Java alkalmazások

Tervdokumentáció

Gyakorlat

|  |  |
| --- | --- |
| **Készítették** | **Feladatkör** |
| Szabó Ferenc | Frontend, Login, Regisztráció, GitHub |
| Nagy Erik | Lekérdezés, Kapcsolat, Üzenetek, RESTful API |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Dokumentum azonosító: | L-K-GINFBAL-JAVALKA-1-EA01 |
| Verzió: | 1.1 |
| Dátum: | 2024. 11. 29. |

|  |
| --- |
| **Lektorálta** |
| Nagy Erik |

**©** Ferinek, nem tetszett a copyright jogi üzenet ezért valamit raktam ide, hogy szép legyen a dokumentáció.

Kellemes ebédszünetet, húsvétot, karácsonyt, nyaralást, és egyéb dolgokat kíván a „Csábító Tea Party”.

Tartalomjegyzék

[**1.** **Bevezetés** 4](#_Toc183782158)

[**2.** **Fejlesztési környezet** 4](#_Toc183782159)

[*2.1* *A fejlesztői környezet: IntelliJ IDEA és Spring boot* 4](#_Toc183782160)

[*2.2* *Verziókövetés: GitHub* 5](#_Toc183782161)

[*2.3* *Használt technológiák* 6](#_Toc183782162)

[**3** **Design** 7](#_Toc183782163)

[*3.1* *Általános kinézet és működés* 7](#_Toc183782164)

[3.1.1 Az oldal fejléce: 7](#_Toc183782165)

[3.1.2 Főoldal: 7](#_Toc183782166)

[3.1.3 Kapcsolatok: 8](#_Toc183782167)

[3.1.4 Üzenetek: 8](#_Toc183782168)

[3.1.5 Lekérdezés: 9](#_Toc183782169)

[3.1.6 Bejelentkezés 9](#_Toc183782170)

[3.1.7 Regisztráció 9](#_Toc183782171)

[3.1.8 Az oldal lábléce: 10](#_Toc183782172)

[*3.2* *Reszponzív megjelenés* 11](#_Toc183782173)

[**4.** **Fejlesztési keret: Frontend, Backend és Adatkezelés** 12](#_Toc183782174)

[*4.1* *Frontend* 12](#_Toc183782175)

[*4.2* *Backend* 13](#_Toc183782176)

[*4.3* *Adatkezelés* 27](#_Toc183782177)

[*4.4* *Tesztek és kiegészítő információk* 28](#_Toc183782178)

# **Bevezetés**

Jelen tervdokumentáció a „Java alkalmazások” elnevezésű tantárgy, a **Csábító Tea Party** csapataáltal fejlesztés alatt álló számítástechnikai projekt, **weboldal** fejlesztői környezetét, koncepcióját, design elemeinek tervezett megvalósítását és fejlesztési tevékenységeit ismerteti.

# **Fejlesztési környezet**

## *A fejlesztői környezet: IntelliJ IDEA és JavaFX*

Az **IntelliJ IDEA** egy professzionális és népszerű fejlesztői környezet (IDE), amelyet a JetBrains fejlesztett ki. Széles körben ismert a fejlett funkcióiról, mint például a kód intelligens kiegészítése, hibák automatikus felismerése, és a mély integrációja a modern programozási nyelvekkel és keretrendszerekkel. Az IntelliJ IDEA kiemelten támogatja a Java nyelvet és kapcsolódó technológiákat, beleértve a Spring Boot keretrendszert is. Ez az IDE különösen előnyös nagyobb projektekhez, mivel hatékony eszközkészletet biztosít a kód navigációhoz, refaktoráláshoz és hibakereséshez.

A JavaFX egy modern, nyílt forráskódú Java GUI (grafikus felhasználói felület) keretrendszer, amelyet elsősorban asztali és beágyazott alkalmazások fejlesztésére használnak. A JavaFX a Java alapú alkalmazásokhoz fejlett grafikai képességeket, animációkat és multimédiás támogatást biztosít, miközben a hagyományos Swing technológiát is kiegészíti és túlszárnyalja.

