React Redux v7.2.6 Javascript könyvtár
 - React Bootstrap v2.0.3

- Lodash 4.17.21
- SASS 1.45.0
- Javascript
- HTML5
- CSS3
- a kliensoldali forráskód elkészítése:
 - Visual Studio Code 1.66.1
- böngésző:
 - Google Chrome v100.0.4896.127
 - Microsoft Edge v100.0.1185.50
- verziókövető rendszer:
 - Git
 - GitHub
- tesztelés:
 - Visual Studio Code 1.66.1
- Dokumentáció elkészítése:
 - Microsoft Word for Office 365
 - Libre Office 6.3.5.2

3.2.2 Részletes feladat-specifikáció, algoritmusok

A fejlesztés során nagy figyelmet fordítottam a felhasználói élmény fokozására. A listák – ahol ezt a környezet igényli és lehetővé teszi – interaktívak, rendezhetőek, szűrhetőek, a kurzor ráhúzására pedig extra információkat szolgáltatnak. A beviteli mezők validáltak és a sikeres, illetve sikertelen műveletek eredményeihez hasonlóan eltérő színnel jelenítik meg az üzeneteket. A jelölőnégyzetekkel nem lehet a logikával ellentétes állapotokat beállítani (pl. számlatörténet esetén vagy a kimenő, vagy a bejövő, vagy mindkét irányú pénzmozgások kérhetőek le, de olyan lehetősége nincs a felhasználónak, hogy egyik négyzetet sem jelöli be). A számlatörténetnél található dátum beviteli mező rugalmasan kezeli a kezdő- és végdátumokat és a köztük lévő periódust is. Az előbbi azt takarja, hogy az alkalmazás nem enged az aznapinál későbbi végdátumot beállítani. Az utóbbi pedig azt, hogy akár a kezdő-, akár a végdátumot módosítja a felhasználó, automatikusan átállítódik a másik dátum a maximális – két hónapos lekérdezési periódusnak megfelelő értékre. Természetesen lehetőség van ennél rövidebb intervallum kiválasztására is.

Szintén nagy hangsúlyt fektettem arra is, hogy az alkalmazást mind asztali számítógépen, mind tablaten vagy mobiltelefonon kényelmesen lehessen használni. A betűméretek, az ikonok méretei, a külső-és a belső margók, de néhol még a megjelenített tartalom és az oldal elemeinek a működése is dinamikusan változik a képernyőméretek figyelembevételével. Az oldalsó menü például mobiltelefonos használat során automatikusan eltűnik egy-egy menüpont kiválasztása után, míg asztali nézet esetében továbbra is látható marad.

3.2.2.1 Az alkalmazás alapjai

- **App.js:** a bejelentkezett állapot kezelése, az annak megfelelő oldalsó menü és programösszetevők megjelenítése
- **index.js:** itt helyezkedik el a Redux store

3.2.2.2 Az alkalmazás bejelentkezés nélkül használható része

A netbank alkalmazás első része nincs bejelentkezéshez kötve, itt általános adatokat és információkat tud megtekinteni a felhasználó, illetve innen történik a belépés a bejelentkezéssel elérhető oldalakra. A programban használt főbb komponensek:

- **Atms:** a bankautomaták listáját jeleníti meg.
- **Branches:** a bankfiókok listáját jeleníti meg. Egy-egy fiók heti nyitvatartási idejét felugró ablak formájában mutatja meg.
- **Contact:** a bankkal történő kapcsolatba lépés lehetőségeit mutatja meg (cím és térkép, telefonszámok, postázási cím, elektronikus elérhetőségek).
- **Currencies:** a valutaárfolyamok listáját jeleníti meg.
- **ForeignCurrencies:** a devizaárfolyamok listáját jeleníti meg.
- **Information:** az alkalmazás adatait (készítő, aktuális verziószám stb.), jogi információkat és az adatvédelmi tájékoztatót tartalmazza.
- **LoginButton:** sikeres azonosítás után átirányítja a felhasználót a saját, személyes oldalára.
- **Main:** megjeleníti a bejelentkező gombot tartalmazó fejléctet, az oldalsó menüt és a lábléctet.
- **MainHeader:** a fejléctet jeleníti meg, a benne található menügombbal és a bejelentkezés gombbal együtt.
- **MainOverview:** megjeleníti a főoldalt, ahol egy összefoglaló látható a Bank fiókjairól, bankautomatáiról, a valuta- és a devizaárfolyamokról, valamint egy tájékoztató a Bank mobilalkalmazásáról.

- **MainSidebar:** a bejelentkezés nélküli rész oldalsómenüjét valósítja meg.
- **MobilebankAd:** egy rövid tájékoztatót jelenít meg a bank Androidos mobilalkalmazásáról.

3.2.2.3 Az alkalmazás bejelentkezéshez kötött része

Az itt található oldalak mindegyike bejelentkezéshez kötött és csak a sikeres azonosítás után jelenik meg. A bejelentkezéshez kötött részen visszaszámláló időzítő működik, amely a beállított időnek megfelelő felhasználói inaktivitás esetén először tájékoztatja a felhasználót, majd további inaktivitás után elindítja az automatikus kijelentkezési folyamatot.

