**Projekt Leírás**  
A projekt célja egy olyan modern időpontfoglalási rendszer létrehozása, amely hatékonyan támogatja egy férfi hajszalon mindennapi működését. Az applikáció fő feladata az időpontfoglalások nyilvántartása és kezelése egy mátrix-alapú megoldással, amely három napra osztva, félórás bontásban tárolja az ügyfelek foglalásait.

**1. Funkcionális követelmények (*Functional Requirements*)**

Eset 1.

Mint vendég, szeretném megtekinteni az elérhető időpontokat, hogy kiválaszthassak egy számomra megfelelőt.  
Acceptance Criteria:

* Az időpontok listája tartalmazza a dátumot (hétfő, kedd, szerda) és az időszakokat (08:00–15:30).
* A szabad és foglalt időpontok egyértelműen elkülönülnek.

Eset 2.

Mint vendég, szeretném lefoglalni a számomra megfelelő időpontot a nevem megadásával, hogy biztosítsam a helyemet a fodrászatban.  
Acceptance Criteria:

* Az időpont foglalható, ha "szabad" állapotban van.
* Ha az időpont foglalt, a rendszer hibaüzenetet jelenít meg.

Eset 3.

Mint fodrászat tulajdonosa, szeretném, ha a foglalások fájlban mentésre kerülnének, hogy később is elérhetők legyenek.  
Acceptance Criteria:

* A foglalások egy .txt fájlba kerülnek mentésre.
* A fájlban a szabad időpontok "Szabad", a foglalt időpontok pedig a foglaló nevét tartalmazzák.

2**. Nem funkcionális követelmények (*Non-Functional Requirements*)**

1. A rendszer képes legyen kezelni legalább 3 napnyi (hétfő, kedd, szerda) időpontot.
2. Az alkalmazás legyen stabil és kezelje az esetleges fájlműveleti hibákat.
3. A felhasználói interfész legyen áttekinthető, és támogassa a könnyű navigációt.

**Rendszer Jellemzői**

1. **Időpontfoglalás kezelése**  
   A rendszer lehetővé teszi az ügyfelek számára, hogy szabad időpontokat keressenek, foglaljanak, és megtekintsék a jelenlegi foglalási állapotot.
2. **Fájlalapú tárolás**  
   A foglalásokat egyszerű szövegfájlban tárolja és onnan tölti be, így könnyű a rendszer integrációja és karbantartása.
3. **Felhasználói interakció**  
   Az alkalmazás egyszerű konzolos felületen keresztül működik, amely biztosítja az ügyfélbarát használatot. A felhasználók névvel regisztrálhatják foglalásaikat.
4. **Foglalások megjelenítése**  
   A foglalási táblázat vizuális megjelenítése segíti a felhasználót az átlátható időpontválasztásban.

**Alkalmazás Felépítése**

* **FoglalasKezelo osztály:** Kezeli az időpontfoglalásokat, valamint a foglalások betöltését, mentését és megjelenítését. Ez a program logikai központja.
* **FodraszatApp osztály:** Ez a program belépési pontja, amely összekapcsolja a felhasználói interakciót a FoglalasKezelo osztály funkcióival.

**Diagramok Részletezése**

**1. Osztálydiagram (Class Diagram)**

Az osztálydiagram bemutatja a rendszer osztályainak felépítését, azok attribútumait és műveleteit, valamint az osztályok közötti kapcsolatokat. Az osztálydiagram szerint:

* **FoglalasKezelo osztály**
  + **Attribútumok:**
    - foglalasok: Egy kétdimenziós tömb, amely a foglalásokat tárolja napokra és időpontokra bontva.
  + **Műveletek:**
    - betoltes(String fajlEleres): Betölti a foglalásokat egy fájlból.
    - mentes(String fajlEleres): Elmenti a foglalásokat fájlba.
    - megjelenitFoglalasok(): Megjeleníti a foglalási táblázatot a konzolon.
    - foglal(int sor, int oszlop, String felhasznaloNev): Foglal egy szabad időpontot a felhasználó nevére.
* **FodraszatApp osztály**
  + **Műveletek:**
    - main(String[] args): A program belépési pontja, amely a felhasználói interakciót és a FoglalasKezelo hívásait koordinálja.