A „Java alkalmazások” tantárgyhoz tartozó projekt fejlesztése során csapatunk az IntelliJ IDEA és a JavaFX használatát részesíti előnyben. Ez a fejlesztői környezet egy hatékony és rugalmas alapot biztosít az alkalmazásunk gyors és megbízható elkészítéséhez. Az IntelliJ IDEA által nyújtott fejlett fejlesztési funkciók, valamint a Spring Boot könnyű konfigurálhatósága és robusztussága lehetővé teszik számunkra, hogy hatékonyan kezeljük a projekt követelményeit. Az eszközök kombinációja zökkenőmentes munkafolyamatot biztosít, lehetővé téve a csapat számára, hogy a projekt technikai kihívásaira összpontosítson, miközben minimalizálja a fejlesztési környezet karbantartásával járó terheket.

## *Verziókövetés: GitHub*

A **GitHub**, mint vezető szoftverfejlesztési platform, lehetővé teszi a csapat számára, hogy hatékonyan kezelje a kódbázist, nyomon követhesse a változásokat, és elősegítse a kollaboratív munkát azáltal, hogy egy központosított helyen tárolja a projekt fájljait és dokumentációját. Az integrált verziókezelési rendszer előnyei közé tartozik a fejlesztési folyamatok nagyfokú automatizálása, a visszaállíthatóság, valamint a különböző fejlesztési ágak (branches) kezelése, ami lehetővé teszi a csapat számára, hogy párhuzamosan dolgozzon különböző funkciókon. A GitHub használatával a Csábító Tea Party csapata kihasználhatja a kódmegosztás és a közösségi hozzájárulás előnyeit is, így gyorsítva a fejlesztési ciklusokat és javítva a projekt átláthatóságát és nyomon követhetőségét.



## *Használt technológiák*

A Csábító Tea Party csapata a webfejlesztés során számos modern és bevált technológiát alkalmaz, hogy biztosítsák a webalkalmazásuk gyors, biztonságos és felhasználóbarát működését:

A **JavaFX** Scene Graph architektúrája lehetővé teszi összetett és dinamikus interfészek létrehozását, miközben a CSS és a FXML használatával szétválasztható a megjelenés és a logika. Az integrált médiakezelés, a 2D/3D grafikai támogatás, valamint az eseményvezérelt programozás egyszerűbbé teszi a fejlesztők számára a modern és intuitív felhasználói élmény kialakítását. A JavaFX a Java platform szoros integrációját kihasználva skálázható és platformfüggetlen megoldást kínál, amely Windows, macOS és Linux rendszereken egyaránt futtatható.

**Hibernate:** Egy népszerű objektum-relációs leképezési (ORM) eszköz, amely lehetővé teszi, hogy a Java osztályok közvetlenül kapcsolódjanak az adatbázis tábláihoz. Ez csökkenti az adatbázis-műveletek kódolási komplexitását.

**MySQL:** Nyílt forráskódú relációs adatbázis-kezelő rendszer, amely hatékony adattárolási megoldásokat nyújt. A Spring Data JPA segítségével könnyen integrálható a Java alapú alkalmazásokba.

**Maven:** Az alkalmazás építési folyamatainak kezelésére használt eszköz, amely automatikusan kezeli a függőségeket, és biztosítja a projekt struktúrájának tisztaságát és karbantarthatóságát.

Ezek a technológiák együttesen biztosítják a fejlesztési folyamat hatékonyságát, az alkalmazás megbízhatóságát, valamint a felhasználói élmény magas színvonalát.

# **Menüpontok**

## *Adatbázis menü*

Az adatbázis menü almaenüjei gyakorlatilag egy CRUD menünek felelnek meg, ahol az összekapcsolt adatbázisból tud a felhasználó lekérdezni, elemeket módosítani, törölni, vagy akár új elemet létrehozni.

### 3.1.1 A kezdőoldal:

**A screenshot of a computer

Description automatically generated**

Az alkalmazás egy egyszerű szöveggel indít, ami bíztatja, hogy próbálja ki annak különböző funkcióit.