- **Accounts:** az ügyfél bankszámláit és azok egyenlegeit tartalmazó listát jelenít meg.
- **Beneficiaries:** a felhasználó kedvezményezettjeinek a listáját jeleníti meg (partner-lista).
- **CreditCards:** ha van a felhasználónak bankkártyája, akkor itt jelennek meg azok adatai.
- **EditBeneficiaries:** a már meglévő kedvezményezettek adatait lehet vele módosítani.
- **EditCreditCard:** a bankkártya beállításait teszi lehetővé. Közvetlenül nem érhető el a menüből, csak a bankkártyák listájából.
- **EditRecurringTransfers** a már korábban rögzített állandó megbízások módosítására ad lehetőséget. Közvetlenül nem érhető el a menüből, csak a listából
- **History:** az adott bankszámlához kapcsolódó bankszámlatörténetet jeleníti meg, a lista elemeiről részletes információkat jelenít meg felugró ablakok formájában.
- **Netbank:** megjeleníti a felhasználói profil gombot tartalmazó fejléct, az oldalsó menüt és a lábléct.
- **NetbankSidebar:** a bejelentkezés után megjelenő rész oldalsómenüjét valósítja meg.
- **NetbankHeader:** a menügombot és a profilgombot magába foglaló fejléc.
- **NetbankOverview** összefoglaló listákat tartalmaz a felhasználó banki adatairól: bankszámlák, megtakarítások és bankkártyák listája.
- **NewBeneficiary:** itt van lehetősége a felhasználónak új kedvezményezettek rögzítésére.
- **NewRecurringTransfer:** állandó megbízások megadására ad lehetőséget.
- **NewSaving:** új megtakarítás létrehozását biztosítja.
- **Profile:** megjeleníti az utolsó belépés idejét, jelszóváltoztatási és kijelentkezési lehetőséget nyújt.

- **RecurringTransfers:** a már korábban rögzített állandó megbízások listáját tartalmazza a megbízások törlési lehetőségével.
- **Savings:** az aktív megtakarításokat listázza ki, lehetőséget ad a megtakarítások részleteinek a megtekintésére felugró ablakban vagy egy megtakarítás idő előtti feltörésére.
- **Statements:** az elérhető bankszámlakivonatokat jeleníti meg. Lehetőség van azok megtekintésére és letöltésére is.
- **Statistics:** egy részletes pénzügyi statisztikát ad a bank pénzeszközeiről.
- **TransferOneTime** egyszeri átutalás rögzítésére szolgál.

3.2.2.4 Közösen használt komponensek és függvények

- **ApiProvider:** az API kéréseket végző függvények gyűjtőhelye.
- **BreadCrumbs:** a kiindulóponttól a felhasználó jelenlegi tartózkodási helyéig vezető utat jeleníti meg.
- **Filterkomponensnév:** az adott komponenshez tartozó szűrő kezelését végzi.
- **Footer:** a láblécben szereplő adatokat és egy gombot jelenít meg, amellyel vissza görgethet a felhasználó az oldal tetejére (a gomb csak bizonyos oldalhosszúság felett jelenik meg).
- **HeaderNav:** az alkalmazás fejléce. Eltérő tartalma van a bejelentkezéshez kötött és a szabadon megtekinthető oldalaknál.
- **SidebarItem:** az oldalsó menü építőeleme.
- **TableHeaderkomponensnév:** egyedi tábla fejlécek kialakítása.
- **TableRowkomponensnév:** egyedi tábla formázások.
- **actions.js:** a Redux store-hoz tartozó action-öket tartalmazza.
- **scripts:** többször, több helyen használt függvények gyűjteménye.

3.2.3 A program telepítése

A frontrend oldal telepítésének a lépései:

1. A Git repository klónozása: git clone <https://github.com/verotomi/X-Bank-Frontend.git>
2. A programot tartalmazó mappa megnyitása: cd X-Bank-Frontend
3. Csomagok telepítése: npm install
4. Az alkalmazás futtatása: npm start

3.2.4 Tesztelési dokumentáció

A frontend oldal elkészítése folyamán komponens tesztelést, integrációs tesztelést és rendszer tesztelést is alkalmaztam. Az első két módszer ahhoz volt szükséges, hogy egy működőképes és vélhetően hibamentes program készülhessen el. A harmadik, a rendszerteszt viszont arra volt hivatott, hogy a programban megbúvó, eddig észre nem vett hibákat feltárja.

3.2.4.1 Komponens teszt

A React alkalmazás egymástól jól elkülönülő komponensekből áll, melyek HTML kódot adnak vissza. Ezek tesztelése során a megjelenő adatok helyessége mellett az elvárt megjelenési formákat is ellenőrizni kellett, ráadásul a reszponzív működés miatt egyszerre többféle képernyő típusnál.

Példa: új átutalási megbízást rögzítő komponens validálási tesztje. Az alkalmazás jelzi, ha egy szükséges mező nincs kitöltve, vagy a bevitt adat nem felel meg az előírásoknak, vagy a körülményeknek. Jelen esetben az alkalmazás az alábbiakat vizsgálja:

- a kedvezményezett nevét ki kell tölteni;
- a kedvezményezett neve nem lehet a beállított értéknél hosszabb;
- a partner bankszámlaszámát meg kell adni;
- a bankszámlaszám 2x8 vagy 3x8 számjegyből kell álljon, kötőjelekkel elválasztva;
- az átutalandó összeg nem lehet nagyobb, mint a kiválasztott bankszámla egyenlege;

- a közlemény kitöltése nem kötelező, de a hossza nem haladhatja meg a beállított értéket.

3. ábra - Egy komponens tesztelése

Ebben az esetben az elvárt módon hibaüzenetet jelent meg, mert a számlaszám formátuma nem volt megfelelő.

