**Cinema API**

**Tartalom**

BEVEZETÉS 3

PROJEKT CÉLJA 3

KÖVETELMÉNYEK SPECIFIKÁCIÓK 3

FELHASZNÁLÓI KÖVETELMÉNY 3

USE CASE DIAGRAM 4

RENDSZERKÖVETELMÉNYEK 4

FUNKCIONÁLIS 4

NEM-FUNKCIONÁLIS 5

TERVEZÉS 5

ÖSSZEGZÉS 5

TOVÁBBI FEJLESZTÉSEINK 5

**Bevezetés**

A mai világ egyik elterjedtebb szórakoztatási formája a filmek. Mozikba járunk, internet segítségével nézzük a kedvenc filmjeinket és sorozatainkat, egyesek akár a saját otthoni mozijaikat ("home theater") is berendezik, hogy jobban élvezhessék a filmjeiket.

Mi több, a film története és grafikai effektusai mellet érdekel minket a film stábja is. Szinte mindenkinek van egy, vagy több, kedvenc színésze, producere, rendezője. Asszociáljuk, hogy egy adott színész mindig a főhőst játssza, vagy a gazfickót, vagy éppen csak egy háttér-karaktert. Bizonyos producer-ek munkáját konzisztensen kedveljük, másokét pedig kerüljük.

**Projekt Célja**

Jelen projekttel azt a célt tűztűk ki, hogy létrehozzunk egy API-t, (Application Programming Interface), ami segítségével a felhasználó információkat érhet el a kedvenc filmjeiről. Kik szerepelnek benne? Mikor készült el a film? Ki munkája valójában? Ezekre a kérdésekre mind választ ad a Cinema API.

A projektnek emellett tartalmaznia kell egy felületet, (user-facing, front-end), ami segítségével a felhasználó intuitív módon hozzáfér az említett API-hoz. Lehetőséget adunk böngészni a filmek, mozik, színészek között, és a megfelelő hozzáférésű felhasználó akár frissítheti is az adatokat, vagy új információt adhat hozzá az adatbázishoz.

Bár az általunk elkészített front-end felület egy weboldal, az API előnye, hogy más környezetben, például egy mobil-alkalmazásban, vagy egy terminál-programban is felhasználható.

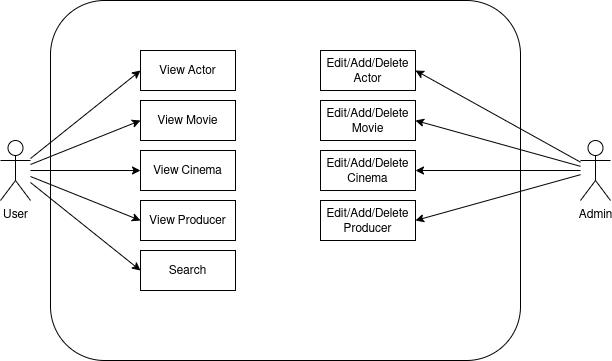
**Követelmények specifikáció**

**Felhasználói követelmények**

A felhasználó amint megnyitja a weboldalt a "Movies" (magyarul Filmek) oldalt látja elsőként. Ezen az oldalon található négy darab film. Mindegyik filmnek van egy címe, egy képe, egy kategóriája, kezdési dátuma (mikortól lett/lesz érvényes a film a mozikban), végzési dátuma (mikortól nem lesz többet érvényes a film a mozikban), státusza (ami a mai dátumhoz viszonyítva kiírja, hogy érvényes-e az a film vagy sem), "Show Details" gombja és "Add To Cart (ár)" gombja. Ezen az oldalon vannak menük is, amik a következőképpen vannak elnevezve: "eTickets" (ha erre rákattint a felhasználó, akkor visszarakja az eredeti oldalra), "Movies" (ugyanazt csinálja mint az "eTickets") és "Management". A menüben a jobb oldalon található egy keresési szöveg doboz.

Ha a felhasználó rákattint a "Management" elnevezésű gombra, akkor három linket talál: "Producers", "Actors", "Cinemas" (magyarul, Szervezők, Színészek, Mozik). Mindegyik link egy külön-külön oldalra irányítja a felhasználót. Ha a felhasználó rákattint az "Actors" gombra, akkor egy új ablak ugrik fel ahol megtalálhatok a színészek nevei, képei és leírása. Ezen az oldalon mindegyik színésznek van külön-külön képe, leírása, és a következő gombjai: "Delete" (Törlés), "Edit" (Szerkesztés), "Details" (leírás). Ha ezen az oldalon a felhasználó legörget, akkor egy olyan gomb jelenik meg amin azt írja, hogy "Add New". Ennek a gombnak a segítségével a felhasználó beteheti akár a kedvenc színészének az adatait.

**Use Case**



**Rendszerkövetelmények**

**Funkcionális**

* A weboldal megnyitása következtében megjelenik a "Movies" oldala.
* A "Management" gomb lenyitása következtében megjelenik három darab link: "Producers", "Actors", "Cinemas".
* A felhasználó rákattinthat ezekre a linkekre és azonnal a felhasználó át lesz téve egy új ablakba.
* Ezekben az ablakokban a felhasználó megtalálhassa a színészeket, a film szervezőit illetve a mozi termek nevét.
* A felhasználó az "Actors" oldalon hozzátud adni új színészt ha szeretne.

**Nem-Funkcionális**

A weboldal megfelelő működéséhez a következő rendszerkövetelményeknek kell megfeleljen:

* Laptop/Számítógép
* Windows 10/11 alapú operációs rendszer
* Egér/Laptop érintőfelület
* Internet hozzáférés
* Böngésző

**Tervezés**

A tervezés legfőbb lépése az API megépítéséről szólt. Az elvárt környezet már ismert volt, ezért csak a kód struktúrájáról kellett döntenünk. MVC (Model-View-Controller) struktúrára esett a választás.

A Controller áll minden központjában - az ő felelőssége a komponensek közötti kommunikáció. Az MVC szellemében figyelnünk kellett arra, hogy a Controller ne tartalmazzon olyan kódot, ami nem tartozik az ő területébe.

A Model komponens felelős az adat kezeléséért (data logic). Itt férünk hozzá a konkrét adatbázishoz, és adjunk vissza a Controller kérésére például egy színész információit.

A View lényegében a megjelenítő réteg. Jelen esetben HTML/CSS/JS kódot tartalmaz, és olyan feladatokat végez, mint szövegek (nevek, filmcímek, stb.) vagy képek (egy film posztere, egy színész fotója) megjelenítése.

**Összegzés**

**További fejlesztéseink**

Mi szerettük volna ezt a weboldalt avval kibővíteni, hogy legyen regisztrálási lehetőség a felhasználónak, illetve rendelési hozzáférése.

Valamint még azt szerettük volna megcsinálni, hogy a weboldal működjön telefonon is, viszont ezeknek a megvalósításához több időre lenne szükség.