### 3.1.2 Olvas:

Az Adatbázis menü Olvas oldalára kattintva rögtön egy, táblázat tárul szemünk elé, amit az adatbázisból tölt be a program.

A screenshot of a computer

Description automatically generated

### 3.1.3 Olvas2 Ide navigálva egy évszámra keresve a táblázat megmutatja nekünk, hogy az adott évben hány olyan húzás volt, ami megfelelt az általünk beállított szűrési kritériumnak.

A screenshot of a computer

Description automatically generated

### 3.1.4 Ír

Itt új lottósorsolást, vehetünk fel az adatbázisunkba. Ehhez meg kell adnunk a nyerőszámokat, a sorsolás évét, hónapját, valamint a nyertesek számát, illetve az egy főre jutó összegeket. A lenti szövegdoboz az űrlap által generált SQL parancsokat jeleníti meg.

A screenshot of a computer

Description automatically generated

### 3.1.5 Módosít

Módosítani tudjuk a húzás tábla egyik sorát, A kért paraméterek:   
A módosítandó elem id-je, a húzás éve, hónapja (Ezeket egyébként automatikusan kitölti a program a már létező adatokkal) valamint a nyertesek száma, nyeremények összege. A lenti szövegdoboz az űrlap által generált SQL parancsokat jeleníti meg.

A screenshot of a computer

Description automatically generated

### 3.1.6 Töröl:

Az adatbázisból betölti a ComboBox elem-be a lehetséges id-ket, amit kiválasztva a törlés gombbal tudunk törölni.

A screenshot of a computer

Description automatically generated

## *Soap kliens menü*

A SOAP kliens menü célja bemutatni a MNB SOAP api-jának a programba való integrációját.

### 3.2.1 Letöltés

A Soap kliens menüpont „Letöltés” almenüje nem rendelkezik önálló nézettel. Rákattintva felugrik egy ablak, aminek a segítségével el tudjuk menteni számítógépünkre a generált MNB.txt fájlt, amiben a MNB SOAP apija által lekért adatok találhatóak meg.

### 3.2.2 Letöltés2

Itt a MNB által szolgáltatott adatok alapján lehet lekérdetzni egy-egy valuta árfolyamát a kiválasztott napon. Az ereményt egy MNB.txt fájba tudjuk elmenteni. A rádió gombra kattintva a program átkonvertálja az emberek által nehezen olvasható XML formátumot egy könnyebben érthető formára.

A screenshot of a computer

Description automatically generated

### 3.2.2 Grafikon

Egy látványos grafikonon jeleníti meg a kiválasztott valuta árfolyamváltozását a két megadott dátum között.

A screen shot of a graph

Description automatically generated

## *Párhuzamos menü*

Ennek a menüelemnek egyetlen almenüje van „Párhuzamos végrehajtás” névvel ellátva.

Feladata a párhuzamos programvégrehajtás bemutatása. Az oldalon két számláló jelenik meg, valamint három gomb.

Az indítás gombra kattintva az első számláló másodpercenként, a második 2 másodpercenként növekszik egyel. Ez a szünet/folytatás gombbal ideiglenesen szüneteltethető, valamint a leállítás gombbal megállítható és nullázható.

A screenshot of a computer

Description automatically generated

## *Forex menü*

Az OANDA api-ját használva tudunk a különböző almenükben adatokat lekérni, poziciót nyitni, zárni, valamint grafikonon vizualizálni egy-egy devizapár historikus árait.

### Számlainformációk

A menüpontra kattintva láthatjuk az OANDA felhasználónk különböző adatait egy táblázatba rendezve.

A screenshot of a computer

Description automatically generated

### Aktuális árak

Egy előre megadott devizapár listából kell kiválasztanunk egyet, aminek le tudjuk kérni az aktuális értékét.

A screenshot of a computer

Description automatically generated

### Historikus árak

Egy kiválasztott devizapárra vonatkazó értékadatokat tudunk megjeleníteni egy grafikonon a két megadott dátum között.