3.2.4.2 Integrációs teszt

Néha szükség volt az egyforma szinten lévő komponensek közötti kommunikációra, amit a React hierarchikus felépítése alpból nem kifejezetten támogat. Ilyenkor vagy a Redux store-t, vagy React Context-et használtam. Ezeknek a kommunikációknak a működését vizsgáltam:

- az egyik komponens által a másiknak átadott adatok megfelelően átkerülnek-e?
- másik komponensre történő váltáskor az előző komponens életciklusa megfelelően ér-e véget? Ezzel kapcsolatban a kliens felület tesztelése során egy ritkán előforduló, de súlyos hibát fedeztem fel. Amikor bankkártyák módosítását végző komponens volt látható a képernyőn és a felhasználó rákattintott az *Áttekintés* menüpontra, akkor az alkalmazás hibaüzenet kíséretében leállt. Csak is ebben a speciális esetben. Ezen a hibán nem segített az sem, ha a kérdéses komponens életciklusának a végén lefutó *componentWillUnmount* függvényben próbáltam lekezelni a hibát. Végül rájöttem, hogy az *Áttekintés* menüpont által meghívott komponens működése azzal kezdődik, hogy frissíti a Redux store-ban található *accounts* tömböt (technikailag valójában újra létrehozza). Ez a működés volt az, ami miatt leállt a kérdéses komponens, ugyanis egy olyan tömbön akart műveletet

végezni, ami nem létezett. A megoldás végül az lett, hogy a hitelkártya módosítást végző komponens megnyitásához beraktam egy vizsgálatot, ami ellenőrzi az *accounts* tömb meglétét.

3.2.4.3 Rendszer - rendszer integrációs teszt

Jelen esetben ez a kliens oldal és az API közötti kommunikáció vizsgálatát takarja. Mivel az API kérések többsége POST metódusokkal történik, így a hibakeresés során nagy segítséget nyújtott a Visual Studio Code ThunderClient nevű bővítménye, mellyel hatékonyan tudtam vizsgálni a kérésekre érkező válaszokat.

3.2.4.4 Rendszerteszt

Egyrészt – felhasználói szemmel – saját magam teszteltem a program rendszerszintű működését. Igyekeztem szélsőséges eseteket is letesztelni, mint például, hogy hogyan reagál a program az internetkapcsolat menet közbeni megszüntetésére, vagy a program futása közbeni váratlan böngésző-bezárásokra.

Emellett néhány ismerősömnek saját fiókot hoztam létre az alkalmazásban, és megkértem őket, hogy alaposan teszteljék a netbankot. Kértem, hogy minden apró észrevételüket jelezzék felém. Ennek során az olyan hibák kerültek felszínre, melyek mellett én – aki programozói oldalról is átlátja az alkalmazás működését – elsiklottam:

- Az alkalmazás pontatlanul jelenített meg a térképen a bankfiókok és a bankautomaták egy részét. Az oka az volt, hogy az adatbázisban hibásan, vagy pontatlanul voltak rögzítve a koordináták.
- A számítógépek, laptopok böngészőjében az oldalsó menü kényelmi okokból folyamatosan látható, de szükség esetén el tudja rejteni a felhasználó. Mobiltelefonon használva az alkalmazást, az oldalsó menü viszont lényegesen nagyobb részt foglal el a képernyőből és zavarta a kényelmes használatot, hogy minden menüpont-választás után be kellett zárni a menüt, új funkció használatához pedig megnyitni. A megoldás az lett, hogy képernyőfelbontás-függővé tettem a menü működését, és egy bizonyos felbontás alatt automatikusan bezáródik az oldalsó menü.
- Az egyszeri pénz átutalás során a program „*Sikeres rögzítés!*” üzenettel nyugtázta a művelet sikerességét. Ez az üzenet így félreérthető volt, a felhasználó azt hitte, hogy megtörtént a mentés és gondoskodnia kell a megbízás beküldéséről. Megoldásként kicseréltem az üzenetet erre: „*Az átutalás sikeresen megtörtént!*”. Ez ugyan nem egy klasszikus

értelemben vett hiba, de mégis egy éles helyzetben tud olyan bonyodalmat okozni, amit jobb elkerülni.

3.2.4.5 A program ismert hibái

Van néhány apró hiba, melyről tudomásom van, de az okokat nem sikerült megtalálni vagy kiküszöbölni:

- mobiltelefonon megnyitva az alkalmazást, a bejelentkezéshez nem kötött oldalak minimális mértékben görgethetők jobbra-balra, függetlenül attól, hogy ezt bármi is indokolná. A bejelentkezést igénylő programrészen ez a hiba nem jelentkezik.
- állandó megbízás módosításakor a *kedvezményezett mentése* gomb nem működik.
- a bejelentkezéshez kötött programrészen, ha egy aloldal van megnyitva és frissítem a böngésző tartalmát, akkor az minden esetben visszaugrik az áttekintő oldalra. Lényegtelen, és nem túl zavaró hiba ez, az elhárítására a token ellenőrzés módosítására lenne szükség.

3.2.5 Továbbfejlesztési lehetőségek

Egy olyan komplex program, mint egy netbank alkalmazás, számtalan továbbfejlesztési lehetőséget tartogat. Igyekeztem a lehetőségekhez képest minél több funkciót megvalósítani a program készítése során, ezek a hétköznapi bankolási szokások nagy részét lefedik.

Amennyiben a későbbiek folyamán a program továbbfejlesztésre kerül, szeretném még implementálni az alábbi funkciókat:

- tevékenységi napló vezetése a felhasználó által elvégzett műveletekről.
- csoportos beszédési megbízások kezelése.
- bankkártyák kezelésénél a különböző használati módokhoz (vásárlás, készpénzfelvétel, online használat) tartozó, naponta vagy havonta maximálisan megengedett tranzakciók számának a kezelési lehetősége.
- SMS aláírás, mely nagyobb biztonságot nyújt a bankszámlánkon lévő pénz védelméhez.
- tranzakciók csoportos kezelése: egyszerre több tranzakció rögzítése, majd azok csoportos beküldése. Ennek előfeltétele az SMS aláírás megléte.
- keresési lehetőség a bankszámlatörténetben szöveg, pontos összeg vagy összeghatár szerint.
- jelszóemlékeztető használata.
- adatok érkezésére való várakozás során animáció (vagy animált kurzor) megjelenítése.