**Kapcsolatok:**

A FodraszatApp használja a FoglalasKezelo osztályt a foglalások kezelésére.

**2. Használati eset diagram (Use Case Diagram)**

Ez a diagram az applikáció főbb funkcióit szemlélteti, és bemutatja a felhasználók (ügyfelek) és a rendszer közötti interakciókat. Főbb használati esetek:

* **Időpont foglalása:** Az ügyfél egy szabad időpontot foglal a kívánt napon és időben.
* **Foglalások megtekintése:** Az ügyfél megtekintheti az aktuális foglalási táblázatot.
* **Foglalások mentése és betöltése:** A rendszer automatikusan kezeli a foglalások fájlba mentését és betöltését.

**Résztvevők:**

* **Felhasználó:** Interakcióban áll a rendszerrel a foglalások kapcsán.
* **Rendszer:** Kezeli a foglalásokat és azok állapotát.

**3. Interakciós diagram (Interaction Diagram)**

Az interakciós diagram azt mutatja be, hogy az osztályok hogyan kommunikálnak egymással egy adott művelet során. Például:

**Időpont foglalása művelet:**

* **Felhasználó** -> **FodraszatApp**: Kérdés az ügyfél nevéről, majd az időpont koordinátáiról.
* **FodraszatApp** -> **FoglalasKezelo**: Foglalási kérés (foglal() metódus meghívása).
* **FoglalasKezelo**: Ellenőrzi, hogy az adott időpont szabad-e. Ha igen, rögzíti a foglalást.
* **FoglalasKezelo** -> **FodraszatApp**: A foglalás sikerességének visszajelzése.
* **FodraszatApp** -> **FoglalasKezelo**: A foglalások mentésének kérése.
* **FoglalasKezelo**: A foglalási mátrix mentése fájlba.

**Felhasználói Felület Tervezése**

A projekt része egy vizuális prototípus, amely bemutatja, hogyan néz ki és működik a foglalási rendszer felhasználói oldala. A design fő célja a könnyű kezelhetőség és a modern, letisztult megjelenés biztosítása. Az UI design figyelembe veszi a felhasználói élményt (UX), és támogatja a hatékony navigációt a rendszer funkciói között.

**UI Design Részletei**

1. **Kezdőoldal**
   * Tartalmazza a **felhasználó név bekérésére szolgáló mezőt** és a bejelentkezés gombot.
   * Átvezet a foglalási oldalra, ahol az időpontokat lehet kiválasztani.
2. **Foglalási Oldal**
   * **Foglalási mátrix**:  
     Egy vizuális táblázat, amely bemutatja a különböző napok és időpontok foglalási állapotát (szabad vagy foglalt).
   * **Időpont kiválasztás**:
     + A felhasználó rákattinthat egy szabad időpontra.
     + A kiválasztás után megjelenik egy popup ablak a megerősítéshez.
   * **Foglalási adatok mentése**:  
     Egy "Mentés" gomb, amely rögzíti a választott időpontot.
3. **Visszajelzés Oldal**
   * Egy egyszerű képernyő, amely értesíti a felhasználót a sikeres foglalásról, valamint a következő lépésekről.
4. **Rendszerüzenetek**
   * **Sikertelen foglalás**: Amikor egy foglalt időpontot próbál megválasztani a felhasználó, megjelenik egy értesítés, amely új időpont kiválasztását kéri.
   * **Sikeres mentés**: A foglalás véglegesítése után a felhasználó kap egy visszajelzést.

**UI Komponensek és Funkciók**

Az UI design interaktív elemei a következők:

* **Időpontok színek szerinti kódolása:**
  + Zöld: Szabad időpont.
  + Piros: Foglalt időpont.
* **Interaktív táblázat:**
  + A táblázat celláira kattintva a felhasználó közvetlenül kiválaszthatja a kívánt időpontot.
* **Navigáció:**
  + Vissza a kezdőoldalra.
  + Tovább a foglalás mentéséhez.

**UI Technológiák és Implementáció**

A felhasználói felület a következő technológiákat használhatja:

* **HTML/CSS**: A statikus elemek és az oldalak stílusának kialakításához.
* **JavaScript**: Az interaktivitás és az adatok validálása érdekében.
* **Bootstrap vagy más UI keretrendszer**: A reszponzív design biztosítására.