A screenshot of a computer

Description automatically generated

### Pozició nyitás

Ez az osztály egy grafikus felületen keresztül lehetővé teszi, hogy a felhasználó egyszerűen válasszon ki devizapárt, irányt és mennyiséget, majd az OANDA platformon egy piaci megbízást küldjön ezekkel a paraméterekkel. (A teszt időpontjában zárva van a piac, ezért sikertelen a nyitás)

A screenshot of a computer

Description automatically generated

* + 1. Pozició zárás

Az aktuális nyitott pozicióinkat zárhatjuk ennek a menüpontnak a segítségével. A program az OANDA api segítségével lekérdezi a nyitott poziciókat és betölti a legördülő listába az id-ket.

A screenshot of a computer

Description automatically generated

### Nyitott poziciók

Egy táblázatba betölti az nyitott pozicióinkat.

A screenshot of a computer

Description automatically generated

### .

# **Fejlesztési keret: Frontend, Logika és Adatkezelés**

## *Grafikus megjelenés (Frontend)*

**Feladata:** A weboldal elsődleges célja és feladata, hogy bemutassa a projektmunkánkat, valamint információkat nyújtson a feladathoz kiadott adatbázisból, valamint API-okkal lekérdezhető adatokról.

**Felépítése:** A program felhasználói felületének megjelenését a JavaFX keretrendszerben az FXML fájlok határozzák meg. Ezek a fájlok hasonlóak a HTML dokumentumokhoz, mivel itt definiáljuk a felület elrendezését és a megjelenített elemeket. Az FXML lehetővé teszi a vizuális komponensek deklaratív leírását, ami elősegíti az átláthatóságot és a fejlesztés egyszerűségét.

**Működése:** Az FXML elemek mindegyike egyedi azonosítóval (ID) látható el, amely kulcsfontosságú szerepet játszik az alkalmazás működésében. Ezek az azonosítók biztosítják, hogy a vezérlőosztály (controller) hozzáférjen az adott UI elemhez, és annak viselkedését vagy megjelenését a program logikája alapján módosítani tudja. Az azonosítók segítségével a vezérlő képes például frissíteni egy szöveges címkét, kezelni egy gomb eseményeit, vagy dinamikusan módosítani egy lista tartalmát. Ez az elválasztás elősegíti a felhasználói felület és az üzleti logika közötti tiszta kapcsolatot, javítva ezzel a program átláthatóságát és karbantarthatóságát.

## *Az alkalmazás logikája*

Az alkalmazás felépítése az MVC (Model-View-Controller) tervezési minta elveit követi, amelyet gyakran alkalmaznak webes rendszerekben is. Ez az architektúra tiszta elválasztást biztosít a felhasználói felület, az alkalmazás logikája, valamint az adatok kezelése között, megkönnyítve ezzel a kód karbantarthatóságát és bővíthetőségét.

Az alkalmazás indításakor a Main osztály fut le elsőként. Ez az osztály hasonló szerepet tölt be, mint egy weboldal frontend vezérlője (front controller), hiszen az induláskor ez felel a rendszer alapvető beállításainak elvégzéséért. Felparaméterezi a felhasználói felület alapvető elemeit, például a menüstruktúrát, amelyeken keresztül a felhasználó navigálhat az alkalmazás különböző funkciói között.

A Main osztály emellett közvetlenül meghívja a vezérlőosztályok (controller-ek) releváns metódusait, amelyek felelősek az adott oldalak betöltéséért és a hozzájuk tartozó üzleti logika végrehajtásáért. A vezérlők kapcsolódnak az FXML fájlokban definiált nézetekhez (View), és az FXML elemek egyedi azonosítói (ID) segítségével képesek manipulálni a felhasználói felületet. Például a vezérlőosztályok dinamikusan módosíthatják az UI elemeket, kezelhetik az interakciókat, és biztosítják, hogy a megjelenítés összhangban legyen az alkalmazás aktuális állapotával.