4. Felhasználói dokumentáció

Az X Bank netbank alkalmazás egy fiktív bank internetes bankolási felületét valósítja meg. Teljes funkcionalitással rendelkezik, de értelemszerűen nem tartalmaz valós adatokat és az elindított tranzakciók sem kerülnek feldolgozásra élesben.

4.1 A program általános specifikációja

A netbank alkalmazás az alábbiakban ismertetett szolgáltatásokat nyújtja a Bankkal szerződésben lévő ügyfeleinek:

- a Bank fiókjainak a címeit és a fiókok nyitvatartási idejeit megjelenítő lista.
- a Bank pénzkidó automatáinak a listája.
- a Bank által kezelt valuták és devizák aktuális árfolyamai.
- a bankkal történő közvetlen vagy közvetett kapcsolatba lépési lehetőségek.
- bejelentkezéshez kötött funkciók:
 - bankszámlák áttekintése;
 - egyenlegek lekérdezése;
 - egyszeri és ismétlődő átutalások kezelése;
 - megtakarítások kezelése;
 - bankszámlatörténet, banki kivonatok;
 - hitelkártyák áttekintése, kezelése;
 - limitek módosítása;
 - kártya inaktívvá tétele;
 - kapcsolódó bankszámla módosítása;
 - kedvezményezettek /megbízási sablonok/használata;
- admin jogosultsággal áttekintési lehetőség a bank pénzeszközeinek az áttekintésére.

4.2 Rendszerkövetelmények

Az X Bank mobilbank alkalmazás korszerű, internetes böngészésre alkalmas PC, okos-telefon vagy tablet használatát igényli.

Az alkalmazás elsősorban asztali gépekre lett optimalizálva, melyek képernyőelrendezésre fekvő tájolású, így célszerű ilyen készüléket használni a program futtatásához. Ugyanakkor fokozott figyelemmel lett kialakítva a mobiltelefonos és a tabletes nézet is.

4.3 A program telepítése és indítása

A program különösebb telepítést nem igényel. Az alkalmazás használatához a böngészőben a <https://www.xbank.verovszki.eu/> oldalt kell megnyitni.

4.4 A program használata

A netbank használatát interaktív listák segítik: a táblázatok fejléceire kattintva az adatok csökkenő vagy növekvő rendezési sorrendben jelennek meg. Lehetőség van a listák szűrésére is különböző feltételek szerint. A rendezés és a szűrés tetszés szerint kombinálható. Némely listánál extra információk jelennek meg a lista adott eleméről, ha az egérmutatót az adott elem fölé húzzuk. A kurzor kinézetének a megváltozása jelzi az ilyen helyeket.

Az alkalmazás különböző biztonsági elemekkel van ellátva. A felhasználói jelszavak titkosítva vannak tárolva, így azokat senki nem tudja megnézni, nyilvánosságra sem tudnak kerülni. A program bizonyos ideig tartó inaktivitás után figyelmezteti a felhasználót, majd, ha továbbra sem történik reakció a felhasználó felől, akkor kilépteti őt az alkalmazásból, annak érdekében, hogy illetéktelenek ne tudjanak hozzáférni az ügyfél személyes pénzügyi adataihoz.

4.4.1 Bejelentkezés nélkül használható rész

Az alkalmazás weboldalának a felkeresésekor egy áttekintő oldal jelenik meg, a bankfiókok, automaták és a valuta- és devizaárfolyamok adataival.

Bankfiókok - Összes

Ír.sz.	Település	Cím	Mai nyitvatartás	Térkép
1051	Budapest	Deák tér 1.	00:00-23:59	
1137	Budapest	Szent István krt. 10.	00:00-00:00	
1044	Budapest	Főti út 4.	08:00-15:00	
1035	Budapest	Vörösvári út 22.	08:00-15:00	
1025	Budapest	Pusztaszeri út 67.	08:00-13:00	

Bankautomaták - Összes

Ír.sz.	Település	Cím	Bankjegy	Térkép
1052	Budapest	Deák tér 1.	Euro	
1137	Budapest	Szent István krt. 10.	Euro	
1065	Budapest	Andrássy út 29.	Forint	
1044	Budapest	Főti út 4.	Forint	
1035	Budapest	Vörösvári út 22.	Forint	

Valuta árfolyamok

Valuta	Vétel	Eladás	Érvényes
EUR	375.77	399.01	2022-04-22 08:04:03
USD	318.35	338.04	2022-04-22 08:04:03
GBP	424.52	450.78	2022-04-22 08:04:03
AUD	251.00	266.53	2022-04-22 08:04:03
BGN	182.03	193.29	2022-04-22 08:04:03

Deviza árfolyamok

Deviza	Vétel	Eladás	Érvényes
EUR	383.52	391.27	2022-04-22 08:04:03
USD	324.91	331.48	2022-04-22 08:04:03
GBP	433.28	442.03	2022-04-22 08:04:03
AUD	256.18	261.35	2022-04-22 08:04:03
BGN	185.79	189.54	2022-04-22 08:04:03

2021-2022 © Készítette Verovszki Tamás. Minden jog fenntartva!

