**Színházi Jegy Értékesítő Program Fejlesztői Dokumentáció**

**1. Bevezetés** A Színházi Jegy Értékesítő Program egy egyszerű, konzol alapú alkalmazás, amely lehetővé teszi a felhasználók számára színházi jegyek foglalását és kezelését. Az alkalmazás célja, hogy egy intuitív felhasználói felületen keresztül biztosítsa a jegyfoglalásokat, valamint lehetőséget nyújtson a már meglévő foglalások módosítására és törlésére.

**2. Felasznált technológiák**

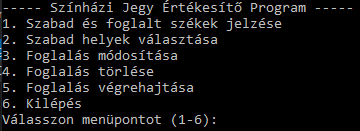
Az alkalmazás a következő technológiákat és eszközöket használja:

* C# programozási nyelv.
* .NET keretrendszer.
* XML adatkezelés.
* Konzol alapú felhasználói felület.

**3. Funkcionális Specifikációk**

Az alkalmazás a következő funkciókat teszi lehetővé:

* Szabad és foglalt helyek jelzése: A felhasználó megtekintheti a színház nézőterét, ahol a foglalt és szabad helyeket egyaránt megjeleníti a rendszer.
* Szabad helyek választása: A felhasználó szabad helyeket választhat, majd foglalást hajthat végre ezeken a helyeken.
* Foglalás módosítása: A felhasználó módosíthatja a már elvégzett foglalásokat, választhat más szabad helyeket vagy akár a foglalás nevét is megváltoztathatja.
* Foglalás törlése: A felhasználó törölheti a korábban elvégzett foglalásokat.
* Foglalás végrehajtása: Kilistázza az összes elmentett foglalást.
* Adatmentés és -betöltés: Az alkalmazás képes elmenteni és betölteni az elvégzett foglalásokat XML formátumban.



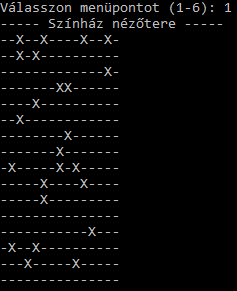
**3.1 Hibakezelés és Fejlesztési Javaslatok**

Az alkalmazás jelenleg nem tartalmaz kiterjedt hibakezelést, és csak alapvető ellenőrzéseket végez a felhasználói bemenetekre. Fejlesztéskor érdemes lenne a következőket figyelembe venni:

* **Bemeneti Ellenőrzések**: Ellenőrizd a felhasználói bemeneteket, hogy megakadályozd a hibás vagy érvénytelen adatokat.
* **Kiterjedt Hibakezelés**: Implementálj részletes hibakezelést és visszajelzéseket a felhasználó számára, hogy segítsd a hibák azonosítását és kezelését.
* **Skálázhatóság**: Mérlegelj olyan funkciók hozzáadását, amelyek lehetővé teszik az alkalmazás bővítését és kiterjesztését, például online fizetési rendszerek integrálását vagy helyfoglalások pontosabb kezelését.

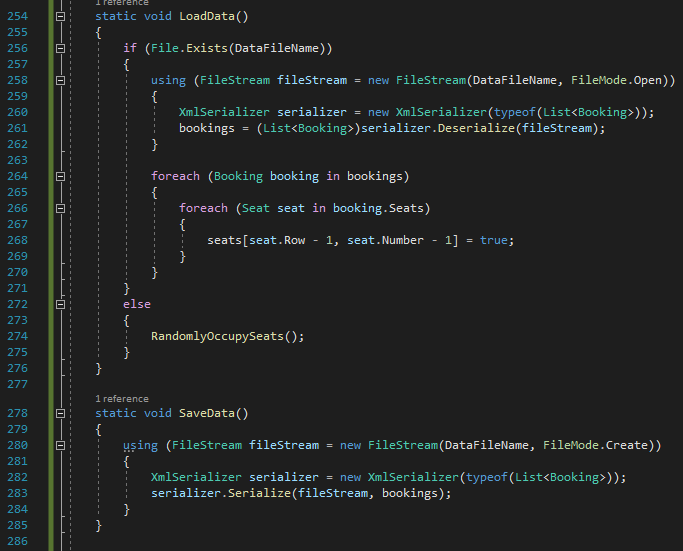
**3.2 Tesztelés**

Az alkalmazás funkcióinak és részeinek tesztelése kulcsfontosságú a stabil és megbízható működés érdekében. Ajánlott unit tesztek, integrációs tesztek és elfogadási tesztek írása az alkalmazás különböző részeihez. Ezzel garantálható, hogy az alkalmazás különböző szempontból megfeleljen a specifikációknak és helyesen működjön.



**3.3 Biztonság**

Ha az alkalmazás kezeli a felhasználók személyes vagy érzékeny adatait, fontos biztonsági intézkedéseket bevezetni annak érdekében, hogy megvédjük ezeket az adatokat illetéktelen hozzáféréstől vagy támadásoktól. Ez magában foglalhatja az adatok titkosítását, az autentikáció és autorizáció implementálását, valamint az adatvédelmi szabályozások betartását.



**4. További Funkciók**

* Mielőtt az alkalmazás késznek tekintenénk, érdemes átgondolni, hogy vannak-e további funkciók vagy javítások, amelyeket hozzá lehetne adni az alkalmazáshoz. Például online jegyvásárlási lehetőségek, jegyárak megjelenítése vagy akár helyfoglalások testre szabása lehet néhány további ötlet.
* Ezek a kiegészítő elemek és figyelembe veendő tényezők tovább gazdagítják a fejlesztői dokumentációt, és segítenek a fejlesztési folyamat zökkenőmentes és sikeres lebonyolításában.

**5. Dokumentáció Frissítése**

Az alkalmazás fejlesztése során fontos, hogy a dokumentációt is frissen tartsuk. Minden változtatást és frissítést meg kell dokumentálni a forráskódban és a fejlesztői dokumentációban is. Ez biztosítja, hogy a dokumentáció mindig aktuális legyen és megfeleljen az aktuális állapotnak.