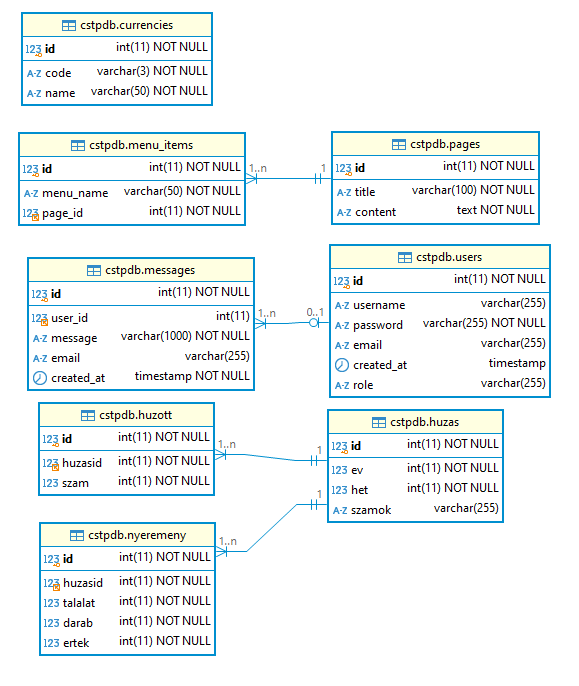
Ez az architektúra biztosítja a program modularitását és tesztelhetőségét, hiszen az alkalmazás különböző rétegei – a modell (Model), a nézet (View) és a vezérlő (Controller) – egymástól függetlenül fejleszthetők és bővíthetők. A Main osztály így egyfajta központi koordinátorként működik, amely összefogja a nézeteket és a vezérlőket, valamint biztosítja az alkalmazás működéséhez szükséges alapvető összeköttetéseket.

## *Adatkezelés*

Az adatkezelés során a nethely.hu által szolgáltatott **MySQL** alapú adatbázist használjuk.

**Feladata:** Az adatbázis elsődleges feladata egy olyan megbízható háttérszolgáltatás biztosítása, amely lehetővé teszi a felhasználók regisztrációját és bejelentkezését, valamint azonosításukat az űrlapok kitöltésekor. Emellett támogatja a különböző jogosultságok kezelését.

**Felépítése:** Az adatbázis pontos felépítése az alábbi ER diagrammon tekinthető meg:



Az adatbázismodell magában foglal egy felhasználók táblát, amely az alapvető felhasználói adatokat, mint például felhasználónév és jelszó tárolja, valamint egy külön email táblát az űrlapokkal kapcsolatos adatok kezelésére. Ez a struktúra lehetővé teszi az egyszerű felhasználói műveletek, mint a regisztráció és bejelentkezés, valamint az űrlapokon keresztüli kommunikáció kezelését.

# **Mappaszerkezet, fájlok tartalma**

A projekt logikájának nagy része a main/java/com mappastruktúrában helyezkedik el, ahol az alkalmazás különböző funkcióit és moduljait tartalmazó csomagok találhatók.

* Az example.javaeloadasbeadando mappa tartalmazza a program működéséért felelős fő logikai modulokat, amelyek az alkalmazás alaprendszerét és üzleti folyamatait kezelik.
* Az mnbapi mappában találhatók a Magyar Nemzeti Bank SOAP API-jából generált osztályok, amelyek lehetővé teszik a banki adatokkal történő integrációt és kommunikációt.
* Az oanda.v20 mappa az OANDA API-hoz való hozzáférést támogató osztályokat foglalja magában, biztosítva az API-val történő kereskedési műveletek zökkenőmentes végrehajtását.

A projekt egy másik kulcsfontosságú része a **resources** mappa, amely különféle erőforrásokat tartalmaz, például:

* A config.properties fájlt, amely az alkalmazás érzékeny információit tárolja, például az adatbázis-hozzáférési adatokat és az API-kulcsokat. Ezzel biztosítjuk az alkalmazás konfigurációs adatainak elkülönítését és védelmét.
* A pages mappa az alkalmazás megjelenéséért felelős FXML fájlokat tartalmazza, amelyek a felhasználói felület struktúráját és elrendezését határozzák meg.

Ez a mappastruktúra világosan elkülöníti a projekt különböző funkcionális részeit, ezáltal megkönnyíti az alkalmazás fejlesztését, karbantartását és bővítését.

**GitHub projekt elérhetőség:**   
**https://github.com/irkm813/java\_gyak\_beadando**