4. ábra - A Főoldal

Az oldalsó menü az alábbi lehetőségeket tartalmazza:

- **Főoldal:** áttekintő oldal jelenik meg, a bankfiókok, automaták és árfolyamok adataival.
- **Bankfiókok:** a Bank fiókjainak a címeit és a fiókok nyitvatartási idejét jeleníti meg.
- **Bankautomaták:** a Bank pénzkiadó automatáinak a listáját tartalmazza.
- **Valuta árfolyamok:** a Bank által kezelt valuták és devizák aktuális árfolyamait nézhetjük meg.
- **Deviza árfolyamok:** a Bank által kezelt valuták és devizák aktuális árfolyamait tekinthetjük meg.
- **Kapcsolat:** ebben a menüben lehetőségünk van kapcsolatba lépni a bankkal.
- **Információk:** az alkalmazás adatai (készítője, verziószáma stb.) mellett jogi információk és adatvédelmi tájékoztató található itt.
- **Mobilbank:** a Bank Androidos telefonokra fejlesztett mobil-applikációját ismerteti.

A fejlécben találhatók:

- **Menü gomb:** az oldalsó menü elrejtése / megjelenítése. Kis képernyőméretnél a menü automatikusan eltűnik egy-egy funkció kiválasztása után annak érdekében, hogy ne foglalja feleslegesen a helyet.
- **Bejelentkezés gomb:** sikeres azonosítás után elérhetővé teszi a hozzáférést a személyes banki oldalunkhoz. A jogosultság igazolása után itt érhetők el a bankszámla adatok és azok egyenlegei. Itt lehet átutalási megbízásokat adni, kezelni a bankkártyánkat, lekötni a pénzünket, számlatörténetet megtekinteni és számlakivonatot letölteni. A teszteléshez használható belépési adatok:
 - felhasználói azonosító: verotomi123
 - jelszó: 8p56bZrK

4.4.1.1 Bankfiókok listája

A Bank fiókjainak a listájában látható, hogy az adott bankfiók éppen nyitva vagy zárva van-e? A könnyebb áttekinthetőség érdekében színekkel is meg van különböztetve a nyitva és a zárva állapot. A lista egy-egy elemére húzva az egeret megjelenik az érintett bankfiók teljes heti nyitvatartása. Ha a jobb oldalon látható térkép-ikonok valamelyikére kattintunk, akkor egy új lapon az adott bankfiók megjelenítésre kerül a térképen.

4.4.1.2 Bankautomaták listája

A bankautomaták listájában az automaták címe mellett az is látható, hogy milyen fajta bankjegyekhez tudunk hozzájutni belőlük. A jobb oldalon látható térkép-ikonok valamelyikére kattintva az adott bankfiók egy új böngészőablakban megjelenítésre kerül a térképen.

4.4.1.3 Valutaárfolyamok

Itt van lehetőségünk megtekinteni az X Bank legfrissebb valutaárfolyamait. A valuta 3 betűs rövid neve mellett feltüntetésre kerül az éppen aktuális vételi- és eladási árfolyam és a legutolsó árfolyamfrissítés dátuma és ideje is.

4.4.1.4 Devizaárfolyamok

Az előző menüpontnál szereplő leírással egyezik meg, de devizákra vonatkozik.

4.4.1.5 Kapcsolatfelvételi lehetőségek

A Bankkal történő különböző kapcsolatba lépési lehetőségeket találjuk ezen az oldalon.

4.4.1.6 Információk

A programnak ebben a részében informálódhatunk az aktuális program-verzióról, valamint az alkalmazás készítőjéről. A szolgáltató adatai, a jogi információk és az adatvédelmi tájékoztató is itt található meg.

4.4.1.7 Mobilbank

Egy rövid tájékoztatót tekinthetünk itt meg a Bank Androidos mobilbank alkalmazásáról, letöltési lehetőséggel.

4.4.2 Bejelentkezéshez kötött rész

Amennyiben a belépési felületen sikeresen azonosítottuk magunkat, akkor az áttekintő oldalra jutunk. Az itteni menüből lehet többek között átutalási megbízásokat indítani, kezelni a bankkártyánkat, lekötni a pénzünket, számlatörténetet megtekinteni és számlakivonatot letölteni. Az oldalsó menü az alábbi lehetőségeket tartalmazza:

- **Áttekintés:** áttekintő oldal jelenik meg, a bankszámláinkról, a megtakarításainkról és a bankkártyáinkról.
- **Egyenleg lekérdezés:** a Banknál nyitott bankszámláinkat és azok egyenlegeit tudjuk itt megtekinteni.

- **Egyszeri átutalás:** ezen az oldalon egyszeri átutalási megbízást tudunk benyújtani a Bank felé.
- **Állandó megbízások:** állandó átutalási megbízásokat tudunk itt megadni. A megbízások különböző intervallumokkal adhatók meg, valamint inaktív állapotúra is állíthatók.
- **Megtakarítások:** különböző időszakokra – és ezáltal különböző mértékű elérhető kamatokkal – lekötethetjük a Banknál lévő pénzünket. Lehetőségünk van a lejáratidő előtt is hozzáférni a lekötött pénzünkhöz, de ebben az esetben a Bank nem fizet kamatot.
- **Számlatörténet:** a bankszámláink korábbi tranzakcióit tudjuk itt megtekinteni lista formájában. Különböző szűrési lehetőségeket tudunk igénybe venni, valamint egy-egy tranzakcióról részletes információkat tudunk megjeleníteni, ha az egérmutatót a kérdéses tranzakció fölé mozgatjuk.
- **Bankszámla kivonatok:** az eddig elkészült bankszámla kivonatok listáját találjuk itt. Lehetőség van a kivonatok megtekintésére és letöltésére is.
- **Bankkártyák kezelése:** itt találjuk a Bank által kibocsájtott bankkártyáink listáját. Lehetőségünk van a kártyákhoz kapcsolódó vásárlási limitek módosítására, a kártyához kapcsolódó bankszámla módosítására, valamint a kártya – biztonsági okokból történő – átmeneti inaktiválására.
- **Kedvezményezettek:** azon partnereink adatait tudjuk itt eltárolni, akiknek gyakran szeretnénk pénzt utalni. Lehetőség van későbbi módosításra és törlésre is.

A fejlécben találhatók:

- **Menü gomb:** az oldalsó menü elrejtése / megjelenítése. Kis képernyőméretnél a menü automatikusan eltűnik egy-egy funkció kiválasztása után annak érdekében, hogy ne foglalja feleslegesen a helyet.
- **Profil gomb:** a nevünket tartalmazó gomb. Megnyomására legördül egy menü jelszóváltoztatási és kilépési lehetőségekkel, valamint az utolsó bejelentkezésünk időpontjának a megjelenítésével.

4.4.2.1 Áttekintés

Ez az oldal összefoglaló áttekintést ad aktív bankszámláinkról, megtakarításainkról és a bankkártyáinkról. A bal oldalon található menüsorból választhatjuk ki, hogy milyen műveletet szeretnénk végezni. A jobb felső sarokban található profil gombban a nevünk látható. Erre rákattintva megnézhetjük az utolsó bejelentkezési időnk, lehetőség van jelszóváltoztatásra vagy

kijelentkezésre. Jelszócsere esetén meg kell adni a jelenlegi jelszavunkat, majd ezután – biztonsági okokból – kétszer az új jelszót.

The screenshot displays the 'Netbank / Áttekintés' (Overview) page of X Bank. The left sidebar contains navigation links: Áttekintés, Egyenleg lekérdezés, Egyszeri átutalás, Állandó megbízások, Megtakarítások, Számlatörténet, Bankszámla kivonatok, Bankkártyák kezelése, and Kedvezményezettek. The main content area is divided into three sections: Bankszámlák, Megtakarítások, and Bankkártyák.

Bankszámlák

Számlaszám	Típus	Egyenleg	Pénznem	Állapot
99999999-11111111-11111111	Lakossági folyószámla	21,090,727.00	Forint	Aktív
99999999-11111111-22222222	Devizaszámla	304,740.00	Euro	Aktív
99999999-11111111-33333333	Megtakarítási számla	29,225,872.00	Forint	Aktív

Megtakarítások (Összes)

Típus	Forrásszámla	Lejárat	Kamat	Összeg	Pénznem	Állapot
Heti lekötés	99999999-11111111-11111111	2019-04-02	1.15 %	33.00	Forint	Feltörve
Heti lekötés	99999999-11111111-33333333	2019-04-02	1.15 %	5.00	Forint	Feltörve
Heti lekötés	99999999-11111111-33333333	2019-04-02	1.15 %	2.00	Forint	Feltörve
Heti lekötés	99999999-11111111-11111111	2019-04-02	1.15 %	55.00	Forint	Feltörve
Heti lekötés	99999999-11111111-11111111	2019-04-05	1.15 %	1,666.00	Forint	Feltörve

Bankkártyák

Kártyaszám	Típus	Érvényesség	Állapot	Kapcsolódó számlaszám	Elérhető összeg	Pénznem
------------	-------	-------------	---------	-----------------------	-----------------	---------

2021-2022 © Készítette Vereszkői Tamás

5. ábra - Áttekintés

4.4.2.2 Egyenleg lekérdezés

Ha ezt a menüpontot választjuk, akkor a bankszámláink listája jelenik meg, ahol az elérhető egyenlegek is feltüntetésre kerülnek.

4.4.2.3 Egyszeri átutalás

Itt indíthatunk új átutalási megbízást. A bankszámla, amelyről az utalást indítjuk, egy legördülő menüből választható ki. Az utalás kedvezményezettjének a megadásánál lehetőségünk van korábban elmentett partner kiválasztására. Ha új kedvezményezettnek utalunk pénzt, akkor pedig lehetőségünk nyílik a partner adatainak az elmentésére. Ha valamelyik adatot nem, vagy tévesen adtuk meg, arról a program tájékoztat minket. A megfelelő adatok megadása után a *Mentés* gomb megnyomásával tudjuk az átutalási folyamatot elindítani. Ha nincs elegendő pénz a számlánkon az utalás elindításához (elutalt összeg + utalási díj), akkor figyelmeztetést kapunk. A különböző pénznemű számlák közötti utalás kezelése a mindenkor aktuális banki árfolyam figyelembevételével történik.

4.4.2.4 Állandó megbízások

4.4.2.4.1 Állandó megbízások listája

Ha már korábban adtunk meg állandó megbízást, akkor azokat és azok részleteit láthatjuk itt. A listában szereplő gombok segítségével az adott állandó megbízást tudjuk módosítani vagy törölni.

4.4.2.4.2 Új állandó megbízás

Az itt megadandó adatok többsége megegyezik az egyszeri átutalás ismertetésénél szereplő adatokkal. Azokon kívül viszont meg kell adnunk az állandó megbízási sablon elnevezését, és az utalás gyakoriságát is. A gyakoriságnál az alábbiak közül tudunk választani:

- minden nap;
- minden hét adott napján;
- minden hónap adott napján.

Ezeket az adatokat legördülő listából választhatjuk ki. Itt is igaz, ami ez egyszeri átutalásnál, tehát a megbízást csak akkor tudjuk beküldeni a banknak, ha a megadott adatok helyesek. A hibás vagy hiányzó adatokról a program tájékoztat minket. Az állandó megbízási sablon elnevezése nem egyezhet meg egy már meglévő másik megbízás nevével.

4.4.2.5 Megtakarítások

4.4.2.5.1 Megtakarítások listája

Ha van aktív megtakarításunk, akkor azokat itt tudjuk megtekinteni. Láthatjuk a lekötött összeget, a lekötés módját (futamidejét), a lejárat idejét és a Bank által fizetendő kamat összegét. Valamely megtakarítás fölé húzva az egeret egy új ablak nyílik meg, melyben az megtakarítás alábbi adatai jelennek meg:

- a megtakarításhoz kapcsolódó bankszámlaszám;
- a megtakarítás típusa;
- a megtakarítás állapota;
- referenciaszám (tranzakció azonosító);
- a lekötés ideje;
- a lekötött összeg;
- a lejárat ideje;
- a lejáratkori összeg.

Az aktív megtakarításokat bármikor megszüntethetjük a mellettük található *Feltörés* gomb használatával. Ilyenkor egy megerősítő jóváhagyás után a lekötött összeg (kamat nélkül) azonnal visszakerül a folyószámlánkra, azonban a Bank a lejáratú idő előtti feltörés esetén kamatot nem fizet.

4.4.2.5.2 Új megtakarítás

A forrásszámla, a megtakarítás típusa (a lekötés ideje és kamata) és a lekötött összeg megadása után a *Mentés* gomb megnyomásával tudunk új megtakarítást létrehozni.

4.4.2.6 Számlatörténet

Ebben a programrészben áttekinthetjük a számláinkon történt korábbi pénzmozgásokat. Ha több számlánk van, lehetőségünk van választani közülük a legördülő menü segítségével. Rendelkezésünkre állnak szűrők is: választhatunk, hogy csak a kimenő, csak a bejövő vagy az összes tranzakciót szeretnénk-e látni? Továbbá be tudjuk állítani a számlatörténeti lista kezdő- és záródátumát. Alapesetben az utolsó 60 nap eseményei láthatók.

Netbank / Számlatörténet

Számlatörténet

Forrásszámla
Lakossági folyószámla - 99999999-11111111-11111111 - 22,090,860.00 Forint

Kezdődátum: 2022. 03. 26. Záródátum: 2022. 04. 26. Pénzmozgás iránya: ☒ Terhelések ☒ Jövéirások

Dátum / Idő	Tranzakció típusa	Partner	Közlemény	Összeg
2022-04-26 11:35:34	Átutalási költség	Verovszki Tamás	Verovszki Tamás 10044 Forint. Ref No: 284127130	-140.00 Forint
2022-04-26 11:35:34	Kimenő átutalás	Verovszki Tamás		-10,044.00 Forint
2022-04-26 00:38:29	Átutalási költség	Fenyvesi Erika	Fenyvesi Erika 100 Forint. Ref No: 816172379	-100.00 Forint
2022-04-26 00:38:29	Kimenő átutalás	Fenyvesi Erika		-100.00 Forint
2022-04-25 21:56:19	Bejövő átutalás	Baky Dániel	most teljesen profi minden, megvenném az otp helyében!!	1,000,000.00 Forint
2022-04-25 21:35:02	Bejövő átutalás	Véletlen-generált	Véletlen-generált megjegyzés	15,739.00 Forint
2022-04-25 21:25:18	Bejövő átutalás	Verovszki Tamás		607.00 Forint
2022-04-25 21:25:18	Átutalási költség	Verovszki Tamás	Verovszki Tamás 607 Forint. Ref No: 108388649	-102.00 Forint
2022-04-25 21:25:18	Kimenő átutalás	Verovszki Tamás		-607.00 Forint
2022-04-25 00:02:03	Állandó megbízás díja	X Bank Limited	Öltai Emese 5000.00 Forint	-120.00 Forint
2022-04-25 00:02:03	Állandó megbízás	Öltai Emese		-5,000.00 Forint

2021-2022 © Készítette Verovszki Tamás

6. ábra - Számlatörténet

Ha a megjelenített lista egyik elemére mozgatjuk az egérmutatót, akkor a kiválasztott tranzakció alábbi részleteit tudjuk megtekinteni:

- az a bankszámlaszám, amihez a tranzakció tartozik;
- a tranzakció összege;
- a tranzakció típusa;
- a tranzakció iránya;

- ellenoldali partner neve (ha van);
- ellenoldali partner számlaszáma (ha van);
- közlemény (ha van);
- referenciaszám (tranzakció azonosító);
- a könyvelés ideje.

4.4.2.7 Számlakivonatok

Az adott bankszámlához tartozó bankszámlakivonatok letöltésére és megtekintésére szolgáló programrész. A listában az eddig elkészült kivonatokat láthatjuk. A *Megtekintés* gombra kattintva egy új ablakban megnyílik a PDF kivonat, melyet le is tudunk tölteni.

4.4.2.8 Bankkártyák

Az itt megjelenő lista tartalmazza a Bank által kibocsátott bankkártyáinkat. Láthatjuk a kártyák számát, típusát, érvényességi idejét és aktuális állapotát.

7. ábra - Bankkártyák

A listában található gombok megnyomására megjelennek az adott kártya beállításai, melyeket módosítani tudunk. Lehetőségünk van megváltoztatni az alábbi adatokat:

- a kártyához kapcsolódó bankszámla;
- bankautomatákból történő készpénzfelvételi limit;
- bolti vásárlások maximális összege;
- interneten történő vásárlások maximális összege;

- a kártya állapota (aktív vagy inaktív).

A kártya állapotának az általunk történő inaktív állapotba helyezése nagy mértékben növeli a számlánkon lévő pénz biztonságát. Ha ugyanis inaktív állapotú a kártya, akkor nem lehetséges visszaélést elkövetni egy esetlegesen elloptott kártyával (vagy az elloptott kártyaadatokkal). Elegendő csupán csak arra az időszakra aktívvá tenni a kártyát, amíg mi saját magunk éppen használni szeretnénk.

4.4.2.9 Kedvezményezettek

4.4.2.9.1 Kedvezményezettek listája

Ha már korábban kezdeményeztünk átutalást és az ott megadott partner adatait rögzítettük, akkor azok – egyfajta sablonként – ebben a programrészben tekinthetők meg. A listában található gombok segítségével tudjuk az adott partner adatait módosítani vagy törölni a listából.

4.4.2.9.2 Új kedvezményezett

A sablon neve, a kedvezményezett neve és számlaszáma megadása után az újonnan felvitt partner bekerül a kedvezményezettek listájába. A sablon neve nem egyezhet meg egy már meglévő sablon nevével.

4.4.2.10 Profil gomb

Ezzel a nevünket tartalmazó gombbal érjük el azt a legördülő menüt, amely a jelszóváltoztatási vagy kilépési lehetőséget, valamint az utolsó bejelentkezési idő megjelenítését tartalmazza.

4.4.2.10.1 Jelszó változtatás

Ha úgy érezzük, hogy valakinek a tudomására jutott a jelszavunk, vagy csak biztonsági okokból időnként meg szeretnénk változtatni, akkor azt itt tudjuk azt megtenni. Először – biztonsági szempontból – a régi jelszavunkat kell megadnunk, majd ezt követően az újat. A véletlen elírási lehetőségének a csökkentése érdekében az új jelszót ismételtlen meg kell adnunk. Ha a megadott kódok egyeznek, akkor az új jelszó rögzítésre kerül és innentől kezdve azt tudjuk használni a bejelentkezések során.

4.4.2.10.2 Kilépés

Biztonsági okokból célszerű minden esetben kilépni az alkalmazásból, ha már nem kívánjuk használni. Az alkalmazásban beépített biztonsági elemek vannak, melyek által bizonyos hosszúságú inaktivitás esetén automatikusan megtörténik a felhasználó kijelentkeztetése.

5. Összegzés

Már a vizsgaremek készítésének az elején tudtam, hogy nagy fába vágom a fejszémet, ugyanis mind a szerver oldali, mind a kliens oldali részmeglehetősen összetett. Mivel közel áll hozzám a banki téma, így nagy lelkesedéssel csináltam a vizsgaremeket az elejétől a végéig, amikor csak lehetőségem volt rá. Összességében nagy kihívást jelentett ez a feladat, rengeteg mindent kellett megismernem és megtanulnom hozzá. Sokszor voltak kisebb-nagyobb elakadások, de végül – hosszabb-rövidebb időn belül – mindig sikerült megoldást találnom rájuk. Csúppán néhány apró hiba maradt a programban, melyek a program mindennapi használatát nem befolyásolják. Ezeket a dokumentáció megfelelő helyén feltüntettem.

Érdekes volt az is, hogy nem egy kész feladatot kaptam leírásokkal és specifikációkkal, hanem nekem, saját magamnak kellett mindent kitalálnom. Nekem kellett felépítenem az egész alkalmazást a saját ötleteim, elképzeléseim szerint. Számítalan hasznos tapasztalattal gazdagodtam, melyeket később kamatoztatni fogok tudni. A sok-sok hajnalba nyúló munka viszont meghozta a gyümölcsét, jóleső érzés volt látni, ahogy kezd összeállni a program és úgy működik és úgy néz ki, ahogy én szeretném. Sok új ötlet jött a fejlesztés során, melyeket kezdetekben nem terveztem megvalósítani, de ezek implementálása hozzájárult ahhoz, hogy végül egy olyan munkát tudok kiadni a kezemből, melyre büszke tudok tekinteni.

Összességében elmondhatom, hogy amellet, hogy sikerült a React és a Slim4 keretrendszert mélyebbről is megismernem, sikerült nagyobb rutint szereznem mind a Javascript, mind a PHP nyelv területén. Tapasztalatot szereztem továbbá egy komplex alkalmazás kifejlesztésében is, ahol többféle programozási nyelvet kellett használni és meg kellett oldani a frontend és a backend közötti kommunikációt is.

Ezúton köszönöm a támogatást és a messzemenő segítőkészséget Budaházi Bence, Czinkóczi Tamás és Halász Gábor tanárainnak.

Irodalomjegyzék, forrásjegyzék

Alkalmazásban használt ikonok, ábrák: Bootstrap Icons [Online]. - 2022. április 26. - <https://icons.getbootstrap.com/>.

BootstrapMade: Free Bootstrap Themes and Website Templates | BootstrapMade [Online]. - 26.. - <https://bootstrapmade.com/>.

PDF dokumentumok generálása: FPDF [Online]. - 2019. április 9. - <http://www.fpdf.org/>.

Programozási technikák, megoldások: Stack Overflow - Where Developers Learn, Share, & Build Careers [Online]. - 2019. április 9. - <https://stackoverflow.com/>.

React Javascript könyvtár: React – JavaScript könyvtár felhasználói felületek létrehozásához [Online]. - 2022. Április 26. - <https://hu.reactjs.org/>.

React Redux: React Redux [Online]. - 2022. Április 26. - <https://react-redux.js.org/>.

Slim4 keretrendszer: Slim 4 Documentation - Slim Framework [Online]. - 2022. Április 26. - <https://www.slimframework.com/docs/v4/>.

Ábrajegyzék

1. ábra - a MariaDB adatbázis felépítése és a táblák közötti kapcsolatok.....	9
2. ábra - Sikertelen bejelentkezési kísérlet	25
3. ábra - Egy komponens tesztelése.....	32
4. ábra - A Főoldal.....	36
5. ábra - Áttekintés	40
6. ábra - Számlatörténet.....	42
7. ábra - Bankkártyák	43